

Early Results in the Grading of Vegetative Cuttings using Computer Vision

Qiang Ji and Sanjiv Singh

CMU-RI-TR-96-22

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

May 1996

© 1997 Carnegie Mellon University

The work reported was sponsored by Oglevee Ltd.

Abstract

Commercial vegetative propagation of floricultural crops requires the segregation of plant cuttings into categories based on size. The cuttings however must be graded when they are planted (“stuck”), at which time the grade of a cutting is not easy to determine.

This report describes a system that learns to classify cuttings from being shown examples of images of cuttings that have been graded by a human expert. Based on the example set, the system learns to grade cuttings into categories.

We report the results based on a set of 150 geranium plants that were graded by our system and compare the results to the performance of an expert grader.

1. Introduction

An important procedure in vegetative propagation of floriculture crops is to grade harvested vegetative cuttings into different categories according to market standards. Currently, the grading process is carried out manually by trained human graders. Human graders visually inspect each vegetative cutting to estimate its physical measurements like stem caliber, length, leaf surface area, weight, etc. Vegetative cuttings are subsequently classified into one of the three to five classes based on estimation of these physical parameters.

Manual grading is labor intensive and expensive due to the cost of skilled labor required. In addition, precision varies greatly from one grader to another. Automation offers the opportunity to reduce production costs and improve quality.

Several commercial systems are now available that grade plant cuttings based on their weight. Also, a recent commercial system from Holland grades cuttings based on the leaf area as measured by a machine vision system. While the first approach is simple, it is not accurate since the weight of a vegetative cutting may not correlate well with its grade. The second approach also presents accuracy problem. Placed on a conveyor belt, the leaves may be folded and occluded. It would therefore be difficult to accurately estimate the leaf area from a single view.

Our project is developing a system that learns to grade vegetative cuttings based on features measured by computer vision and grades assigned by a human grader. Computer vision offers an effective means to automate the grading process since it can make full use of all the visual clues used by human operators only at much faster speed. Unlike the Dutch vision system, the system we propose will not only make use of all visual features employed by human graders but also exploit other available visual features. The computer vision system recognizes a vegetative cutting transported on a conveyor and then computes its properties. A classification system then classifies the cutting based on its property measurements from the vision system.

This report describes the preliminary results of our efforts in developing an automated vision system for analyzing and grading stationary geranium cuttings. Efforts described in this report are a preamble to a subsequent design and implementation of a commercial robotics vision system for automatically grading geranium cuttings transported on a conveying production line. The geranium cuttings described in this report are limited to a variety known as *veronica*. The techniques however are also applicable to other varieties. Early results indicate that the classification performance of the machine classification system is comparable to that of human graders.

This report divided into six sections. Section 2 discusses the process of acquiring images. Low level processing of the images is covered in Sections 3 and 4. Section 3 discusses the process of segmenting the images into background and plant. Section 4 discusses the techniques for segmenting the plant image into various parts. Extracting and computing the primary characterizing features of each component are dealt with in section 5. Section 6 deals with the issues of the classification. The report concludes with a summary of the machine classification system and a discussion of the future directions.

2. Image Acquisition

The project described in this report has concentrated on analyzing stationary images of manually handled vegetative cuttings. By stationary and manual handling, we mean vegetative cuttings to be studied are manually placed on a stationary platform as opposed to being automatically placed a moving conveyor. The stationary platform consists of a base table and a vertically movable camera holder. During image acquisition, vegetative cuttings are manually placed on the base table one at a time. A camera is mounted on the movable holder, aiming vertically downward at the vegetative cuttings. The distance from the camera to the table is adjustable, depending on plant size, magnification, and resolution required.

To enhance the contrast between vegetative cuttings and the background, a base table with uniform white surface finishing was selected. This is because the vegetative cuttings tend to look dark in the acquired gray scale image. The enhanced contrast can greatly facilitate subsequent image segmentation through thresholding. To minimize possible shadows and light reflection, additional lighting and photographic equipment were employed. Two photographic umbrellas were used to generate diffuse light to prevent presence of shadows. Caution was also exercised to experimentally adjust lighting strength, direction, and distance to reduce the light reflection from the white table onto the plants.

During image acquisition, vegetative cuttings were manually placed on the base table within the field of view of the camera one at a time. A Panasonic super VHS camera was used to acquire its image. The camera settings were set at f-stop 4-4.8 and magnification at 8x. Our experiments reveal that to better distinguish different grades of plants, high camera magnification and/or high resolution camera should be employed. Increasing the feature measurement distinctions among different grades of plants is important since if the precision of the vision algorithm is comparable to the measurement differences of plants, classification program would have difficulty distinguishing different grades of plants. The acquired images were subsequently digitized using the imaging utilities from the Silicon Graphic work station. The resulting digital images are 640x486, with 256 gray scales, stored in PGM format. Figure 1 shows images of two different categories (veronica and sincerity) of geranium cuttings.

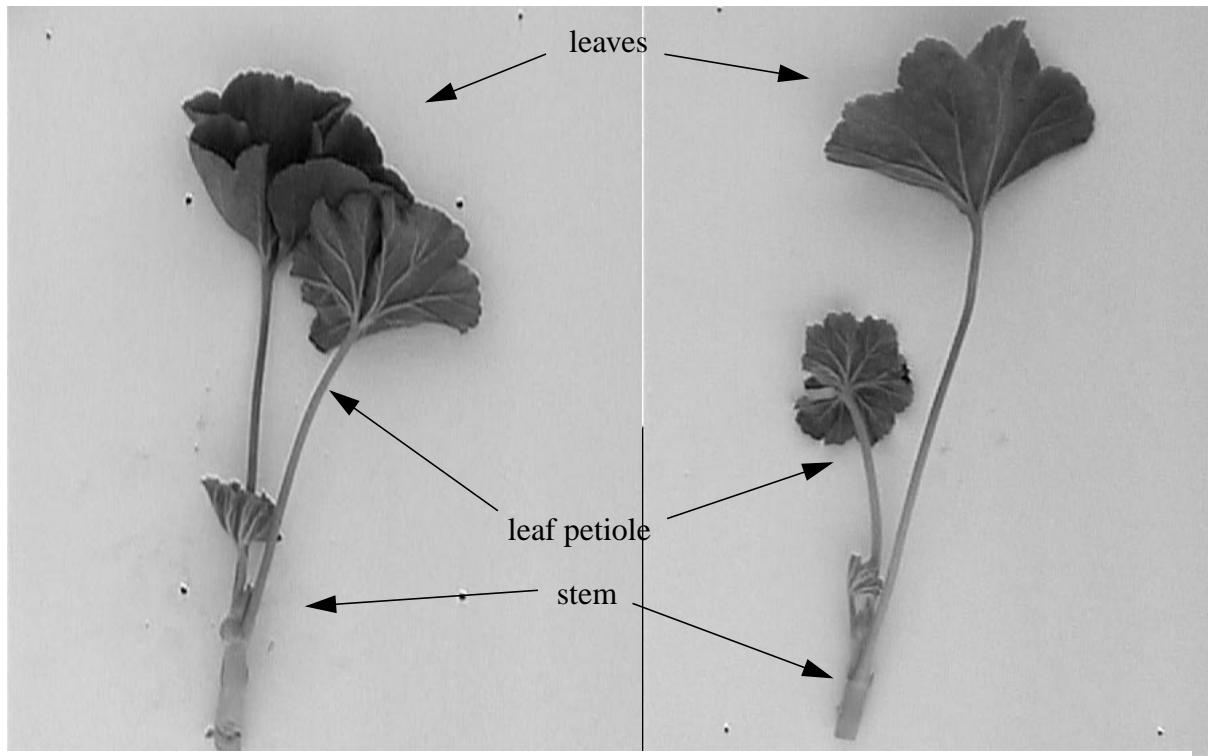


Figure 1 Digital images of two varieties of geranium cuttings (a) Veronica (b) Sincerity.

