Group Component of COMP9517 Project

Group Name: Hairless programmer —— Wenjing Gao, Han Chen, Liqi Jiang, Yang Liu, Yilin Liu

***Abstract*—This paper includes 3 tasks. In task1, it localizes required objects with bounding boxes and find the average precision. In task2, it finds the biomass of the plants from the pictures containing trays of plants. In task3, it performs individual leaf segmentation, which is a multi-instance segmentation.**

Keywords—plant phenotyping, leaf segmentation

I. Introduction

A. Task1

For this task, we need to find a Python solution to detect and localize plants in the input images. And also draw bounding boxes around each individual plant and display the total number of plants in the image. After that, we need to find a way to test and train the model and evaluate the performance of the algorithm using Average Precision.

For the datasets, we have PNG files and csv files. We need to detect every picture and draw correct boxes which are depending on bounding box annotations.

B. Task2

Biomass is one of the important characteristics of the plant growth condition. Plant growth monitoring is an important method to help with precision agriculture. Knowledge of growth condition of the plant can be helpful to find the relationship between plant growing process and growth conditions, which will provide efficient agricultural services, such as proper timing of fertilization, irrigation, and control of insects and diseases. Biomass is proportional to its leave area, especially for broad-leaved plant (Huang et al., 2019) [1]. The objective of task 2 is to find the leave area in the given photos in order to estimate the biomass. Figure 1 shows an example of the give photo in the datasets.

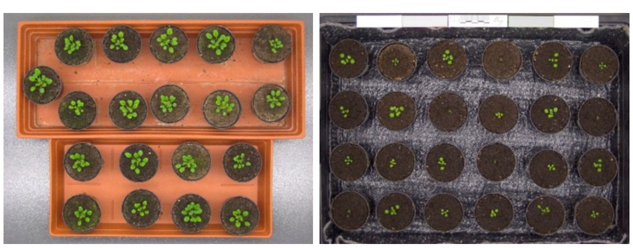


Fig. 1. *Examples of a photo of trays of plants*

Since the photo of the plants are taken from the top of those plants down, it can be assumed that the leave area is projected on the image evenly without distortion. Hence the aim of this task is to find the projected leave area (PLA) of the plants in those photos given in a dataset.

C. Task3

The need of segmenting some specific instances from an image has raised in the application field of modern computer vision. Task 3 has provided an occasion in which images of plants from the Plant Phenotyping Dataset should be segmented with only leaves left. This task is defined as multi-instance segmentation problem which is extensively studied in both computer vision and artificial intelligence fields. In this task, we focus on the necessary processing which can be contributed to accomplishing the multi-instance segmentation task such as image thresholding, filtering, and applying algorithms of segmentation and consider methods raised in other authoritative references.

Images of plants in the Plant Phenotyping Dataset show complete plants with petioles and leaves. An important information is a substantial number of leaves of different kinds of plants are overlapping and almost every plant from the total dataset has unique number of leaves, which causes tremendous challenge when trying to segment leaves from the origin plants with one fixed algorithm.

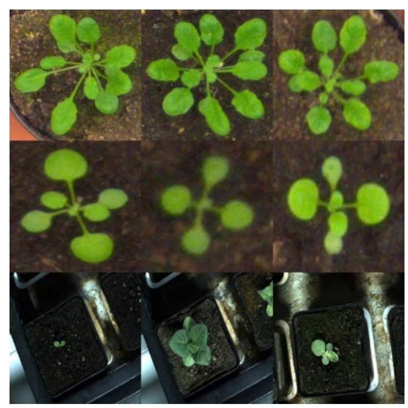


Fig. 2. *Example of original images of plants from the Plant Phenotyping Dataset. Plants from different kinds generally have different numbers of leaves.*

In modern computer vision field especially in agriculture application, however, processing object which is similar to samples in this task is ubiquitous. Therefore, there are a large amount of experiments based on the same dataset. Before implementing our method to solve this problem, many research work has been done on those available literature. Relative comparing and analysis also contain in the report.

This report mainly focus on using the traditionary algorithms of image segmentation as a method trying to solve the multi-instance problem, show the result of this method and will also enumerate difficulty in our experiment.

II. LITERATURE SURVEY

A. Task1

In order to detect and localize objects in the input images, I found two techniques to use. If we only use OpenCV then we can choose HOG and SVM algorithms. While we can also use YOLO algorithm from deep learning.

Firstly, speaking of HOG, this algorithm is to extract useful features from each picture since it can detect objects based on describing the edge features.[2] Its basic idea is to use the gradient information to reflect the edge information of the image target and to characterize the local appearance and shape of the image through the local gradient size. And in order to implement this, we need to divide images into cell units, then collect histogram of pixels and finally combine those histograms into a feature descriptor.

When it comes to SVM [3] algorithm, it is used for the image classification. It is a two-class model. Its basic model is a linear classifier with the largest interval defined in the feature space. And its basic idea is to solve the separating hyperplane that can correctly divide the training data set and have the largest geometric interval.

