Plant Image Process Based On Computer Vision

Zhenxiang Lin
Faculty of Computer Science and Engineering
University of New South Wales
NSW, Australia
z5240946@ad.unsw.edu.au

Qianhui Guo
Faculty of Computer Science and Engineering
University of New South Wales
NSW, Australia
z5291528@ad.unsw.edu.au

Pengchao Zhai
Faculty of Computer Science and Engineering
University of New South Wales
NSW, Australia
z5277229@ad.unsw.edu.au

Haojin Guo
Faculty of Computer Science and Engineering
University of New South Wales
NSW, Australia
z5216214@ad.unsw.edu.au

Xiaoyu Dong
Faculty of Computer Science and Engineering
University of New South Wales
NSW, Australia
z5323011@ad.unsw.edu.au

***Abstract*—This paper proposes several plant image process algorithms based on computer vision containing plant detection, segmentation and then evaluates the performance by Average Precision (AP), Dice Similarity coefficient (DSC), and Intersection over Union (IOU) measures, Symmetric Best Dice respectively(SBD). The evalution of AP is mainly applied to estimate the effectiveness of plant detection and localization in this paper, which in this paper are mainly based on color feature(HSV) and combined with Hog-SVM classifier. And effect of the process can have a relative high efficiency. Texture detection filter and K-means clustering clustering is applied to segment the plant information. In this way, the projected leaf area is successfully extracted from the background. Unet is used and segments the leaves successfully. It can get a high SBD score for special plants, while for some samples, its performance is still unsatisfied.**

Keywords—HSV color-space, HOG, Average precision, bounding boxes, Machine learning, SVM, Image Blurring, CIELAB, Texture detection filter, K-means clustering, DSC, IOU, Unet, SBD

I. Introduction

The dataset for task 1 and task 2 has a total of 70 tray images, with a bounding box around each individual plant in the tray, and the data of the bounding box can be obtained from the CSV file. The plant is photographed at different stages during its life-cycle. It is a challenge because of the neon light illumination, shadows, and moss as shown in Fig. 1, which result in noise and scene complexity.

Fig. 1. Example of a Tray Image from Ara2012

The dataset for Task3 includes many groups of original colorful images and labelled images, which are divided into three categories according to plant type and shooting mode and year, and stored in three folders respectively, namely Ara2012, Ara2013-Canon, Tobacco. There are 120,165 and 62 images in the three folders separately. Labelled images divide all plant leaves in the original picture into separate leaves and label them with different colors. They will be used as ground truth in the calculation loss function in the training set of deep learning and the assessment in the test set. The combination of three folders and each of them will be set as data set separately.

In task 1, we will detect the green plants in the given tray images, draw bounding boxes around each of them and display the total number of plants in the image. Furthermore, we evaluates the performance of the algorithm applied with Average Precision (AP).

Biomass is essential as a kind of plant breeding feature. Biomass is measured as projected leaf area (PLA) to reflect the overall plant quality. In task 2, the texture detection filter is applied to get the PLA (the number of plant pixels) and then is evaluated using the Dice Similarity coefficient (DSC) and Intersection over Union (IOU) measures.

Task 3 is an instance segmentation problem. For the same picture, semantic segmentation means to separate or extract different kinds of objects, while instance segmentation requires to separate different instances of the same object. We will segment each leaf from the plants and label them respectively. Both traditional and deep learning methods could be considered for task 3. According to Kulikov et al, some neural network for semantic segmentation can also be used for instance segmentation.[4] We considered FCN and Unet. After the comparison of their effect from different researches, finally, we chose the deep learning method that expands limited data sets to train models by building Unet and selecting the appropriate loss function. Symmetric best dice was used to evaluate their performance. In addition, we also test the influence of different parameters in the model on the accuracy, trying to find the optimal model for a specific training set.

II. Literature Review

Plants play an important role in the ecosystem and it is an important source for human being's life. Plant phenotype studies how the interaction between the plant genome and the

environment affects the observable traits of a plant (phenome). Computer vision can be applied to study the plant phenotype, identify the features of the plant in its different stages of the life cycle for further application. Therefore, researchers can explore multiple issues in the environmental or agriculture area. For example, plant growing; resistance to plant parasites or diseases, resource usage efficiency; and how to minimize the bad environmental impact. There are numerous papers recording the applications of computer vision techniques in environmental or agriculture related aspects. With the assistance of computer vision, human beings can pursue longer-lasting agricultural development and a healthy relationship between human beings and the earth.

In this paper, for the detection and localization of plants, the core idea of this task is that making the local target appearance and shape of the target plant in the given image can be well described. In hence, the easiest idea to think of in modern computer vision processing methods is to use Histogram of Oriented Gradient (HOG) [6] to describe the object and counting the gradient direction histograms of the local areas of the image. Furthermore, this processing method is quite extensive in the field of pedestrian or face detection [12]. To be more specific, the sliding window method is used In the detection process, the detection window size is fixed, the scale of the image to be detected is scaled, and the fixed-size sliding window is used to extract the Hog feature on each layer of the image, and detection window is judged according to the trained classification model whether is the target (pedestrian). However, the sliding window in this method is not effective in target detection and positioning of plants. The biggest reason for this is that the characteristics of plants are very different from those of pedestrians or human faces, for example, different plants have different growth cycles, thus their size and details are very different, moreover, each plant could have many breakpoints and interference factors when sliding window is detected, which will reduce the whole detection effect.