3. Primary Segmentation of Plant Image

The very first task is to distinguish the plant from the background that is we would like to uniquely identify the parts of the image that correspond to the plant cutting. This section examines various issues involved in this process.

3.1. Global Thresholding

To recognize the vegetative cuttings from their digital images, regions that contain vegetative cuttings need to be identified and separated from their background. This can usually be accomplished through thresholding if the premise that gray levels for pixels of vegetative cuttings are sufficiently different from those of background holds true. This assumption is basically true for this project since gray levels for vegetative cuttings tend to be darker than those of background. Thresholding converts an input gray scale image into a binary image, with white pixels representing vegetative cuttings and black pixels background.

To automatically threshold an input gray scale image requires automatic determination of an appropriate threshold value. An appropriate threshold value is crucial since a too high threshold may incorrectly label background pixels as vegetative cuttings while a too low thresholding may label part of vegetative cuttings as background. In either case, errors would be introduced to subsequent feature measurements. Based on the underlying intensities distributions, various techniques have been developed for automatic selection of an optimal threshold value [1]. Employing different specialized principles, these algorithms usually require that the intensity distribution of the image be bimodal. The relative high contrast between vegetative cuttings and the background allows most of these thresholding algorithms applicable to our problem.

Among these algorithms, we have used the algorithm developed by Otsu [4] due to its simplicity in implementation and fast execution speed. According to this algorithm, given a threshold, it divides the intensity distribution into two groups, representing the object and background respectively. Means and variances are then computed for each group. The sum of the two variances, weighted by the probabilities of each group, is the criterion function that needs to be minimized in order to identify the best threshold value. So the best threshold yields the smallest weighted sum of within-group variance. Given this criterion function, this techniques searches for all possible threshold values to locate the threshold that minimizes the sum of variance. In actual implementation, a recursive procedure was implemented to use results computed for the previous threshold value for next threshold value. This can result in substantial saving in computation time. Figure 2 shows the binary image for the veronica cutting shown in Figure 1(a), resulting from using above thresholding technique.

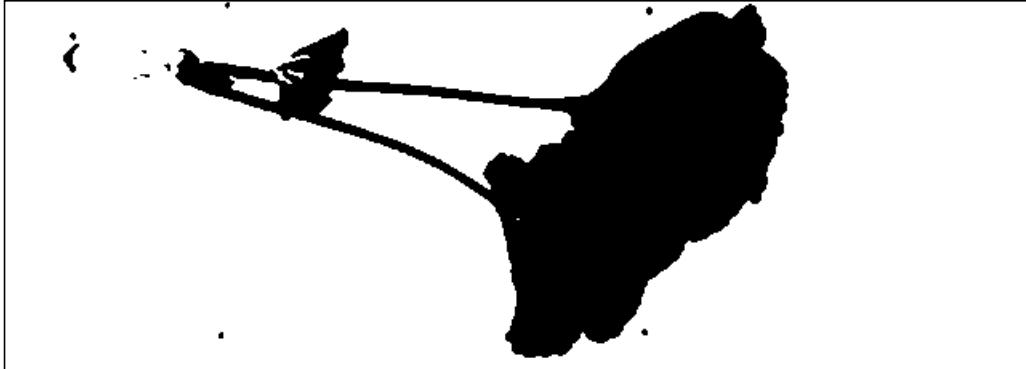


Figure 2 The binary image for the veronica cutting shown in Fig. 1(a) after global thresholding.

3.2. Adaptive Thresholding

Despite our efforts during image acquisition to increase the contrast between background and plants and to reduce the shadows, we still can not completely eliminate shadows and generate a high contrast throughout the vegetative cutting. This implies a global thresholding, which does not account for local contrast variation, may not always yield desired results as shown in Figure 2. Compared with the original image, that most part of stem was misclassified as background. On the other hand, some of the shadow near the leaves was misclassified as leaves. This would result in incorrect identification of stem and leaves. In addition, the mis-classification of part of stem as background may in some cases result in tiny connections between different parts of stem. This could pose a serious problem for subsequent stem identification. Thus, an adaptive thresholding technique was implemented. The adaptive threshold can account for the spatial variations in image contrast and compensate for a variety of local spatial context effects by using a spatially varying threshold.

The adaptive thresholding we implemented is a modified version of the technique developed by Chow and Kaneko [1]. The technique starts with dividing an entire image into mutually exclusive blocks. Based on experimentation, the optimal block size for our project was found to be 80x54. It assumes that the intensity distributions for block regions consisting of only the object or background are unimodal and that distributions for regions containing both object and background are bimodal. Furthermore, the bimodal distributions can be modeled as the sum of two Gaussian distributions. The variances and means are subsequently computed for each block. The histograms of blocks with large variance are then computed.

For each selected histogram, a bimodal test is conducted to determine whether the distribution is bimodal or unimodal. The criteria for bimodality test include difference in means and ratio of variance. For every histogram with appreciable bimodal distribution, the Ostu technique is used to compute an optimal threshold. For regions that are not assigned thresholds, their thresholds are subsequently estimated as weighted averages of the computed thresholds of their neighboring regions. Finally, a binary decision is performed for each image block using the thresholds obtained.

While adaptive thresholding can take the local contrast into consideration, it is so sensitive to small local spatial variations in intensities that it may classify certain details of plants as background, especially in leaf area as shown in Figure 3.

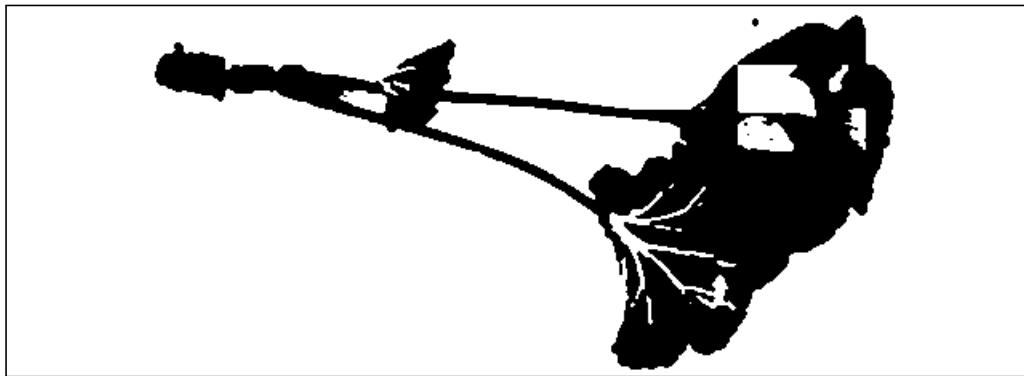


Figure 3 Undesired effects in the binary image from adaptive thresholding for the veronica cutting shown in figure 1(a).

This is undesirable for we do not want to split a plant into different parts. To avoid this, the algorithm was modified as follows. A global threshold is obtained first. For blocks with mean intensities lower than global threshold (this is mostly in leaf area), the global threshold is used for thresholding. For blocks with mean intensities higher than global intensities (mostly this is in stem region), adaptive thresholding is used. Figure 4 shows a binary image resulting from the modified adaptive thresholding technique.

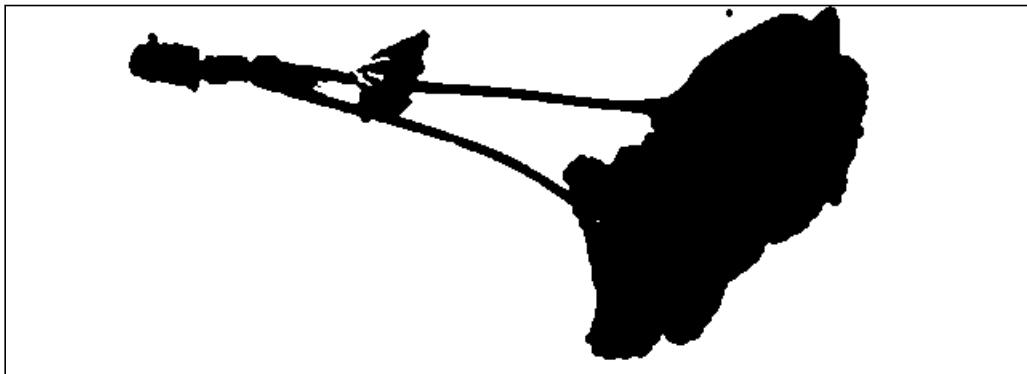


Figure 4 Binary image resulting from the modified adaptive thresholding for the veronica cutting shown in figure 1(a).