From above, we need to use both HOG and SVM to detect and classify objects, while it can be inconvenient.

Secondly, in order to reduce the number of steps and make it easier to get the results. We can use YOLO, which has YOLO v4, v5 and so on. YOLO network is composed of three parts: backbone, neck and head. Today we will take YOLO v5 as an example to introduce its background and other information. It was an improved version of YOLO v3. And the new features and some modifications in YOLO v5 are mainly focused around activation functions, data augmentation and post-processing into the established YOLO structure to achieve the best possible object detection performance. Apart from that, YOLO v5 includes four different models ranging from the smallest YOLO v5s and the largest YOLO v5x.[4]. Compared with YOLO v4, YOLO v5 is much faster to calculate and get the final results. The figure1 below shows the network architecture of YOLO v5.

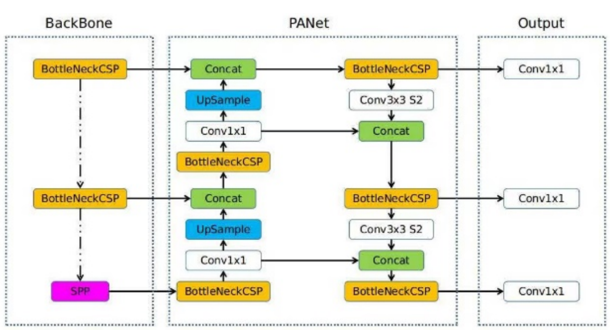


Fig. 3. *Overview of YOLOv5*

B. Task2

In the images of the plants, for example in Figure 1, there are mainly three types of object, each of which has a unique color range. The plant is in green color. The soil is mainly dark brown to light brown. And the tray is either in light brown or black with some white light reflection. Hence the plant can be easily distinguished with the color difference.

The RGB color space of the original photos is affected by the image brightness, and the greenness identification result of segmenting the plant from the background is heavily affected by the brightness. In order to eliminate the impact from brightness, the HSV (Hue, Saturation and Value) color space will play an important role. In HSV color space, the color distribution of a pixel is invariant to the brightness variation (Jose et al., 2010) [5]. Hence, in this task, HSV color space will be used in the segmentation model.

C. Task3

1. *Reduce instance segmentation to semantic segmentation*

Multi-instance segmentation is explained as a method to detect and distinguish individual instances of several semantic classes in images, and in this task, leaves. In general instance segmentation problem, it seems as an image segmentation problem with high-level difficulty because the number of instances is unknown before image processing. However, the problem of semantic segmentation, which is similar to instance segmentation, can be sufficiently solved by deep learning architectures. Therefore, one possible method to deal with instance segmentation problem may be treated it as semantic segmentation problem. This train of thought has been discussed and proved in Kulikov, V, Yurchen, V and Lempitsky, V in their paper, Instance Segmentation by Deep Coloring in July, 2018 [6]. They provide a specific way to solve instance segmentation problem which also has been tested in this task and it performs satisfying in processing images of plants from the Plant Phenotyping Dataset.

The guiding ideology of this method is to reduce instance segmentation to semantic segmentation in order that it can be solved by the same method used to solve semantic segmentation problem. Experimenters make a rule of coloring each instance based on real process of coloring, which leads to the reduction of small semantic segmentation. After that they adopt standard end-to-end trained deep learning architectures to solve the problem.

This experiment has inspired us that it may be efficient to relate the instance segmentation problem to some existing solved image segmentation problem.

2. *Segmentation with SLIC superpixels*

Scharr, H, Minervini, M and their groupmates have collected several kinds of different methods which can successfully deal with leaf segmentation in plant phenotyping. [7] In their paper: Leaf segmentation in plant phenotyping: a collation study, April 2015, they introduced principle of each method, such as segmentation via 3D histograms, segmentation with SLIC superpixels and leaf segmentation with Chamfer matching. Each of these approaches is highly representative and provides a new way to analyze multi-instance segmentation problem.

Based on the method we applied in task3, segmentation with SLIC superpixels in this paper is relative to our method because they both apply watershed algorithm as a key process. The first result of our method is not satisfying and after that we discuss and intensively study the SLIC superpixels method, a improved version of our method is raised in this report.

3. Dataset and evaluation annotation

Finely-grained annotated datasets for image-based plant phenotyping, Minervini, M, Fischbach, A, Scharr, H and Tsaftaris, S has introduced the information about the image data in the Plant Phenotyping Dataset in detail including the importance of analyzing and tackling such data, the relative tasks in computer vision field, and also provided the principle of evaluation criteria: Best Dice and Symmetric Best Dice [8].

Those information is helpful in understanding the background of task3 and learning the new evaluation measure. During the experiment, the content of leaf detection and boundary estimation contributed to our preparation of this task, such as searching relative paper and deciding method of pre-processing.