However, the part of Hog extraction and SVM classifying from Pedestrian detection mentioned above are quite useful, and these techniques will be applied in this project. In addition, in order to better solve the defects of the above methods (sliding window) in the task1, it is noted that extraction based on HSV color features [5] can be better to adapt the plant group in this experiment, and the important reasons for this are that the color green is the most important and significant feature in images, and also HSV color-space is easier to represent a certain color than RGB. What's more, according to the paper of detecting unstructured road method [13], the approach of using HSV is insensitive to the shadows and environment of objects. Therefore, the HSV color-space with the feature extraction of Hog can be applied in this project.

To segment the plant to obtain the PLA, we should analyze the complicity of the scene. In this task, the illumination, shadows and moss contribute to the complicity of the scene. To solve these challenges, Massimo et al. apply Texture From Blurring in the CIELAB color space, then do the clustering using K-means algorithm.[1]

For clustering part, initial centroids should be chose carefully. According to Massimo et al, there are two methods to choose. The first one is using Ostu's thresholding on EXG image and getting a single threshold to determine the initial centroids. The second one is taking advantage of clustering result. Comparing the performance of two method, we choose the second one to achieve better result. After evaluating the viability and performance of this algorithms, we simulate their approach to complete the task2.

In instance segmentation area, many research teams have conducted different studies. For the traditional method, 3D histograms, SLIC superpixels, chamfer matching, and watershed are implemented and compared. IPK got 62.6% SBD, Nottingham got 59.0%, MSU got 62.4% and Wageningen got 63.8%.[2] For the deep learning method, FCN could be used. It was firstly mentioned in 2015 and realize the semantic segmentation by neural network, while its accuracy is not high.[3] Kulikov et al. conclude and implement a deep coloring approach based on Unet.[4] Its performance is high compared with other algorithms. In this task, we will simulate this approach.

III. METHODS

A. Plant Detection and Localization

The whole implementation method of this task is divided into three parts: color extraction by HSV, merging local boxes, filtering boxes by the classifier (Hog-SVM) and main steps of plant detection in the image are shown in Fig. 2.

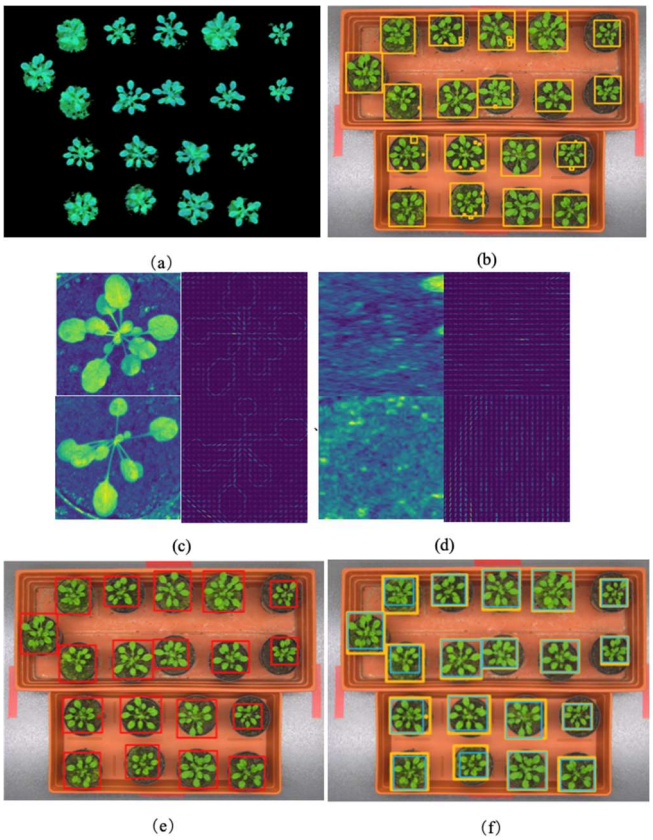

Fig. 2. Description of plant successfully detected. (a) The green-extracted portion with morphological dilation. (b) Draw contour boxes after getting green part. (c)Positive samples and Hog features extraction. (d) Negative samples and hog features extraction. (e) Detected bounding boxes after merging local boxes and Hog-SVM classifier. (f) The comparison between detected bounding boxes(yellow) and the standard boxes(blue).

- *Color extraction by HSV:* In HSV, it is more commonly used in image processing, which is a color closer to the human experience of color [5]. It has three components: hue, saturation and value. HSV describes colors (hue) according to its saturation or amount of gray and the value of brightness. In the task1 of plant detection, therefore, the main method to detect the tray plant for sample images is based on HSV color space partly because the green part is the main color feature in the foreground of the image. Meanwhile, after extracting the green area, the necessary morphological dilation will be performed to reduce the influence of breakpoints upon the detected part. Then the green portion can be framed for the first time.

- *Merging local boxes:* It is inevitable that will be affected by light and moss, resulting in many additional "noising bounding boxes" after the image is color-extracted and bordered by HSV. Hence, the first filter for these borders is by merging the local boxes. Specifically, setting the center distance of any two boxes to less than 100 pixels can be combined. This process could remove the most interference boxes (mainly from moss) within the range of the plants.