4. Image Segmentation

Once we recognize a vegetative cutting and separate it from its background, our next task is to identify the primary components of a vegetative cutting. Generally speaking, a geranium plant consists of a stem, leaves, and leaf petioles. The task of segmentation is therefore to segment the plant blob we obtained from thresholding into three distinct smaller blobs representing leaves, leaf petioles, and stem respectively.

Various techniques have been developed to perform segmentation. For example, Peleg [3] suggested use of curvature of boundary edge pixels for segmenting a plant image into distinct blobs. The basic premise of this method is that each blob can be delimited by two edge points with maximal curvatures. For example, the two intersection points between a stem and leaf petioles have curvatures higher than those of the rest of edges points on the stem and therefore can be used to delimit the stem from leaf petioles. The problem with this technique is that it is time consuming (requiring to identify the boundary of the plant and compute the curvature of each boundary point) and sensitive to noises and local irregularities.

Instead, we observed that there exist substantial morphological differences between leaves, leaf petioles, and those of stems. For example, the stem caliber is usually three to five times larger than those of leaf petioles while leaf widths are normally at least twice as big as those of stems. Based on these observations, a mathematical morphological algorithm was developed for segmentation.

The algorithm consists of three steps. First, it separates stem and leaves from leaf petioles. This can be achieved by removing leaf petioles. Based on their differences in size, the leaf petioles can be removed through a morphological opening operation. A square disk structure element was designed. The width of the structure element was selected such that it is slightly larger than the diameters of leaf petioles while much smaller than those of stem and leaves. As a result, a single opening operation removes leaf petioles from the original binary plant image. Figure 5 shows the result of the morphological opening operation. It contains only leaf and stem blobs. Here the leaf petiole blob has been effectively removed.

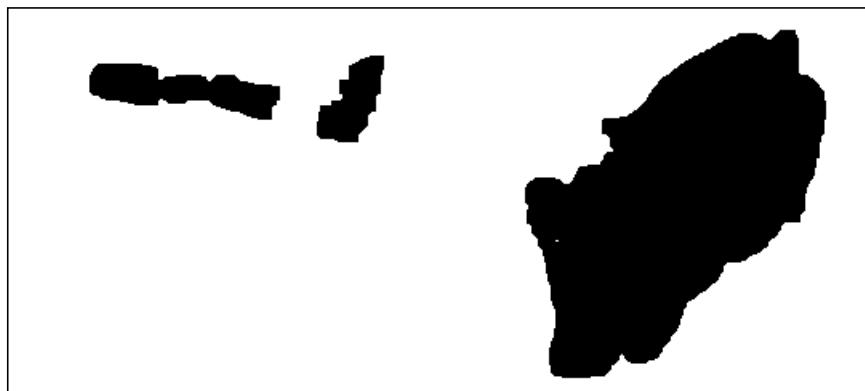


Figure 5 Removal of the leaf petioles from the veronica cutting through morphological opening.

Since this operation requires prior knowledge of relative size of leaf petiole diameter, an alternative is to open with a small structure element (3x3) in several iterations. The advantage is that it does not require prior knowledge of leaf petiole diameter size and that it can minimize the contour and shape change of leaves and stem as discussed below. The disadvantage is that it requires several iterations. Since opening is a rather time consuming operation and time is critical for us. We decided to use the first method since given a relatively fixed image acquisition environment, the camera parameters remain constant. So we can always easily estimate the approximate size of leaf petiole diameter.

The resulting blobs consisting of leaves and stem tend to swell in perimeter due to the opening operation as shown in Figure 5. This is because an opening operation consists of an erosion operation, followed by a dilation operation. While the erosion operation removes the leaf petioles. It has minimal impacts on the stem and leaves blobs. The subsequent dilation operation, however, smooths the contour, breaks narrow isthmuses, and eliminates small islands of stem and leaves. The expansion in blob perimeter must be minimized so that stem and leaves can retain their original shape and contours. It ensures the accuracy of the measurements of the characteristic features of each blob. To do so, the resulting image from the opening was AND with the original binary image, generating a new image that contains only leaves and stem with their original shape and contours. Figure 6 shows the resulting image. Clearly, compared with Figure 5, the original shape and contours of stem and leaves have been kept intact.



Figure 6 The leaf and stem blobs resulting from an AND operation to minimize the change in blob contours from opening.

The second step is to identify the leaf petioles. This can be easily obtained by subtracting the above leaves and stem blobs from the original blob image. The resulting blobs contain only leaf petioles as shown in Figure 7

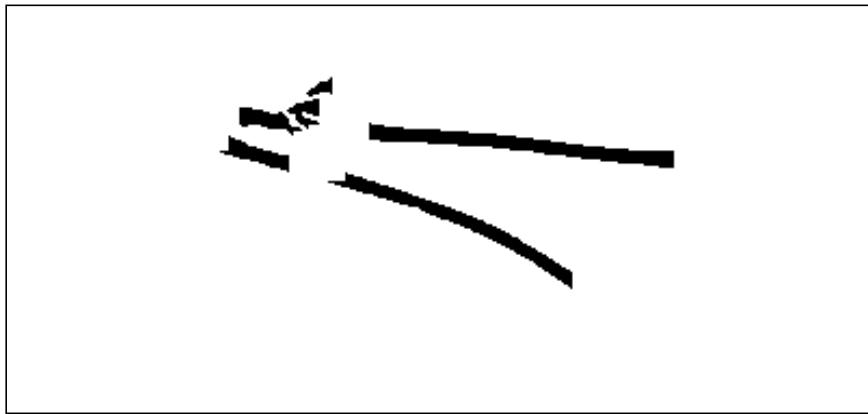


Figure 7 The extracted leaf petiole blobs.

The last step is to separate the stem and leaves. Distinguishing between leaves and stems is made based on their inherent physical and geometric differences like shape and size, difference in mean intensity, and relative location in the image. Geometrically, stems tend to be of a rectangular shape while leaves can assume any shape, although most likely fan-shaped. Physically, stems are usually smaller in area than leaves. Intensity-wise, stems tend to be brighter than leaves. It is these differences plus any prior knowledge about the location of stem and leaves that are used for separating leaf blob from stem blob. Figure 8 shows the resulting image; Figure 8(a) is the stem blob while Figure 8(b) is the leaf blob.

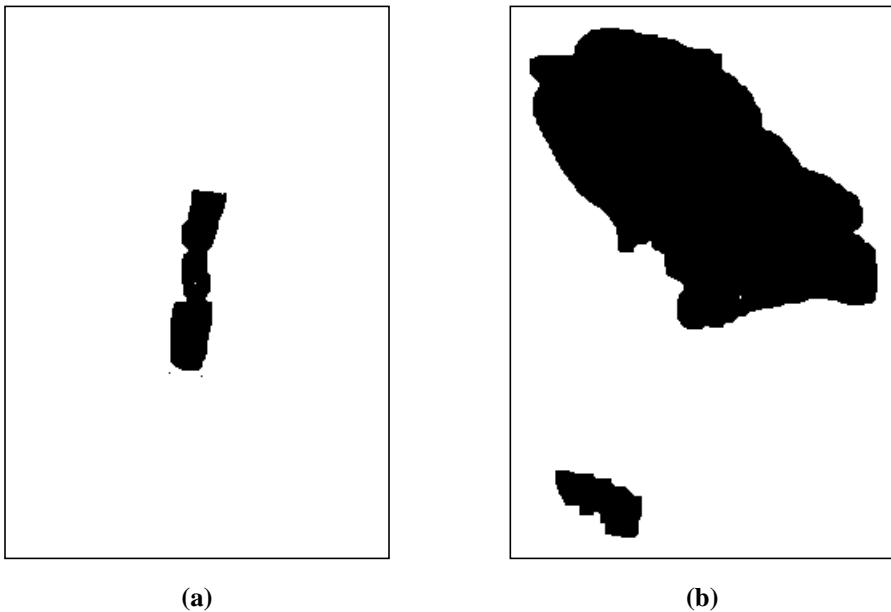


Figure 8 The extracted stem (a) and leaf (b) blobs.