4. Textbook about computer vision algorithms

Some professional books about computer vision algorithms and image processing methods also provide tremendous help for our work. One representative textbooks are Learning OpenCV 3: Computer Vision in C++ with the OpenCV library, Kaehler, A and Bradski, G, 2016 which collects the principle of watershed algorithm and its scope and also help in understanding image morphology, methods of which is used when preprocessing the original image [9].

III. METHODS

A. Task1

From what is mentioned above, we can conclude that there are two efficient methods to detect and localize required objects. For the HOG based SVM classifier, which is a traditional way, it requires a lot of hyperparameters when the dataset is being tested. In that case, it can be quite time-consuming. In addition, since the target scale is different, and there are multiple targets with various scales on one image, this traditional method is not that suitable [10].

As for the YOLO algorithm, it can play an important role in both small and big objects detecting. YOLO solves object detection as a regression problem, and the entire detection network pipeline is very simple. [11] The false detection rate of YOLO for background images is quite low. It has more than one type of network categories to choose. In this task, I decided to use YOLO v5s, since it is the fastest YOLO model and the dataset is not that complicated. Even though the depth and breadth of YOLO v5s are not that great, it is enough to detect this dataset.

In conclusion, I choose the method of YOLO v5 to implement this task.

B. Task2

1. Creating segmentation model

This task involves two stages:

(1) Segment the plant leaves from its background

Convert the BGR color space to HSV then use greenness identification will make the result more reliable. Then presume a greenness range to generate a binary image which will shows all the green pixels within the predetermined range.

(2) Optimize the segmented image

It can be observed from the sample photos that there is still some green particles in the soil, which is not part of the plant leaf, and those green particles will also be selected by the greenness identification. In order to removed those noise pixels from the result images, some algorithms such as erosion, dilation and median filter may be applied to the binary image to eliminate those noise pixels.

2. Evaluation

The finally generated binary image presents the leaves in white and its background in black. In order to examine and evaluate the result, expected images from the given dataset will be used to compare with the generated image by the segmentation model mentioned above.

There are two indices for this evaluation. One measure index is the Dice Similarity Coefficient (DSC). DSC is independently introduced by Sørensen (1948) [12] and Dice (1945) [13]. The equation to calculate DSC is as shown below:

$$\mathrm{DSC} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Where TP, FP and FN stand for True Positive, False Positive and False Negative respectively. In this task, TP is where the color spaces in the generated binary image and given binary image are both 255 at the same pixel location. FP is where the color space in the generated binary image is 255 but the given binary image is 0 at the same pixel location. FN is the opposite way to FP.

The other measure index is Intersection over Union (IoU), also known as Jaccard index (Jaccard, 1912) [16]. IoU index is not very different from the DSC index, and it is calculated as shown below:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

From the equation of DSC and IoU, it can be seen that the DSC will give higher score when TP is higher comparing with IoU, given the same datasets.

By scanning and comparing the generated binary image with the given binary image, FP, FP and FN can be found by counting the pixels that matches each vibrable condition. Then the DSC and IoU scores can be calculated.

C. Task3

Before starting designing our model to solve task3, we collect and analyze some literature about multi-instance segmentation. Most of relative methods apply deep learning and obtain a satisfying result to this problem. However, the method applied in our experiment is mostly about traditional computer vision algorithms due to our insufficient knowledge of deep learning. Meanwhile, some traditionary algorithms of image segmentation which perform well in normal segmentation problem may also be efficient in multi-instance segmentation problem if they are modified to some extent. For example, the MSU approach mentioned in Leaf segmentation in plant phenotyping: a collation study, April, 2015, has achieved this goal by using traditional thresholding, contrast stretching and rotating [7].

In task3, our method mainly contains two part of segmenting. The first part is distinguishing green plants from other object such as soil and flowerpot. To achieve the goals of this phase, threshold segmentation is applied. And the most necessary part which is segmenting leaves from plant, our method based on the traditional image segmentation algorithm: watershed. The reason why watershed algorithm is chosen is that considering the original images and the required outputs, the target of the method is to label portions of the foreground after segmenting the original image to background and foreground. It is typically usage of watershed algorithm. And also because watershed algorithm has been taught in previous lab activities.

Besides the algorithm of segmentation, other traditionary image processing algorithms also have been used in the method during the period of our experiment. For example, after the first testing, it is found that there are plenty of noise point existing after threshold segmentation thus the eroding and dilating from image morphology is used to remove those noise.

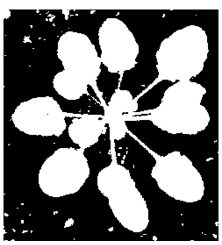


Fig. 4. *Example of image after segmenting green plant from the original image. The noise point can be seen all around the processed image.*

And for the evaluation, the introduction of task3 has required Symmetric Best Dice measure. This new criteria has been provided in Finely-grained annotated datasets for image-based plant phenotyping, Minervini, M, Fischbach, A, Scharr, H and Tsaftaris, S. According to the paper, Best Dice(BD) is defined as:

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

Where |•| represents the number of pixels which constitute leaf, and leaf segmentation L^a, L^b contains $1 \le i \le M$ and $1 \le j \le N$ two sets which is defined as L_i^a and L_j^b.