- *Hog-SVM:* The second filtering process for bounding boxes extracted by color is achieved by forming a classifier through Hog-SVM. In the process of image processing, feature extraction of targets through the Histogram of Oriented Gradient (Hog) is an effective approach [6], especially in the local operation of the image, the change of image geometry and optics has good stability. The classifier of support vector machines (SVMs) is highly effective for binary classification problems [7]. In this experiment, 80% of images of datasets are taken out as training sets and 20% as test sets. Both positive (each plant in the image) and negative (random samples around each plant) training samples will be extract features by Hog and generate labels. Finally, the training process is completed through the SVM classifier. This classifier can largely filter out most invalid bounding boxes.

The outputs for task1 include the total number of correct plants in each image and the comparison between detected bounding boxes and the given correct contour box on each image, meanwhile, the evaluation of these method of algorithm and the analysis of experimental results will be shown in Section IV and Section V.

B. Plant Image segmentation

- *Image Blurring:* Edges or noise belongs to the high-frequency content. They can be removed via convolving the image using a low-pass filter kernel. OpenCV gives four methods of blurring techniques. Averaging Blurring(AB) and Gaussian Filtering(GF) are used in task2. Using a normalized box filter to convolve the image can obtain AB. It takes the average of all pixels under the kernel area, and then replace the central element with this average. We need to specify the width and height of the kernel. GF is efficient for removing Gaussian noise from the image with a Gaussian kernel.[8]

- *CIELAB color space:* To overcome the challenge result from the neon lighting, shadows, and moss, we need to transfer the RGB color space to the CIELAB color space. In order to make the color features, CLELAB uses the a* component to represent the color position from green to red and uses the b* component as the color position from blue to yellow.[1] Even though CIELAB does not take the Helmholtz-Korlaus effect into consideration, it still aims to make the image more likely to be the human beings' feelings to lightness. Therefore, in contrast to RGB or CMYK spaces, CIELAB aims to approximate human vision. Via changing output curves of a* and b* components, and using the L* component to change the lightness contrast using, it corrects color balance accurately.

- *Texture detection filter:* Firstly, applying a uniform kernel H_ρ where the radius is ρ. Secondly Linearly combining the previous result with a Difference of Gaussians (DoG) filter. The DoG filter is used to filter high texture parts in the plant image *I*. To be more specific when applying the DoG filter, two blurred versions of an intensity image on the same image are subtracted (the Gaussian kernels for two blurred version are $K_{\sigma H}$ *and* $K_{\sigma L,}$ and the standard deviation are σH, σL respectively). The result after applying the filter is as follows:

$$f(I;\rho,\sigma_H,\sigma_L) = H_\rho * I_i + (K_{\sigma H} * I_j - K_{\sigma L} * I_j), \quad (1)$$

In this formula, I_i and I_j are channels of LAB color space we transferred in the previous step, and $*$ is the operator of two matrices' multiplication. The result of the "Texture From Blurring" (TFB) filter should furtherly be transformed as the formula shows:

$$f_{TFB}(I;\alpha,\rho,\sigma_H,\sigma_L) = exp(-\alpha|f(I;\rho,H,L)|), \quad (2)$$

where α is the rate of decrease. The pillbox filter is relative to smooth parts and parts with high texture, and the DoG filter is relative to edges. These two filters are applied in the task of classifying moss or earth, which the high texture regions from leaves and stems, which is the smooth regions.

(a^*, b^*) forms the color information, and TFB is the feature space. We set $\sigma_H = 4, \sigma_L = 1, \rho=3, \alpha= 0.02,$ $I_i = a^*$, $I_j = L^*$ in this task.

- *Dice Similarity coefficient (DSC):* It is used for measuring the similarity between two sets of. It has been one of the most popular approaches in the validation of image segmentation algorithms. It can also be applied in other areas, for instance, Natural Language Processing.

The formula for DSC is:

$$DSC = 2 \times \frac{|A \cap B|}{|A| + |B|} \quad (3)$$

where A and B are two sets, a set with vertical lines either side, for example, $|A|$, represents the number of elements in that set, $\cap$ means the intersection of two sets[9].

- *Intersection over Union(IOU)*: IOU is another metric to measure the similarity between two sets. To compute it, we need to get the size of the intersection part and divide it by the size of the union part of two sets.
- *K*-means *clustering:* It clusters data by classifying samples in n groups of the same variance. This algorithm minimizes a criterion, which is known as the inertia or the within-cluster sum-of-squares to choose centroids. The number of clusters needs to be specified. It has a good performance for a large number of samples and has been applied in many aspects [10].

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2), \quad (4)$$

The main idea of the K-means algorithm is dividing the N samples X into k clusters. Each of them represents the mean value of samples in this cluster. These mean values are called the centroids, however, they are commonly not the points from X even they are in the same space.

When the K-means algorithm running, it randomly chooses k samples from the data X as the initial centroids, alternatively, you can choose initial centroids by hand. After the initial step, the K-means algorithm will repeat the following steps. The step one is assigning each sample the close centroid. The next step is getting the mean value of samples for each centroid. By computing the difference between the current and previous centroids, if the value is less than the threshold, the repetition stops. The result is the final centroids which are stable and do not change significantly for further repetition [10].

C. Instance segmentation

For instance segmentation, the architecture of unet is shown as Fig. 3. As can be seen, the net consists of two parts: up-sampling and down-sampling. Firstly, the original RGB image is changed to 32 channels by a convolution. Then, through down-sampling, up-sampling and a convolution with 1×1 kernel, the prediction output with C channels is generalized. C is the number of output channels which are a hyper-parameter and will be discussed in the experimental setup.