5. Blob Properties Computation

Now that we have successfully segmented original images into three meaningful connected components: stem, leaf petioles, and leaves, it is necessary to compute the primary characterizing features of each blob. The primary characterizing features of each blob will be used for classification. Based on what we learn from human graders, the primary characterizing features for each blob are defined as follows. For stem blob, the primary characterizing features are its length and its caliber since they are the most important criteria used by human operator during grading. For leaf blob, the primary characterizing feature is its area. It approximates the leaf surface area. This feature is, however, ill-defined since we can not always accurately estimate the leaf area just from one view point. Leaves, when placed on a table, may be folded or occluded by each other. Therefore, even for the same plant, its leaf area may vary considerably, depending on the camera angle.

Finally, for leaf petiole blob, the primary characterizing features can be the number of leaf petioles, the diameter of each leaf petiole, and the length of each leaf petiole. The number of leaf petioles is important since it indirectly gives us the number of leaves. During human grading, the number of leaves is often used as a tie-breaking factor when stem length and caliber contradict each other. The diameter of each leaf petiole is a feature not used by human grader. But we feel that leaf diameter, especially the mean diameter of all petioles, may somehow correlate with leaf area. In fact, intuitively, there seems to be a positive correlation between them, i.e. the larger leaf surface, the larger the mean leaf petiole diameter. The potential use of this correlation is that we may use mean leaf petiole diameter as a replacement for leaf area since mean diameter of leaf petiole can be estimated much more accurately than leaf area. We will examine the correlation during the classification phase. Finally, length of leaf petioles do not seem to play any role in human judgment of plant grade. It is therefore not included as the primary characterizing feature of leaf petioles.

5.1. Computation of stem length and caliber

In this section, we will discuss the technique we developed for accurately computing the two primary characterizing features of a stem: stem caliber and stem length.

5.1.1. Leaf bud removal

Before proceeding to compute the length and caliber of stem from its blob, we need to ensure that resulting stem blob from segmentation does not encompass any spurious part. In many cases, a leaf bud is attached to the tip of the stem and is sandwiched in between leaf petioles as shown in Figure 9(a). As a result, the stem blob usually contains an additional attachment blob at its tip representing the bud as shown in Figure 9(b). The bud attachment must be identified and removed or it may contribute to the length of the stem blob, resulting in an erroneous estimation of stem length.

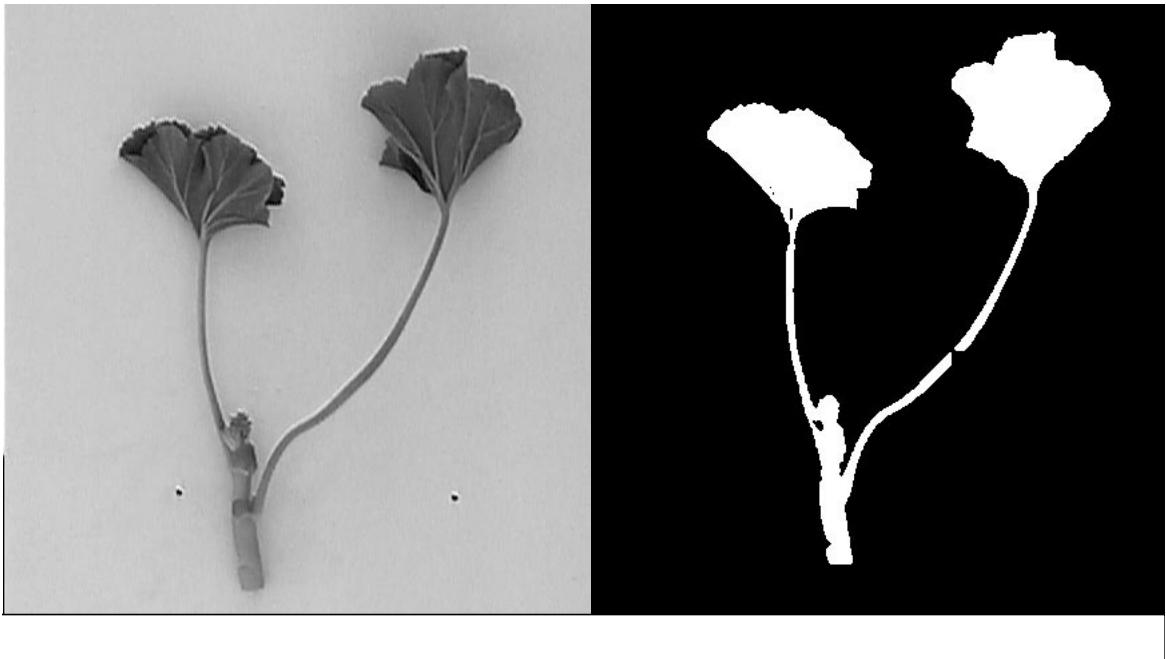


Figure 9 A leaf bud shown in gray-scale (a) and binary images (b).

A algorithm was developed to deal with this issue. The algorithm locates the stem tip at which the stem stops and the leaf bud starts, and it then removes the portion of stem from this point up. The stem tip is located based on the observation that a stem tip is usually located where the highest leaf petiole starts. As a result, the stem tip is located using data from both stem blob and leaf petiole blob.

The algorithm starts with locating the two end points for each leaf petiole and select the one closer to the stem. Among these selected leaf petiole end points, the right most point (or highest point or the one with the largest x coordinate) is selected as the potential candidate point for stem tip. This point is then evaluated to see whether it intersects with the stem. If it does, the intersection point is the stem tip. If not, the leaf petiole is extended until it intersects with stem. This is accomplished by linearly fitting leaf petiole with a line. The intersection of the fitting line with the stem is assumed to be the stem tip. While linearly fitting can identify the stem tip, it has its own limitations: (1) leaf petioles are not always straight, therefore line fitting is a problem. (2) small errors in estimating the slope of fitting line may result in substantial errors in locating stem tip, therefore large errors for estimating the length of the stem. Though this algorithm works well for most of images we have, it occasionally does break down. The breakdowns however can be minimized by instructing the human operator to follow a certain protocol when placing plants on conveyor. Meanwhile, we are investigating a more robust technique. Figure 10 shows the leaf bud in Figure 9(b) removed.



Figure 10 The stem blob for the geranium cutting shown in figure 9(b) with leaf bud removed.

5.1.2. Estimation of Stem Caliber and Length

Once the stem has been isolated from the rest of a plant, we need to estimate its width (caliber) and its length. Since the shape of a stem roughly approximates a rectangle with many local irregularities, it is difficult to estimate its length and width directly from the stem blob. To have a better estimate of the length and width of the stem, we decided to fit a rectangle to the stem blob. Our assumption is that the length and width of the best fitting rectangle should constitute an optimal estimate of the length and width of the stem.

To fit a rectangle to a stem blob, we developed an iterative gradient-descent rectangle-fitting method. Like the popular ellipse fitting algorithm [2], this algorithm is based on the premise that the spatial moments of a mass can fully characterize its orientation, location, its shape, and its contour. Though having been successfully applied to ellipse fitting, this method for rectangle fitting has not been found in literature.