Symmetric Best Dice(SBD) is defined as:

$$SBD(L^{ar}, L^{gt}) = \min\{BD(L^{ar}, L^{gt}), BD(L^{gt}, L^{ar})\} \quad (1)$$

Where L^{ar} is the algorithm result and L^{gt} is the ground truth.

After finishing the task, it is required to use equation (1) to evaluate the method. It shows the specific methods in the experiment below.

1. Pre-processing

The first part of our method is aiming at segmenting the plant from the original image. After reading the image, use median filter to smooth the image. And convert the color space of image from BGR to HSV in order to obtain the mask of the plant part. Using thresholding function, in this experiment inRange function has been applied, to generate a mask of the green area of the original image. After obtaining the mask, do AND operation by the mask then convert the color space of the result image from BGR to GRAY in order to prepare for binary segmentation. In this experiment, adaptive thresholding is applied at the same time for a better binary thresholding segmentation result. The foreground of the output image is the plant.

However, during the experiment there are always noise pixels in the output image of pre-processing. Median filter performs unsatisfying in this process. After testing different methods of removing noise, the eroding and dilating from image morphology are used to removed noise. And combine it with median filter to improve the quality of the output image.

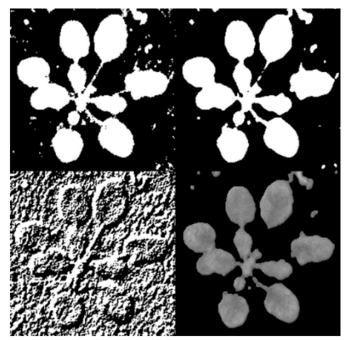


Fig. 5. *Different results of applying different methods of removing noise.*

2. Apply watershed algorithm to segment leaf

This portion is trying to segment leaves from the binary plant image. Firstly, computing the distance which is needed in watershed algorithm. In this experiment it is achieved by threshold function in OpenCV library. The traditionary watershed algorithm defined distance as pixels to the background. Next, applying findContours and drawContours functions in OpenCV library to generate markers which should be provided when applying watershed algorithm. Then using watershed algorithm to do leaf segmentation.

However, this method has a bad performance, it output images of a large mass of adhesions.

3. Optimizing the previous method

Pre-processing is the same as section A. In the application of watershed algorithm, adopting peak_local_max function in skimage to obtain local maximum value as markers of watershed algorithm instead of using findContours and drawContours. Other process is also same to previous method.

The result of this modified method basically achieved leaf segmentation. However, the result leaf always is connected with stem. Therefore, the performance of this method is unsatisfying.

IV. EXPERIMENTAL SETUPS

A. Task1

1. Preprocess of datasets

It is obvious that the dataset includes 70 PNG files and 70 csv files. We selected 60 picture samples randomly as train set and the left 10 samples as test set. Since the original markup files are stored as csv files which don't match with the requirements of YOLO v5, I designed a preprocessing program named [14] generate_yolo_label.py. In that case, each line can be a box and the format of data is: class_id, x, y, w, h, which can refer to the [14] official material.

In addition, I use file named data to store original dataset and new dataset with format of YOLO v5. And there are also some utility functions in common_util.py.

2. Train model

Then we select YOLO v5s which is the smallest model to train the generated datasets, and the related python file is named [15] train.py, which refers to material in GitHub. Then set related parameters and execute the train model. Because of the special data augmentation step of YOLO v5, we can get some examples which are augmented.

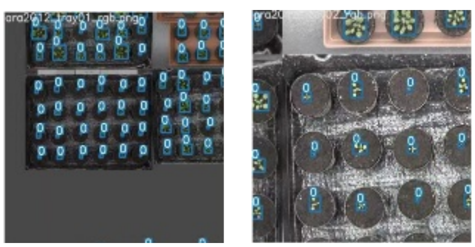


Fig. 6. *Augmented Data*

3. Test model

Since we have built a new dataset and stored it in file named val, YOLO v5 will test this dataset automatically. And the evaluation results like mAP will be stored finally. We use a python file called [15] detect.py to execute the visualization prediction. And we can get the final pictures (shown below) with bounding boxes on each object.

Fig. 7. *Images without bounding boxes*

B. Task2

This task uses Python3 to perform the segmentation algorithms and evaluation methods mentioned in the methodology. OpenCV in Python is a powerful module. It has many existing functions which can segment the target object from its background and optimize the result image. This experiment can be divided into following steps:

STEP 1. Use OpenCV to read the original image and convert the image into 3 channel BGR color space. Then covert the BGR color space to HSV color space.