Unet Architecture

- *Down-sampling:* It is a process of feature extraction. It consists of three down-modules. Each down-module include four convolution layers and a max pooling layer. The first convolution layer doubles the number of image channels. The other three convolution layers do not change the number of channels. After convolution, a dropout layer is also used to avoid over-fit. For down-sampling, ReLU is used as an activation function.
- *Up-sampling:* It is a process of returning the images by the extracted feature. It consists of three up-modules. Each up-module includes a deconvolution layer and two convolution layers. The deconvolution layer cuts the number of image channels in half and the convolution layers do not change the number of channels. For up-sampling, ELU is used as an activation function.
- *Concentration:* The first image of each down-module and up-module will be concentrated.

Prediction label generalization

- Each channel of prediction output is a feature map. The largest value in all channels is extracted and used as the new pixel value of the prediction label image.

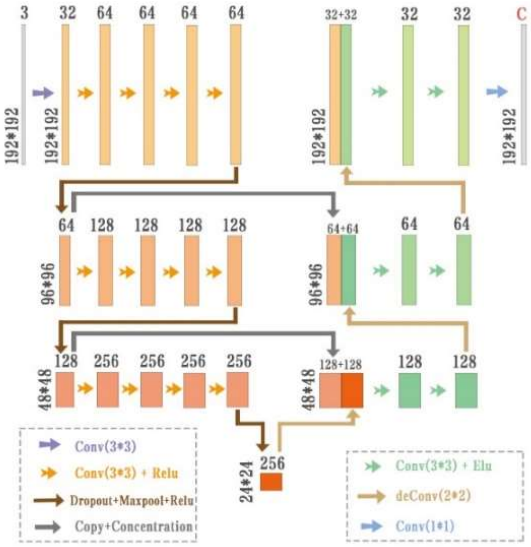

Fig. 3. Architecture of Unet

IV. Experimental Setup

A. The Process of Detection and Localization of Plants

1) Generate a SVM classifier

a) Read the image: The dataset has a total of 70 tray images, with a bounding box around each individual plant in the tray.

b) Read the data describing the plant bounding box area from csv file.

c) Extract the plant bounding box area according to the data in csv file as the positive samples (80% of the tray datasets).

d) Randomly creating negative samples around each plant. A certain number (0~5) of negative samples of a random size (0~200 pixels)are obtained around each plant.

e) Obtain Hog feature vector list from positive and negative samples as the X train data.

f) Generate 1 and 0 labels to positive and negative samples repectively manually as the y predict data.

g) Use data from step e) and step f) train the SVM

2) Process the original image and get the contours

a) Remove noise: non-local Means Denoising algorithm is applied for removing noise in color images.

b) Green features extraction: Transfer RGB original image to HSV image, setting the green lower range as (34, 43, 46) and the green upper range as (77, 255, 255), then extract the green area(the plant mostly), the green-extracted portion is shown in Fig. 2.

c) Morphology: Dilation of the green plant to remove breakpoints, and therefore plant can be detected more easily.

d) Draw the contours through the features function of drawCoutours.

e) Merging local boxes first time: if the center distance between any two neighbor contours within cetrain values (100 pixels), replacing them with the lager one.

f) Filtering the contour boxes scend time: by using SVM classifier trained in step 1), remaining the box whose label is 1, and finally get bounding area of each plant.

3) Count the number of plant contours correctly detectd in each image.

4) Draw the contrast result: detected bounding boxes and the ground truth boxes upon the images for comparsion.

5) Evaluate the performance of the algorithm: using two type of AP metrics described in Section III, and the specific details will be offered in Section V.

a) Average Precision(AP): Average precision values are selected from the maximum precision at each recall value interval on a precision-recall curve, which means AP depends on the multiplication of the mean of the values of precision and recall, and the value of recall is actually close to 1 and precision, hence using precision approximately represents the AP, and the precision is the ratio of true positive results to the total positive results. In this task1, the AP is simply calculated according to the images in the test dataset from Ara2012, Ara2013-Canon, and Ara2013-Rpi, which represent only one class. It is worth noting in task1 that we calculated two AP values from two angles of the experimental results as the metrics to evaluate the results on the test dataset. To be more specific, based on the correct given bound boxes from CSV files, AP_1 focuses on the number of boxes predicted to be correct. AP_2, however, focuses on the exact precision of each prediction as the correct prediction bounding boxes. These mentioned procedures can be formulated as:

$$Precision = \frac{TP}{TP + FP} \approx AP_i \tag{5}$$

$$AP \approx \frac{1}{n_{samples}} \sum_{i=0}^{n} (AP(n_{samples})) = \frac{1}{n_{samples}} \sum_{i=0}^{n} \left(\frac{TP_i}{TP_i + FP_i}\right) \tag{6}$$

where TP and FP refer to true positive and false positive, and n is the number of test tray image of the dataset for AP_1, TPi and TPi+FPi are the number of contour bounding boxes as correct and the number of all ground truth contour boxes respectively. For AP_2, each parameter represents a different meaning, where n refers to the number of all plants in all test images, TPi is the overlap area of the correct detection of the bounding box and the correct box, and TPi+FPi means the area of the ground truth bounding box.