Unlike the ellipse fitting, which may yield a closed-form solution, our algorithm is an iterative procedure. The algorithm starts with computing the first and second order moments for a rectangle. Moments include first and second row and column moments, and their mixed moments. It then computes the same set of moments for the stem blob. If the shape of the stem were a perfect rectangle, we could simply equate above two sets of moments to have a closed form solution for the length, width, and orientation of the rectangle. Since above assumption does not hold, we can not solve the stem length and caliber in a closed form. Instead, we try to identify unknowns (length, width, and orientation) that minimize the differences between above two sets of moments. The error function can be formulated as follows:

$$E = (Mcc_b - L^2/3)^2 + (Mrr_b - W^2/3)^2 + Mrc_b^2 + (A_b - 4WL)^2 \quad (1)$$

where Mcc_b, Mrr_b, Mrc_b, A_b are the moments for the stem blob, where W and L are the length and width of the fitting rectangle and Mcc_r, Mrr_r, Mrc_r, A_r are the moments for the fitting rectangle. Note, the moments for the stem blob are computed in the new rotated coordinate system.

The rectangle fitting algorithm can be summarized as follows:

- Fit an ellipse to the stem blob to estimate its orientation, center, and lengths of major and minor axes. To simplify computation, the center and orientation of the ellipse are assumed to be the same as the optimal fitting rectangle. So they are assumed to be known without further iterations. The lengths of major and minor axes are used as the initial estimations for the stem length and caliber.
- Compute the moments of the stem blob in a new coordinate frame. The origin of the new frame is located in the center of the image blob, with x and y axes in major and minor directions of the fitting ellipse respectively.
- Analytically compute the moments of a rectangle, with the origin of the coordinate system in its center and x and y axes in the major and minor axis direction respectively.
- Set up the error function as shown in equation (2) Using a gradient descent method, we can find following iterative equations to update the length and width of the fitting rectangle to minimize E.

$$\begin{aligned} L &= L - \frac{\alpha p E}{pL} \\ W &= W - \frac{\alpha p E}{pW} \end{aligned} \quad (2)$$

where α is the convergence rate.

- Repeat iteration until error reaches an acceptable level. Figure 11 shows a rectangle that has been fitted to the stem shown in Figure 10.

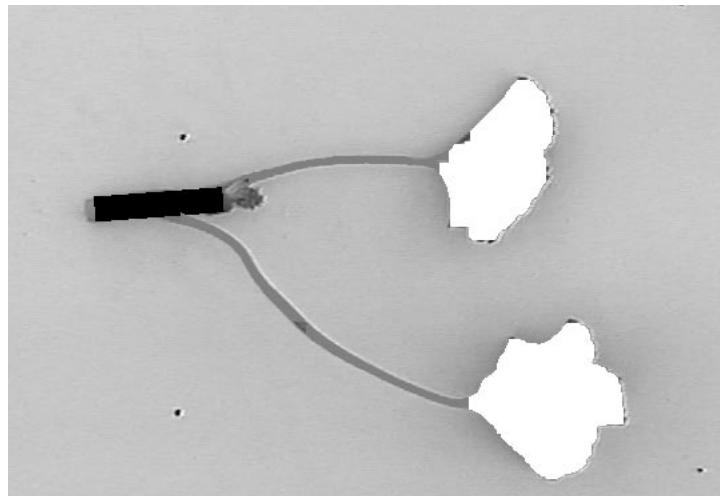


Figure 11 The rectangle that has been fitted to the stem of the cutting shown in Figure 9.

An alternative for determining the length and caliber of the stem blob is to use the extreme points method as introduced by Haralick [2]. It computes the lengths of the major and minor axis using the eight extreme points. The shortcoming with this method is that it is very sensitive to noise since the extreme points of a blob may be subject to local irregularities.

5.2. Computation of leaf petiole diameter

If we assume the leaf petioles are cylindrical, the two dimensional projection of leaf petioles on the image plane can be assumed to be something of ribbon shape with uniform width (or diameter). To estimate the diameter of each leaf petiole from its image, we estimate its area and length. The diameter of the petiole is therefore the division of its area by its length. The petiole area is simply equal to the number of pixels in the blob.

The length of petiole can not be estimated directly. Instead, a skeletonizing operation is performed on the leaf petiole blobs to find their median axes. It is assumed the length of median axis equals to the length of the petiole. Figure 12 shows the median axis for two leaf petioles. With the length and area known, the diameter of the petiole can be easily derived.

5.3. Computation of the number of leaves

To compute the number of leaves, we use leaf petiole blobs instead of the leaf blob since leaf may be folded or hidden. It is difficult to estimate the leaf number from leaf blob. On the other hand, it is relatively easy to estimate the leaf number from leaf petioles. The number of leaves is simply equal to the number of leaf petioles.

5.4. Computation of leaf area

The leaf area is simply equal to the number of pixels of leaf components.

6. Classification

The outputs from the vision algorithm for a plant cutting are the measurements of each feature as discussed in Section 5. These measurements do not directly indicate the membership of the plant, i.e., whether a plant belongs to small, medium, or large class. This is the task of supervised learning. Supervised learning involves a training process, during which the learning software is presented with a set of data points along with their output values. The learning method should be able to derive underlying relationships between the features and the classification and to generalize the relationship to predict the output of an unseen input data.

Generally, there are two types of learning methods: parametric and non-parametric method. While parametric methods require prior knowledge of the probabilistic distribution of the data, the non-parametric methods do not need this. Since we do not have prior information about the underlying distribution of our data, a non-parametric technique was implemented. We have used a method called Memory Based Learning implemented in a package called GMBL [5]. The classifiers it implements include k -nearest neighbor averaging and local regression. One of the most important features of this package is that given a dataset, it can be used to automatically choose optimal features and their weights (relative importance associated with each feature). This feature is important in that it allows reduction of the original feature set to a subset of optimal features, therefore reducing the learning time with minimum loss in classification accuracy.

The GMBL software can perform both regression and classification. Regression is used for function approximation. Given an input inquiry vector, it predicts its output value as a real number. On the other hand, classification is concerned with the membership of an inquiry vector. Its output is usually a binary number 0 or 1 ('1' indicates membership in a class). For this project, classification is more appropriate since we are ultimately concerned with the class membership of an inquiry vector rather than its output value. As a result, classification was employed for the training session (the learning period). However the classes have adjacency. It makes sense to use regression to get more accurate idea of performance. Regression is employed primarily for visualizing classification error distribution.

6.1. Data Acquisition And Data Preparation

To train the classification software, feature data were collected for 150 veronica cuttings. Three classes of cuttings were uniformly represented in the dataset, with 50 cuttings for each class. Data include measurements for five features-stem caliber, stem length, leaf number, leaf area, and mean leaf petiole diameter. Also included in the data are the label for each class. The class membership of each plant was determined manually by a human grader through visual inspection. Plants can be classified into three classes; large, medium, and small. The input vector for the GMBL software therefore consists of the five feature measurements while the output is the class of the corresponding input vector. The output can either be represented using a real number if regression is performed or a binary number of 0 or 1 if classification is performed.

The accuracy of manually determining the class membership of each class is critical since subsequent machine training is based on this classification. Since a learning algorithm can do at best as well as the human grading efforts were made to obtain objective and accurate classification for the cuttings. First, a very good human grader was employed to carry out the classification; second, after initial classification and image acquisition, the 150 veronica cuttings were grown for four weeks for subsequent reclassification. The grown cuttings were reclassified manually. Based on the reclassification, the class membership for previous 150 cuttings were modified, resulting in the second dataset. The premise here is that since the second set of data involves classifying grown vegetative cuttings, it should be more objective and accurate than the first grading even though domain experts still performs the actual classification. In the subsequent sections, we describe the training process using both sets of data. The two dataset are referred to as uncorrected dataset and ground-truth dataset respectively.