STEP 2. Define the upper and lower boundaries of the GREEN color. In this experiment, after many tests, the upper bound of greenness is [36, 100, 50] and the lower bound of greenness is [86, 255, 255].

STEP 3. Use OpenCV inbuild function inRange() to filter out those pixels whose color space is not 'Green', and generate a binary image.

STEP 4. Use erode(), dilate() and then medianBlur() functions to remove the noise pixels.

STEP 5. Compare the results with the given binary image. Evaluate the result using DSC and IOU measures. Run this segmentation model on the give datasets and return the average score.

C. Task3

1. Experimental Setup

In this experiment, Python libraries including os, numpy, torch, matplotlib, skimage, sys, random and math are used. The operating system is Mac OS.

2. Dataset

In the Plant Phenotyping Dataset which has been used in the experiment consists of individual plant images, 120 from Ara2012, 165 from Ara2013 (Canon), and 62 from Tobacco.

3. Manual Parameter Settings

In the method, the upper and lower boundaries of thresholding is alterable and their value is settled randomly. Also the kernel sizes of median filter and eroding are decided by repeated testing and sizes with better performances are chosen.

TABLE I. ARCHITECTURE MANUAL PARAMETER SETTINGS

function	value	explanation
function init in class DownModule	1	repeats
	0	padding
	3	kernel size in Conv2d
	0.1	probability of dropout
	2	kernel size in Maxpool2d
	2	padding in Maxpool2d

function init in class UpModule	1	repeats
	0	padding
	2	kernel size in ConvTranspose2d
	3	kernel size in Conv2d
fuction forward in class UpModule	1	the direction of connection
function init in class EUnet	1	width coefficient
	1	the number of repeats in encoder part
	1	the number of repeats in decoder part
	3	depth
	8	base
	0	padding
	3	kernel size of conv
	1	kernel size of conv1x1

Table I shows the manual parameter settings in document architecture.py.

TABLE II. DATA MANUAL PARAMETER SETTINGS

function	value	explanation
fuction init in class Reader	1	the direction of connection
function create batch generator	3	batch size

Table II shows the manual parameter settings in document data.py.

TABLE III. UTILS PARAMETER SETTINGS

function	value	explanation
flip horizontally	0.5	probability
flip vertically	0.5	probability
rotate90	0.5	probability
resize in function rescale	0	cval
random scale	0.2	scale variance
random noise	0.5	probability
	0.001	gain random
blur	1	sigma
	0.5	probability
	0.1	gain random
random contrast	0.2	low digits
	0.8	high digits
	0.1	gain random
random gamma	0.4	gamma
	0.9	gain
	0.1	gain random
	0.5	probability
visualize	40	minimum point

Table III shows the manual parameter settings in document utils.py.

TABLE IV. HALO LOSS MANUAL PARAMETER SETTINGS

function	value	explanation
build halo mask	30	fixed depth
flip vertically	21	margin
rotate90	10	minimum fragment

Table IV shows the manual parameter settings in document halo_loss.py.

V. RESULTS AND DISCUSSION

A. Task1

1. Average precision

The graph shows that when epochs = 300, threshold = 0.5, the value of AP reaches the best situation (0.997).