B. The implementation of Plant Image Segmentation

1) Converting the color space from BGR to CIELAB where component L determines the lightness from black to white (0-100), a* component determines the color value from red to green, and the b* component determines the color value colors from yellow to blue.*

*2) Applying TFB (Texture from blurring) filter on L*a*b* color space and get a new color space, two steps as follow:*

- *Appliying uniform blurring filter linearly combined with a Difference of Gaussians (DoG) filter.*
- *Using the following formula to transfer f and get the final response of TFB filter:*

$$f_{TFB}(I;\alpha,\rho,\sigma_H,\sigma_L) = exp(-\alpha|f(I;\rho,H,L)|) \ , \tag{7}$$

3) Using K-means to cluster the new color space where k = 2.

a) Choose inititial centroids for K-means clustering, the two methods are as follows (we choose the second one in this task):

- *Applying Ostu thresholding on EXG image to get a binary image. Getting a single threshold based on histogram of this binary image.*

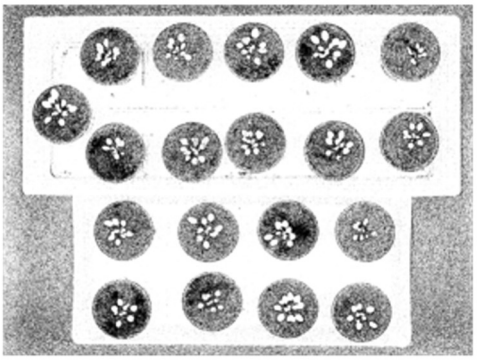

Fig. 4. Binary image after Ostu's thresholding

- *Run K-means algorithm 10 times with different ramdom centroids, choose the best result of 10 runs,*

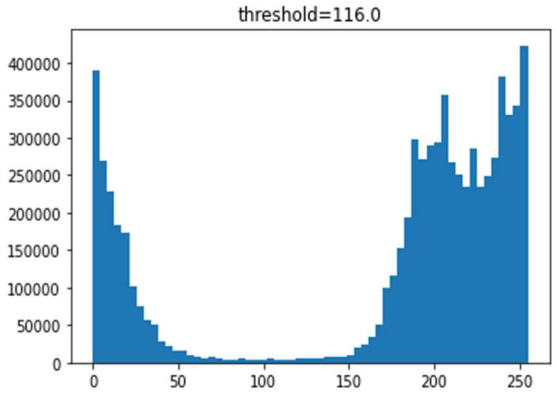

Fig. 5. Histogram of binary image

then choose its centroids as the initial centroids.

b) Apply K-means clustering(k=2) and get the final binary image.

4) Evaluate the performance of the algorithm using DSC (Dice similarity coefficient) and IOU (intersection over union also known as Jaccard similarity coefficient) metrics.

C. Instance Segmentation Implementation

1) Experiment Process

Fig. 6 shows the whole process of Task 3.

a) Preprocessing: After the structure of the neural network is finished, some preprocessing operations are necessary the neural network and achieve good results. The initial step is to unify all images, including RGB images and label images, into the same shape, so as to facilitate the subsequent pre-processing steps. After this, the RGB image needs to be normalized and pixel values converted to the values in the [0,1] interval. We also need to digitize label images to convert pixel values that appear in an image to a continuous value starting at zero (zero being the background).

b) Splitting Sets: In this experiment, because there are three different files and each concludes images of plants with similar phenotypes, we plan to use the images in each folder as training sets to train different models respectively. We have also created a data set that includes all images in three folders. For each folder, we set 70% of the images as training sets and 30% as test sets to evaluate the performance of the model.

c) Random Transformation: Obviously, the datasets available are very limited. In order to train the model with higher accuracy, sufficient training sets are necessary. so we adopt the random transformation to the original images to expand the training set In this way, the same pairs of RGB images and their label images, as source, can be converted into multiple pairs of images used for training. These include, flipping up and down, flipping left and right, rotating by some angles, warp and so on. After these random transformations, the values of the array need to be unified in the interval of [-1,1] again, and in order to meet the requirements of neural network input, the batch dimension needs to be expanded to four dimensions.

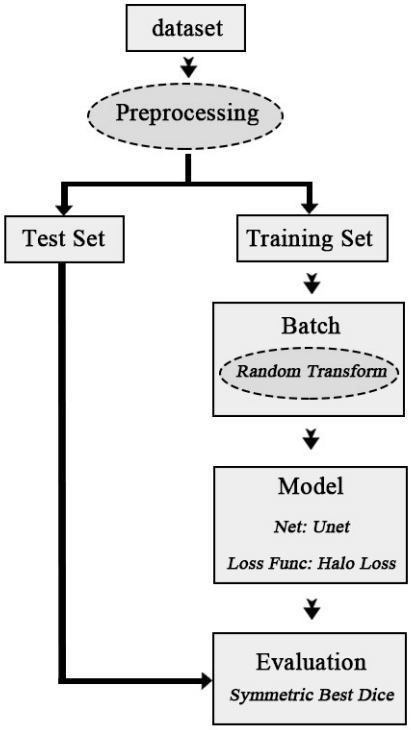

Fig. 6. Process of experiment

d) Evaluation Criteria: We used the Symmetric Best Dice[11] measure whose algorithms and formulas have been developed and applied by other researchers. For the label of the model output, traverse each leaf label and try to find the label most similar to it in the ground truth. This has to do with how well they fit together in pixels. Obtain the mean value of the fitting degree of all output labels. Then, based on ground truth, the output label is compared with the same method. Take the minimum value of the two means as the basis to evaluate the performance of the model.