Since we do not have sufficient data to allow us to have a separate training and testing set, leave-one-out (LOO) cross-validation was used for training and testing. Specifically, the tests were conducted using the leave-one-out strategy (i.e., each instance is tested after first training all other instance in the data set). The mean classification accuracy is used as the performance evaluation criteria. The results are shown in a confusion matrix for classification and error distribution for regression as discussed below.

6.2. Experimentation

6.2.1. Using the Uncorrected Data

In this section, we will discuss training the learning algorithm using the uncorrected data. Strictly speaking, the learning software should be trained using the ground-truth data. However, to compare the classification performance of the learning method under the corrected and ground-truth data, and to study the data distribution, we conducted this experiment.

6.2.1.1. Experiment 1: Data Error Analysis via Regression

In this experiment, we performed a regression classification analysis on the uncorrected dataset using GMBL. The performance criteria for regression is the average absolute errors and its standard deviation. A total of 144 vectors are included in the dataset (the vision algorithm failed to successfully compute measurements for six of the 150 plants). Numerical labels are assigned to each class, with 1 for small, 2 for medium, and 3 for large respectively.

The GMBL software identified the features that are most relevant— 1, 3, and 4 (representing stem caliber, leaf number, and petiole diameter) weights being 1, 9, and 4 respectively. Features 2 and 4 (representing stem length and leaf area) are ignored. It seems to make sense that leaf area and stem length do not help much in classification partially because leaf area is an ill-defined feature as previously explained and stem length does not seem to have any correlation with plant grade.

Given the “optimal” combination of variables, an interface program was written to perform LOO cross-validation. The mean error for the LOO is 0.548, with the standard deviation as 0.567. Figure 12 shows the error distribution.

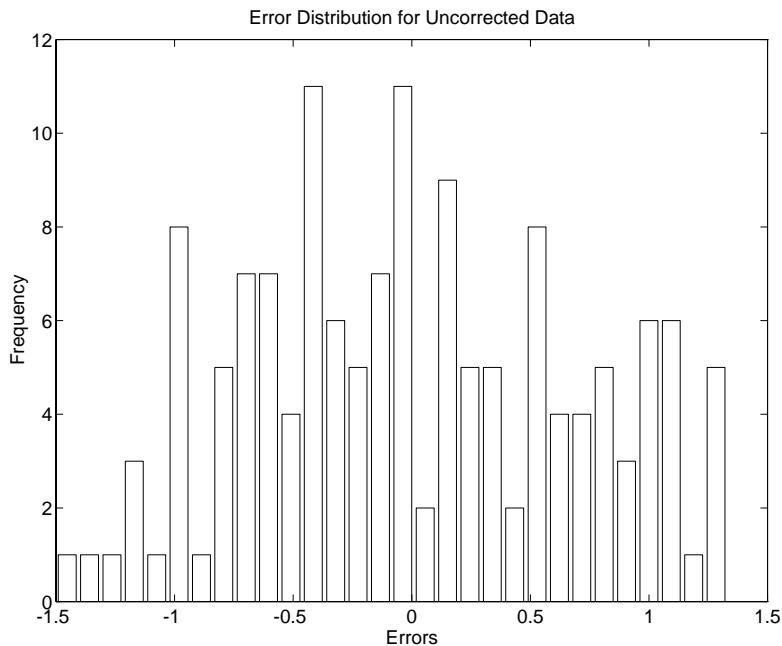


Figure 12 Regression error distribution for the uncorrected data.

Given the class labels of 1, 2, 3, for small, medium, and large classes respectively, an ideal error distribution should be a normal distribution, with mean close 0. The standard deviation depends on the percentage of errors falling within -0.5 and +0.5. For 90%, 80%, 70%, 60%, and 50%, the standard deviations are 0.303, 0.387, 0.481, 0.588, and 0.735 respectively. From figure 13, we can see a lot errors lie outside the -0.5 and 0.5. This implies many misclassifications. Furthermore, the standard deviation of 0.567 we obtained implies a classification accuracy of about 60%.

This led us to study the distribution of original dataset through visualization. Our preliminary study of the data reveals that data belonging to different classes tend to overlap with each other, especially between neighbor classes. This is further corroborated through data visualization. First, the data distribution is visualized in both 3D (using the 3 optimal features) and 2D feature space (using first and third optimal features) as shown in Figure 12 and Figure 12. No distinct cluttering exist. There are a lot overlapping. We hypothesize that overlapping is primarily due to the random errors from the inconsistencies in human grading and ambiguities associated with plants that can be classified to either one the two neighbor classes. Furthermore, random noise may not be uniformly distributed among all three classes if there are not sufficient data points.

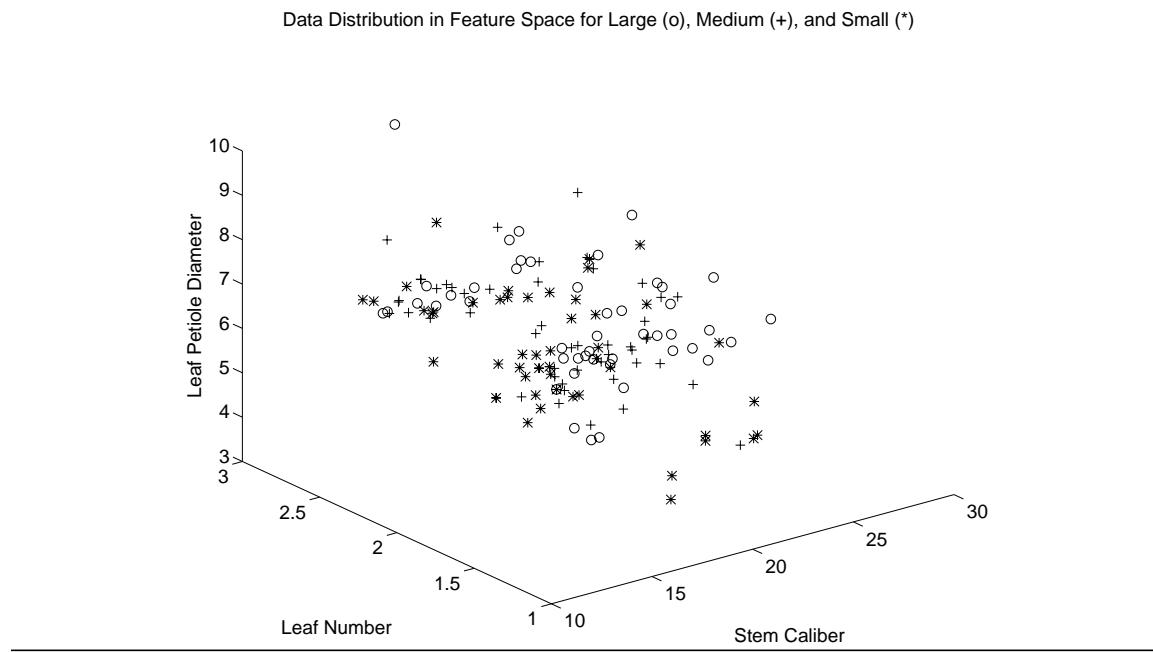


Figure 13 Data distribution for uncorrected data in 3D feature space

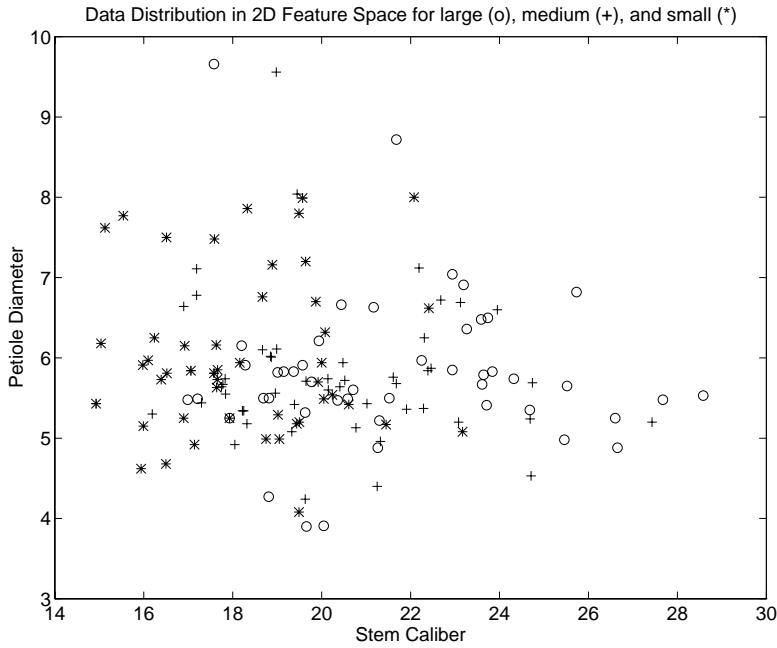


Figure 14 Data distribution for uncorrected data in 2D feature space.