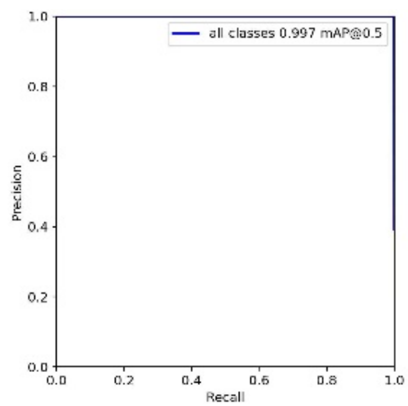


Fig. 8. *Average Precision*

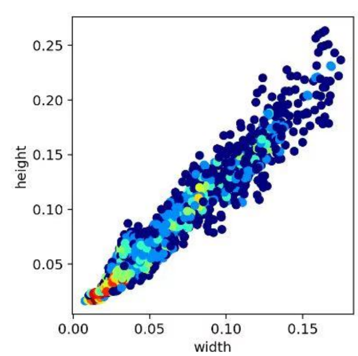


Fig. 9. *Distribution of bounding boxes*

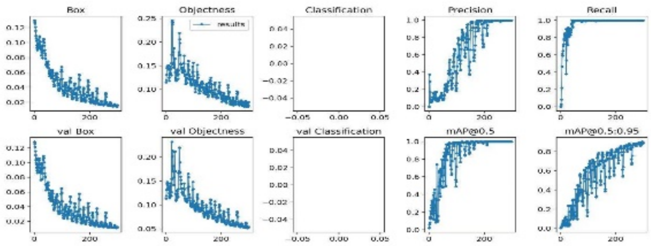


Fig. 10. *Changes of loss and indexes*

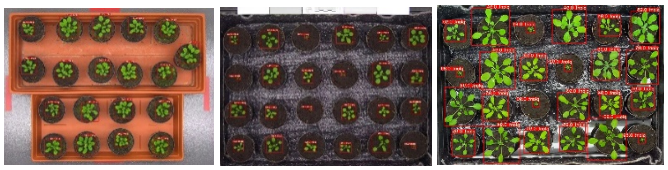


Fig. 11. *Examples of testing result*

B. Task2

There are two datasets files ‘Ara2012’ and ‘Ara2013-canon’, whose plant condition is different. By visual observation, the plants in file ‘Ara2013-canon’ are relatively smaller than those in file ‘Ara2012’. The segmentation model ran on these two files are satisfiable. And the accuracy increases in each noise cancelling step, shown in Figure 12. Only when applying dilate() function after applying erode() function, the accuracy drops, shown in Figure 13, so this dilate() function is abandoned in the segmentation model.

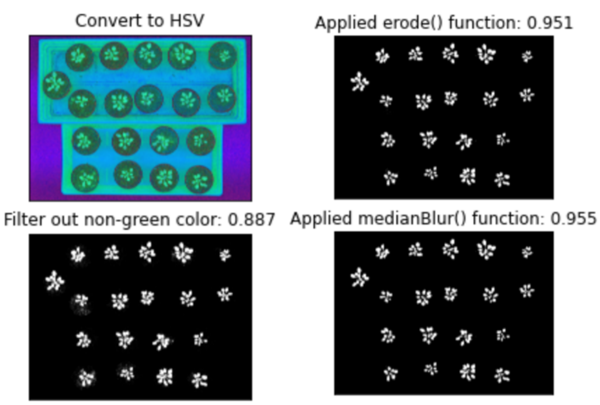


Fig. 12. *Improved accuracy after each application of noise remove*

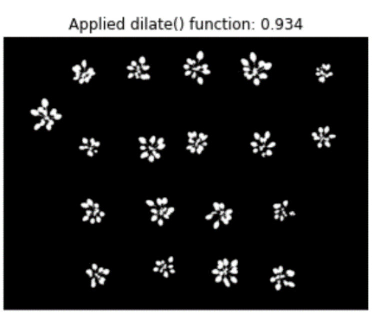


Fig. 13. *Accuracy drops after applying the dilate function*

After running the segmentation model on both files, the mean accuracy is then calculated for each file. Mean accuracy of file 'Ara2012' by DSC measure is: 96.0%. Mean accuracy of file 'Ara2012' by IoU measure is: 92.3%. Mean accuracy of file 'Ara2013-Canon' using DSC measure is: 94.6%. Mean accuracy of file 'Ara2013-Canon' using IoU measure is: 89.9%.

C. Task3

1. Result of the initial method

The first method performs badly in instance-segmentation. Several different leaves and stems usually have the same label in the result images.

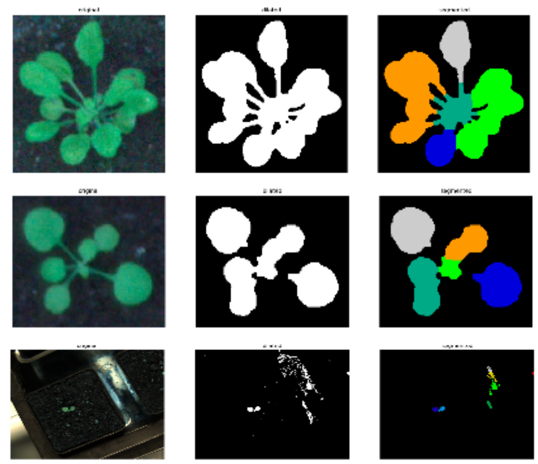
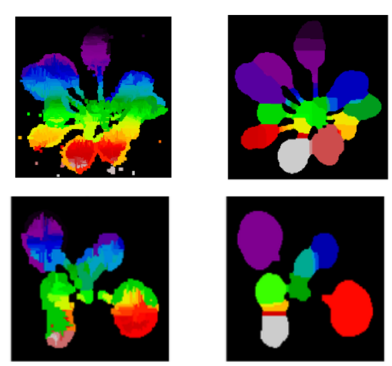


Fig. 14. *Result images from the first method*

The reason of bad performance may be that the information in receptive field of adjacent pixels is similar, when these adjacent pixels distributed at the boundary of the area of single leaf. It may cause reduction of the segmentation performance. Therefore, this method is more suitable for semantic segmentation problem instead of instance segmentation problem.

2. Result of the optimized method

Although the optimized method performs better than the initial method, it cannot segment leaf from stem. The leaf in the result image is connected with its stem.