2) Experiments Effect Elements

There are many elements that could affect the result of neural network output. They are as follows:

- *The number of output channels C:* It is required to be controlled into an appropriate range. Either a too large or too small value will lead to a low performance.
- *The epochs number M:* This parameter will affect the size of dataset fed to the network. A larger epochs number will lead to a larger dataset.
- *Loss function:* Different loss function will cause a great difference of result. In this paper, the function provided by [4] is used. Meanwhile, the hyper-parameters of loss function are also followed.

V. Results and Discussion

A. Evaluation of Detection and Localization

In order to evaluate the entire process of the algorithm from different perspectives. There are two types of calculations of average precisions (APs) as mentioned in Section III. In short, the first AP focus on the number of correct plants detected, which refers to the quantity macroscopical detection to a large extent. However, the second is based on the overlap area between detected bounding boxes out of the image and the ground truth contour, and it reflects the quality of the microscopic details each detected bounding box. This paper counts the APs values of the three times and compute the average value, and these results are recorded in TABLE I. It is shown that the AP_1 and AP_2 can achieve approximately 96.45% and 94.86% respectively.

TABLE I. THE EVALUATION OF AVERAGE PRECISION (AP)

Average precision	*1*	*2*	*3*	***Mean***
AP_1	95.72%	97.28%	96.34%	96.45%
AP_2	94.02%	96.17%	94.39%	94.86%

[a.] Notes: AP1 based on correct plants, AP2 based on correct bounding box areas.

Additionally, 80% of the samples are randomly obtained as training sets through the three datasets, Ara2012, Ara2013-Canon, and Ara2013-Rpi, separately. The positive training set can obtain from the CSV data and training sets above. The negative training set, samples are randomly extracted from the truth bounding box of each plant around. Then the positive and negatives samples are trained by Hog-SVM, which is used to filter the detection box. For these reasons, the outputs of this task eventually are based on 16 test images, including four samples of Ara2012, six samples of Ara2013-Canon, and another six images from Ara2013-Rpi. And the

number of bounding boxes correctly detected in each image is counted in TABLE II. By comparing the result of the data in TABLE II, it is found that the error range between the number of detected and localized plants and the actual number of the plants is from 0 to 3.

TABLE II. THE TOTAL NUMBER OF DETECTED PLANTS IN TEST IMAGE

Test Image No.	***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***
Num of detected plants	19/ 19	17/ 19	17/ 19	19/ 19	23/ 24	23/ 24	23/ 24	24/ 24
Test Image No.	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***
Num of detected plants	24/ 24	24/ 24	24/ 24	22/ 24	24/ 24	23/ 24	23/ 24	23/ 24

b.

Notes: The detected results are on the left, and the correct number of plants are on the right. In addition, this table only records the results of one experiment.

c.

Combining the experimental results above, it is shown that the algorithm in this task has a relatively positive effect on plant detection and localization. However, on the other hand, this experiment did not achieve 100% of the plant target positioning, and the correct rate of the test results of the datasets of Ara2012 is relatively lower than the other two datasets of images. Therefore, there are still a lot of work to be modified and improved for this the experimental algorithm, for example, how to achieve a more reasonable and comprehensive acquisition of negative samples from images, and the specific analysis will be shown in Section VI.

B. *Evaluation of Plant Image Segmentation*

After applying texture detection filter and do the clustering on the new new color space, we get the binary images which successfully segment the plant information from the background. Fig. 7 are the comparation between binary image we got and the ground truth image.

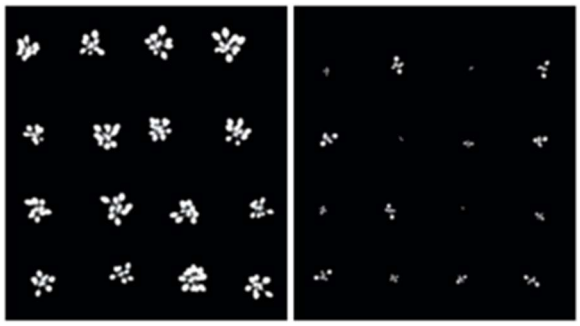

Fig. 7. Binary image after cluster

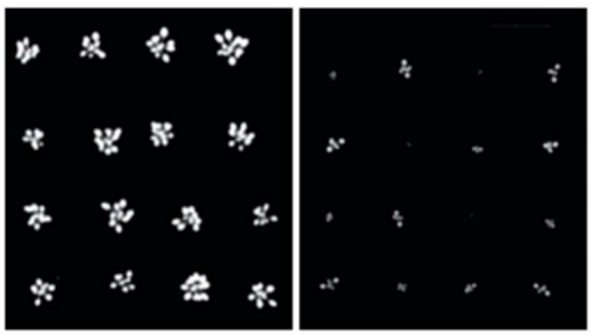

Fig. 8. Ground truth image

We use two metrics to do the evalution of the segmentation, IOU(Intersection over union) and DSC(dice similarity coefficient). The weighted Average IOU is up to 96.28% and DSC is up to 92.93%.

TABLE III. THE IOU AND DSC EVALUATION

	IOU	***DSC***
Ara2012	93.94%	88.67%
Ara2013	97.66%	95.45%
Weighted AVG	96.28%	92.93%

C. *Evaluation of Instance Segmentation*

1) *General Result*

Taking SBD as the evaluation criteria, the highest accuracy of four models is shown in Table IV.