6.2.1.2. Experiment 2: Regression using only small and large classes for data validation.

In this experiment, we provide support for the premise from the above analysis. We show that dataset itself is valid and primary factors for data points overlapping are due to the random errors from the inconsistency in human grading. Since we assume that most overlapping occurs between small and medium and between medium and large. Presumably, there should not much overlapping between small and large classes and distinct clustering should be formed in the feature space. Another purpose of this study is to identify the optimal feature set using the new data. The assumption here is that the empirical or heuristic rules used by human graders for grading large and small plants are the same as for grading the medium plants. Translated into statistical classification, this assumption means that the significance of a feature in discriminating large and small classes stays the same for discriminating medium class. It would be very difficult to identify the important features and their significance using the original data set since there is a lot overlapping and ambiguities associated with the membership of each plant. Therefore, removal of the data points of medium class can greatly reduce the overlapping and therefore facilitate the classification software to correctly identify the important features.

To do this, data points belonging to the medium class was removed from the original dataset, resulting in a new data set consisting of only data points for large and small plants. The new data was then fed to the classification software to identify the significance of each of the five features in discriminating the large and small classes. The optimal feature set was found to be feature 1 and 5, representing stem caliber and leaf petiole diameter. They are the most important features while other three features contribute little to the discrimination of the two classes. The result was subsequently supported by visualizing data distribution. Figure 15 shows the decision boundary and distribution of data points for large and small classes in the feature space of stem length and leaf area. It is clear from this figure that a simple linear decision boundary divides the feature space into two hyperplanes, one side mainly consisting of large classes while the other side mainly including small classes.

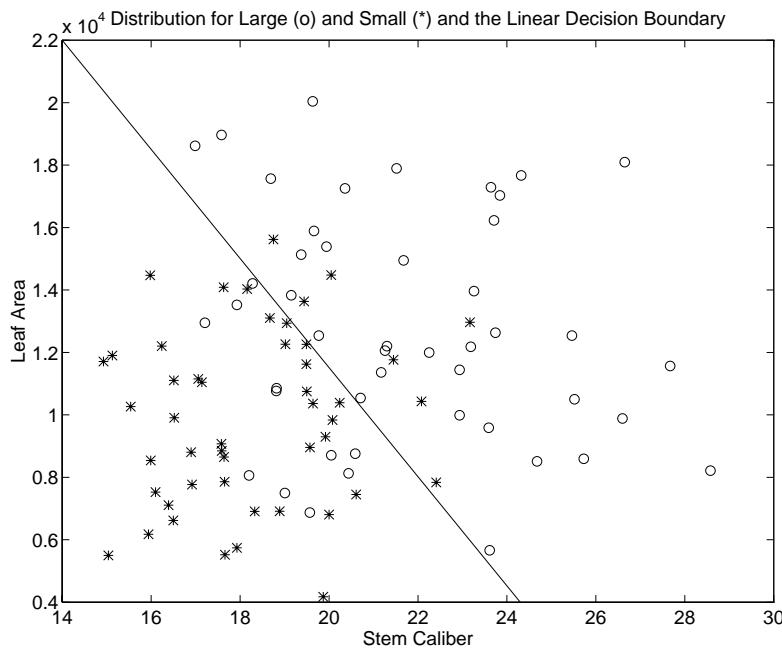


Figure 15 Distribution for large and small and the linear decision boundary

The conclusion from this study is that 1) there is a lot overlapping and ambiguities in the dataset primarily due to random errors from the inconsistency in human grading; 2) there are also biases in the original dataset due to random errors from human classification and insufficient data; 3) the optimal features are the stem caliber and the average diameters of leaf petioles. It means most important features are stem caliber and mean leaf petiole diameter. The data point distribution in the stem caliber and leaf area feature space shows much less overlapping. In fact, a linear decision boundary can be drawn to separate the two classes; 4) finally this study validates the data, which means that data from our vision algorithm is reasonably good and primary source of errors is the inconsistencies in human grading. In next section, we show that training the software with data containing bias and overlapping may lead to large classification errors.

6.2.1.3. Experiment 3: Classification Using All Data

In this section, we will perform a classification on the uncorrected dataset. While performing classification, the software expects the number of outputs equal to the number of classes. The output therefore has three nodes, representing three classes respectively. A confusion matrix is used for evaluating the performance of the classification. The confusion matrix expresses classification accuracy in terms of percentage of misclassified (or percentage of correctly classified). In addition, it also reveals classification rate within each class and to which class of a data point is misclassified and the percentage of such misclassification.

A new data set was created with new output format. The GMBL software was then run to search for the best optimal features. The best features were found to be feature number 1 (stem caliber), feature number 3 (leaf number), and number 5 (leaf petiole diameter).

Using the best string, LOO classification was performed. The classification results are shown in confusion matrix as shown in Table 1.

Table 1:

Class	Classified as Class 1	Classified as Class 2	Classified as Class 3	% Correct	% Error
1	36	4	8	75	25
2	11	21	17	42.86	57.14
3	11	13	23	48.94	51.06
Total				55.31	44.67

The average classification accuracy is about 56%, which is compatible with the 0.536 standard deviation obtained from previous regression analysis. It is also noted that of the three classes, small class has the highest classification accuracy of 75% while medium has the lowest of 38.75%, followed by large of 48.9%. We may infer that there are more ambiguities involved with medium classes than small and large classes. This seems to be intuitively correct.

Breaking down into each class, of 48 small cuttings, 12 are misclassified, out of which, four are misclassified as medium while eight are misclassified as large. This does not seem to be intuitively right. Since medium is closer to small than large, intuitively one would think more misclassification should occur in mis-classifying small as medium. This is probably either caused by the imprecision in measurements from the vision algorithm or great inconsistency in human grading. For large cuttings, 24 out of 47 was mis-classified. Out of the 24 misclassified 13 were misclassified as medium and 11 were misclassified as small. This misclassification seems to be right. Finally, for mediums, of 49 medium, 28 was mis-classified. 17 of the 28 (or 61%) was mis-classified as large and 11 (or 39%) was misclassified as small. This probably indicates the tendency of human grader in classifying medium as large. Data from all three categories seem to suggest that human graders tend to be more confused between large and medium than between medium and small.

To summarize, the best performance for the software using the uncorrected is a classification accuracy of 56%. As discussed above, the low classification accuracy primarily results from random errors from inconsistencies in human grading, the ambiguities between the number classes, the lack of quantitative measurements for each feature (the measurement of each feature is very fuzzy), and the lack of systematic protocol for human grader while visually inspecting each cutting.