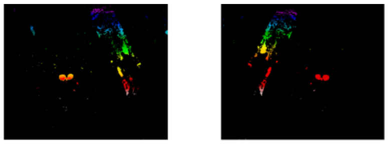


Fig. 15. *Result images from the first method*

The result images have the problem of over-segmentation. It is caused by noise from original images and imperceptible pixel value change of objects. Besides removing noise, the optimized method also applied limiting the gradient image by thresholding in order to reduce the over-segmentation by change of pixel value. The result is improved. However, the over-segmentation still exists.

3. *One thought about segmenting leaf without stem*

Based on traditionary image segmentation algorithm, extracting foreground is the key process which directly related to the quality of segmentation result. Therefore, it is possible to segment leaf without stem if the method can successfully extract a satisfying foreground which means only leaves from plants. However, traditionary watershed algorithm cannot do this work because it will mark the same label to adherent objects and treat them as one object. It is a shortcoming of traditionary watershed algorithm.

Therefore, the key point to solve this problem is changing the traditionary watershed algorithm and make it can segment leaf without stem. The distance map of traditionary watershed algorithm is defined as the distance from the current pixel to the background pixels, which is the root cause of the disability of segmenting leaf without stem.

Our expectation is that the stem of plants is not including in the distance map of the watershed algorithm. It has been found that comparing the stem and leaf of plants, leaf area tends to have more pixels. Therefore, one possible approach is using a specific size of window to retrieve the image, and compute the percent of foreground pixels in the window in order to distinguish stem and leaf. Equation (2) can be used to compute the label for this area to achieve leaf segmentation.

$$current = \frac{r}{n} * 255 \tag{2}$$

Therefore, instance segmentation may be accomplished by changing the definition of distance map thereby changing the local maximum and result of segmentation.

3. *Evaluating the performance using Symmetric Best Dice*

TABLE V. SYMMETRIC BEST DICE OF THREE DATASETS

	Ara2012	Ara2013-Canon	Tobacco
SBD	62.5%	66.4%	35.5%

The table above shows the average symmetric best dice on the three datasets. The images in Ara2012 and Ara2013-Canon have very obvious green features to extract, which results in better performance than Tobacco. The plants in Tobacco dataset are very small, so it is hard to generate foreground of the images, which brings about bad performance on segmentation.

VI. CONCLUSION

A. *Task1*

After completing this task, we can conclude that YOLO algorithms are much easier to execute and can get much better evaluation standards like precision and recall. And YOLO v5 is quite a time-saving and cost-effective method. However, YOLO v5 still got some disadvantages. It is not that accurate like YOLO v4 and it needs to be further refined.

B. *Task2*

In conclusion, using greenness identification to segment the plant from the background is reliable, with accuracy higher than 95%. When the plant is relatively small, for example in Figure 16, some plant leave is difficult to identify visually. When applying noise removal methods on this photo, some leaf parts may also be filtered out because these parts are too small and was treated as noise, and the result image only has 88.4% accuracy.

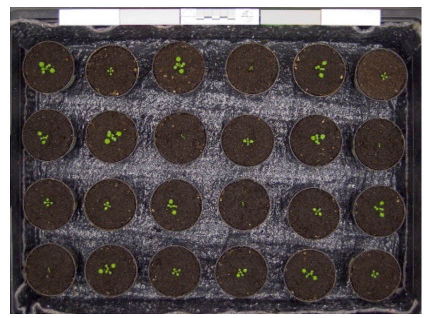


Fig. 16. *Original image before applying segmentation algorithm*

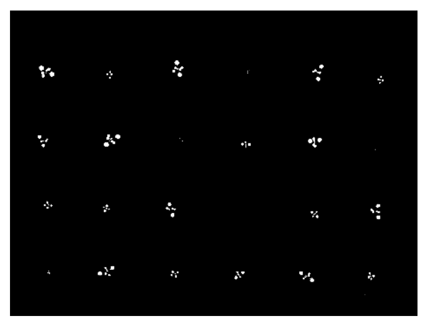


Fig. 17. *Image after applying segmentation algorithm*

To improve the algorithm, a box or a shape can be generate first to cover the area only of the plant leave, this will reduce the noise pixels dramatically. Another way is to try a different greenness color range, which can keep more parts of the plant leaves, but it may cause more noise pixels.

C. *Task3*

It is difficult to finish multi-instance segmentation by using traditionary image segmentation methods due to the limitation in their definition. Approaches which are effective in multi-instance segmentation problem are roughly divided into applications of deep learning methods and optimizing traditionary algorithm to fit the multi-instance segmentation problem.

During our testing, it seems that the application of deep learning perform better than the optimized traditionary algorithm.