TABLE IV. THE SYMMETRIC BEST DICE OF FOUR MODELS

	Ara2012	***Ara2013***	***Tobacco***	***Combination***
SBD	76.42%	64.20%	42.90%	51.97%

The model with the best performance we obtained so far is the one with the images in the Ara2013 folder as the training set (C = 9, epoch = 5000), and the evaluation of SBD is 76.42%. Relatively, the model of tobacco performs pooly. The model of the whole dataset is at the mid-level.

The comparison between the original images, the final output images of the model and the ground truth results is shown Fig. 9.

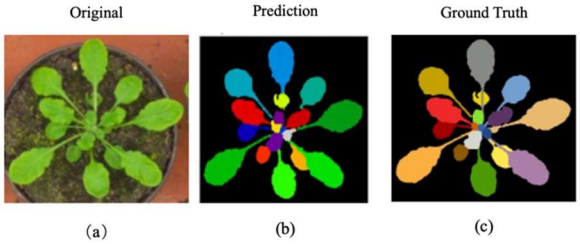

Fig. 9. (a)original image; (b) Prediction image; (c) Ground truth image

The layered images of the nine channels is shown in Fig. 10.

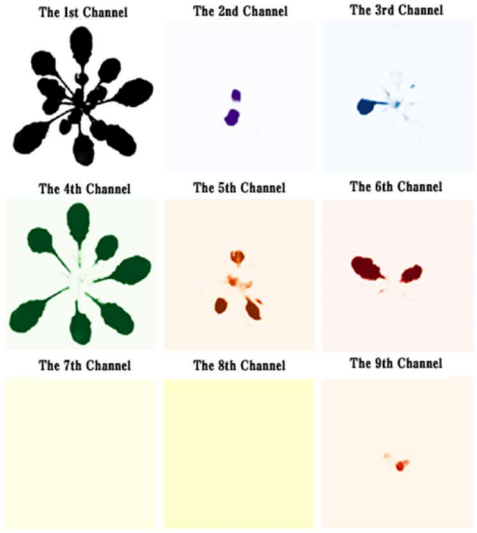

Fig. 10. Images of different channels

It can be considered that this is a relative ideal instance segmentation model. However, this model can only obtain relatively ideal leaf segmentation results in images which are similar to those in the Ara2012 folder. By contrast, the model trained by the tobacco folder was less than ideal. According to the output results of the model, it was speculated that this was due to the high leaf overlap of Tobacco, light, shadow and surrounding objects (such as rust), which also affected the data processing of the model.

2) *Parameter Comparison Results*

a) Epoch:

Fixed other parameters, modifying only the number of Epochs. The results of model accuracy variation obtained are shown in Fig. 11.

It is clear that the performance of the four models all nearly showed a logarithmic increase according to the increase of the number of Epochs. When the number of Epochs is small, the performance of the model increases rapidly with the increase of Epochs. However, fluctuations may also occur. It is speculated that the preprocessing of images and the selection of training sets are relatively random when the epoch is small, which will have a great impact on the training accuracy of the model. As the number of epochs increased, the growth rate of performance gradually slowed down. When the number of Epochs was relatively large, the performance indicators tended to converge gradually.

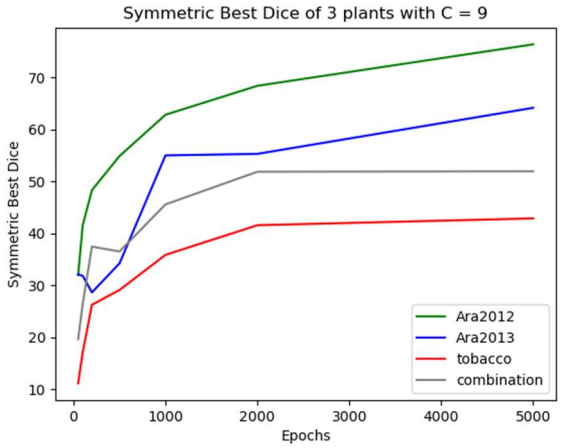

Fig. 11. Symmetric Best Dice of different plants

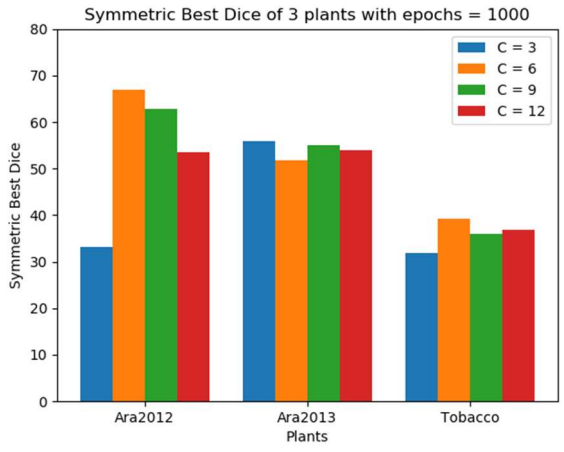

Fig. 12. Effect of channels number

b) Channels number C:

Fig. 12 shows the effect of channels number C. As can be seen, for ara2012 and tobacco, 6 is the best channels number. A small and high value will cause a worse performance. However, for ara2013, 3 is the best value and the performance of 6 is lower. This shows different dataset may has a different best channels number. Therefore, it is better to conduct this part and choose the best channels number firstly.

VI. Conclusion

In terms of detecting and localizing plants, the program finally outputs three types of experimental results, one is about the number of plants to be detected, another is the comparison between the detection bounding boxes and the ground truth boxes out of the image, and the third is the evaluation of the algorithm, that is, the calculation of the two AP values. From the results of TABLE I and TABLE II, it is shown that although that the results can explain that multiple plants in one image can be detected and localized with relatively high efficiency to a large extent, it is still some plants are missed. The existing errors between the experimental results and the truth results could be attributed to the following possible reasons:

- The interference factors are not removed completely, such as moss, and light.
- Negative image samples have limitations, such as the portion between plant leaves and the background area outside the plant range in the image cannot be sampled comprehensively.
- The number of training samples of Ara2012 is limited, the biggest different between the first datasets and the other two is the environment of the plants in the image.

Therefore, the algorithm needs to be improved in terms of removing interference and a more reasonable collection of negative samples, meanwhile, if more training samples can be obtained, we can increase the performance of outputs.

In the task 2, we apply the texture detecting filter to get the new color space, then do the K-means clustering based on the new color space and get the binary image. However the precision is not quite satisfying. Compared with ground truth images, there are some noises in the binary image. We can take the following ways to make an improvement.

- Firstly, the reason behind this is that the centroids we chose are based on one image. We can use more samples to run K-means cluster algorithm to get more proper centroids. It can lower the bias and improve the precision of the segmentation.
- Secondly, we can use current result to localize each plant in the image then do the further segmentation on each plant fraction. We can choose the active contour algorithm or other methods to do further segmentation and get more accurate segmentation of each plant.

For the instance segmentation problem of Task3, we adopt the deep learning approach. It shows that through a series of preprocessing and the Unet structure adopted in the task with appropriate loss algorithm to learn the leaf features in images, a model with moderate performance can be obtained. Our algorithm provides a good beginning for the

problem of leaf instance segmentation. However, due to time and equipment limitations, many parameters may not be set to maximize the accuracy of the model.

In the future, we can adjust the neural network structure, modify the loss function algorithm or use more epochs to train the model to get a better performance, and even test the practicability of this algorithm by treating the images in three folders as a whole training set. From the experience of this experiment, the structure of the neural network should be determined first for each specific training set. For example, prioritize the performance impact of Unet with different channel counts. Then modify the other hyperparameters. The optimized algorithm can be applied to agriculture, botany and computer vision research.

VII. Contribution of Group Members

Zhai: Paper research of plant image segmentation, task 2 part of implementation, report and demo slide.

Dong: Paper Research of plant image detetction & segmentation, introduction, literature review, report of task1 & task2, improve algorithm of task1, reference, demo slides.

Lin: Conduction of Task3 with Guo Q including research of instance segmentation, implementation of Unet , model training and evaluation and report edition.

Guo H: The implementation of the part of plant detection and localization, and the part of method and result analysis of task1 in paper, part of demo slides.

Guo Q: Finish Task3 together with Lin including research of instance segmentation, implementation of preprocessing as well as training model, evaluating result and writing report.

References

[1] M. Minervini, M. Abdelsamea, and S. Tsaftaris, "Image-Based plant phenotyping with incremental learning and active contours," Ecological Informatics, vol. 23, pp. 35-48, 09/01 2014, doi: 10.1016/j.ecoinf.2013.07.004.

[2] H. Scharr et al., "Leaf segmentation in plant phenotyping: a collation study," Machine Vision and Applications, vol. 27, no. 4, pp. 585-606, 2016/05/01 2016, doi: 10.1007/s00138-015-0737-3.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7-12 June 2015 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.

[4] V. Kulikov, V. Yurchenko, and V. Lempitsky, "Instance Segmentation by Deep Coloring," p. arXiv:1807.10007. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2018arXiv180710007K

[5] doxygen. "Changing Colorspaces." docs.opencv.org. https://docs.opencv.org/master/df/d9d/tutorial_py_colorspaces.html (accessed Nov. 12, 2020).

[6] T. s.-i. d. team. "Histogram of Oriented Gradients." The scikit-image development team. https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html (accessed Nov. 10, 2020).

[7] P. e. al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[8] A. M. A. K. R. 43532856. "Smoothing Images." opencv-python-tutroals.readthedocs.io. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html (accessed Nov. 10, 2020).

[9] A. D. Yao, D. L. Cheng, I. Pan, and F. Kitamura, "Deep Learning in Neuroradiology: A Systematic Review of Current Algorithms and Approaches for the New Wave of Imaging Technology," Radiology: Artificial Intelligence, vol. 2, no. 2, p. e190026, 2020/03/01 2020, doi: 10.1148/ryai.2020190026.

[10] F. a. V. Pedregosa, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., "Scikit-learn: Machine Learning in {P}ython," Journal of Machine Learning Research, vol. 12, pp. 2825--2830, 2011.

[11] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," Pattern Recognition Letters, vol. 81, pp. 80-89, 2016/10/01/ 2016, doi: https://doi.org/10.1016/j.patrec.2015.10.013.

[12] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. IEEE, 2005.

[13] Huang, Jingang, et al. "A new method of unstructured road detection based on HSV color space and road features." *2007 International Conference on Information Acquisition*. IEEE, 2007.