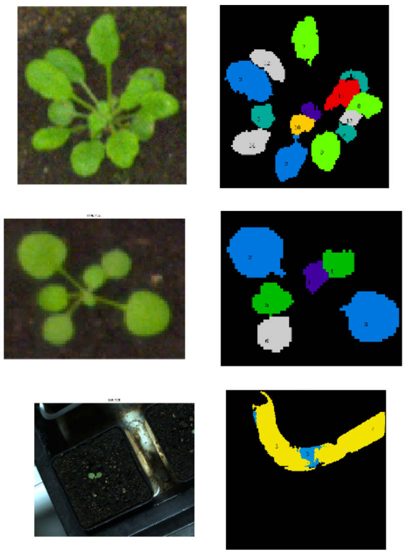


Fig. 18. *Result images from the deep coloring method*

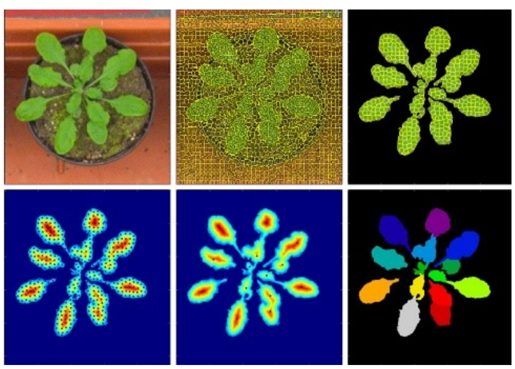


Fig. 19. *Processing images of segmentation with SLIC superpixels in Leaf segmentation in plant phenotyping: a collation study, Scharr, H, Minervini, M, French, A, etc.April, 2015 [7].*

Figure 18 shows the result of deep coloring which is the application of deep learning. Figure 19 is the sample image from Leaf segmentation in plant phenotyping: a collation study, Scharr, H, Minervini, M, French, A, etc.April, 2015, shows the result images of a modified traditionary segmentation algorithm. The leaf after being segmented remains connection with the stem. However, the leaf portion in result images of deep coloring method is perfectly segmented.

VII. CONTRIBUTION OF GROUP MEMBERS

We have five members in our group, our group contribution can be illustrated below.

For Wenjing Gao, she implemented the code and report of task3. And made the introduction and conclusion part in the online presentation.

For Han Chen, he completed task3 and integrated report of it. And made a brief demonstration of task3 in the presentation part.

For Liqi Jiang, she completed components of task3 and introduced it in group presentation. In addition, she modified the group report at last.

For Yang Liu, he implemented the code and classification of task2 as well as report of it by himself and explained task2 in presentation.

For Yilin Liu, she completed the code and report of task1 and made a brief presentation on it.

REFERENCES

[1] W. Huang, D.A. Ratkowsky, C. Hui, P. Wang, J. Su, P. Shi Leaf fresh weight versus dry weight: which is better for describing the scaling relationship between leaf biomass and leaf area for broad-leaved plants? Forests, 10 (2019), p256

[2] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE, 2005, 1: 886-893.

[3] Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines[J]. 1998.

[4] Yap M H, Hachiuma R, Alavi A, et al. Deep Learning in Diabetic Foot Ulcers Detection: A Comprehensive Evaluation[J]. arXiv preprint arXiv:2010.03341, 2020.

[5] Jose M, Miguel A, Juan A, Juan M. Detecting skin in face recognition systems: a colour spaces study. Digital Signal Process 2010;20(3):806–23.

[6] V. Kulikov, V. Yurchen, and V. Lempitsky, “Instance Segmentation by Deep Coloring,” July 2018.

[7] H. Sharr, M. Minervini, A. P. French etc, “Leaf segmentation in plant phenotyping: a collation study,” Springer-Verlag Berlin Heidelberg, April, 2015.

[8] M. Minervini, A. Fischbach, H. Scharr and S. A. Tsaftaris,“Finely-grained annotated datasets for image-based plant phenotyping,” Elsevier Ltd. 2015.

[9] Adrian Kaehler and Gary Bradski, Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, O’Reilly Media, 2016.

[10] Han F, Shan Y, Cekander R, et al. A two-stage approach to people and vehicle detection with hog-based svm[C]//Performance Metrics for Intelligent Systems 2006 Workshop. 2006: 133-140.

[11] Tian Y, Yang G, Wang Z, et al. Apple detection during different growth stages in orchards using the improved YOLO-V3 model[J]. Computers and electronics in agriculture, 2019, 157: 417-426.

[12] Sørensen, T. (1948). “A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons". Kongelige Danske Videnskabernes Selskab. 5 (4): 1–34.

[13] L.R, Dice (1945). "Measures of the Amount of Ecologic Association Between Species". Ecology. 26 (3): 297–302.

[14] GitHub. 2020. Ultralytics/Yolov5. [online] Available at: <https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data> [Accessed 16 November 2020].

[15] GitHub. 2020. Ultralytics/Yolov5. [online] Available at: <https://github.com/ultralytics/yolov5> [Accessed 16 November 2020].

[16] Jaccard, Paul (1912), "The Distribution of the flora in the alpine zone", New Phytologist, 11 (2): 37–50

[17] Joe Minchino and Joseph Howse, Learning OpenCV 3 Computer Vision with Python, 2nd ed., Packt Publishing, 2015.