6.2.2. Ground-Truth Data

In previous data set, the plant membership was manually determined by the best human grader. As a result, our vision system can at best do as well as the human grader. Also, we have realized that due to random errors from inconsistencies in human grading, substantial overlapping and bias exist in the previous dataset. In this experiment, we obtained the ground truth information in the hope to greatly reduce such ambiguities and overlapping. To do so, the previous dataset are reclassified manually using the grown vegetative cuttings. The premise here is that it is much easier to distinguish the three classes once they grow up, and less subjectivity (more objectivity) involved in the classification, therefore bringing the class membership of each cutting closer to its true class. Based on the reclassification, the original data set needs to be redistributed. Memberships of some plants are changed according to the new classification.

Considering the fact that ambiguities still exist between neighbor classes (this is observed during reclassification), we divide the cuttings into five classes; large, large-medium, medium, medium-small, and small to minimize overlapping and bias. The premise here is that most ambiguities will be concentrated in the two in-between classes (medium-large and medium-small), therefore minimizing the ambiguities associated with large, medium, and small. During the learning process, only the data from three distinct classes (large, medium, and small) are used for training. The advantage with above training method is that with reduced random noises and ambiguities associated with each class, the classification software may find a better and finer decision boundaries among three classes. The testing data however will involve data points from all five classes.

6.2.2.1. Experiment 1: Regression with all data

Like the one for the uncorrected, this experiment is primarily aimed at studying the classification error distribution for the new dataset. In the testing dataset, the in-between classes are labeled as 1.5 for small-medium class and 2.5 for medium-large class. We start with identifying the optimal features. The optimal features were found to be all five features with different weights(L07:91151). Based on the optimal feature set, a LOO cross-validation analysis was performed and error of each data point was recorded. The mean error is 0.536 and standard deviation is 0.513. Figure 15 shows the error distribution. Compared to the error distribution for the uncorrected data, we can see that the new distribution tends to be more focusing around mean, with small mean and smaller standard deviation, a sign of improvement in classification.

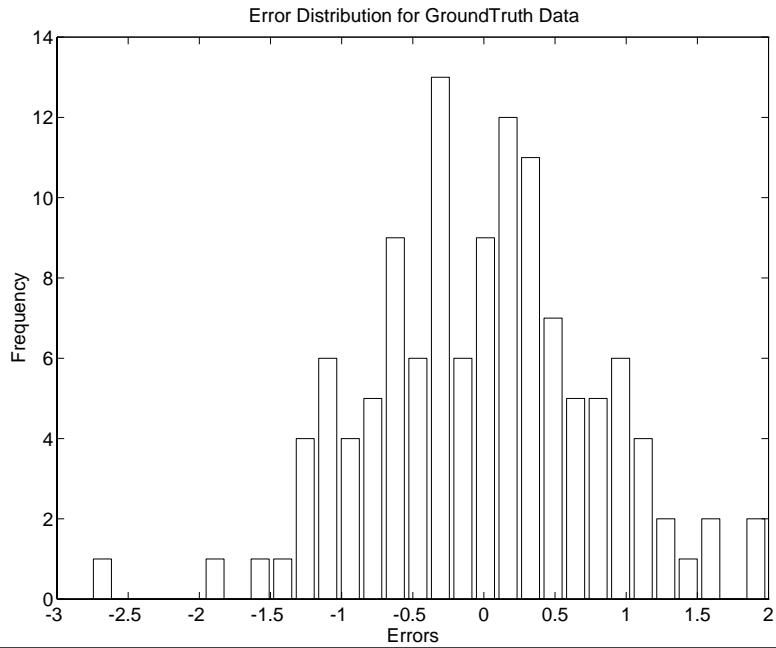


Figure 16 Regression distribution for the ground-truth data.

6.2.2.2. Experiment 2: Classification with all data

In this experiment, we perform a classification for the ground-truth data using all the data. We will also compare the machine classification accuracy against that of human. This experiment starts with identifying the best string. It is found to be L08:90002. So, the best features are found to be the stem caliber and the mean leaf petiole diameter. LOO cross-validation was then performed in two cases as shown below.

Case 1: Of the five classes, data points belonging to the in-between classes (like medium-large class) are removed. LOO was performed on the dataset without data from in-between classes. The confusion matrix for both the human and machine classification are shown in table 2:

Table 2: Human Classification

Class	Classified as Class 1	Classified as Class 2	Classified as Class 3	% Correct	% Error
1	42	4	0	91	9
2	5	29	1	82	18
3	2	14	29	65	35
Total				79	21

Table 3: Machine Classification

Class	Classified as Class 1	Classified as Class 2	Classified as Class 3	% Correct	% Error
1	33	13	4	66	34
2	12	25	7	57	43
3	5	6	18	62	38
Total				62	38

The total average classification rate for machine and human is 61.79% and 79% respectively. The average machine classification accuracy improves by 7% from previous 55%. Breaking down to each class, the small has a classification rate of 66% (down 9% from previous 75%), the medium 57% (up 15% from previous 42%) and finally large 63% (up 14% from previous 49%). The improvement for large and medium classes are big. But, there is a drop for small class. One possible explanation for the drop is that the three classes are not equally represented in the data set, with 50 small cuttings, 44 mediums, and only 27 large cuttings.

Looking at the data from machine classification, it shows that most of small misclassification occurs in mis-classifying small cuttings to mediums. Similarly, most classification error for medium also occurs in classification mediums as small cuttings. Exactly 26% of small cuttings are classified as medium and about 27% of medium is misclassified as small cuttings. 6% of small is misclassified as large. Also, 17% of large is classified small. About 16% of medium misclassified as large while 20% of large cuttings are mis-classified as medium. As a result, the mean mis-classification between small and medium is 26.5%, between medium and large is 18%, and between small and large is 11%. So, the confusion between medium and small is largest. This echoes the results from human classification. Based on above analysis, it can be seen that the performance of machine classification is roughly comparable to that of human.

Case 2: In this experiment, we will include all data points. For human grader, the data points classified into one of the in-between classes are counted as correct classification. For machine, a data point belonging to the in-between class is correctly classified if it is classified as one of the two classes involved. For example, if a point belonging to medium-large and is classified as either medium or large, then the classification is correct. However, if it is classified as small, then it is a mis-classification. For machine classification, only data belonging to the three distinct classes are used for training and all data points are used for tests. The results are shown in Table 3.

Table 4: human

small	46/50	92
medium	44/50	88
large	34/50	68
average	124/150	83

Table 5: machine

small	36/53	68
medium	40/59	68
large	20/32	63
average	96/144	69

The result from the second experiment echoes the conclusion we draw from the first one that the performance of machine classification is comparable to that of human.

7. Conclusions

Errors in accuracy are chiefly due to the following factors:

- *Inaccuracy in segmentation*: the morphology of plant cuttings even within cultivars of the same plant is so varied that it is impossible to always segment an image of a cutting correctly. Testing for a large number of special conditions is possible but will require a large amount of computation. One way to improve segmentation performance is to increase the distinction between different classes of geranium cuttings by increasing camera magnification or image resolution.
- *Small data sets*: We have experimented with small data sets so far. Larger data sets will help offset the effect of measurement errors.
- *Inconsistent ground truth data*: The ground truth data was collected by subjective visual evaluation by a human. It is advisable that ground truth be collected based on accurate measurements of the grown plants.
- *Classification methodology*: So far we have investigated various regression methods to predict grades. We have tried to capture non-linearity with methods (such as GMBL) that presume local linearity. In the future, we will attempt other methods of classification. If probability density functions can be found, then bayesian methods of classification will be tractable. If not we will try using neural networks for prediction. Early results with neural network prediction show a slight improvement (5%-7%) over the results reported above.

Despite the fact that our current results are not as high as desired (greater than 80% accuracy), the performance of our system is approximately equal to that of an average human grader.

References

- [1] Chow, K. C. and Kaneko, T., "Boundary Detection Of Radiographic Images By A Threshold Method," Frontiers Of Pattern Recognition, P61-82.
- [2] Haralick, R. B., Robot and Computer Vision, 1993.
- [3] Peleg, K., Cohen, O, Ziv M., and Kimmel, E., "Machine Identification of Buds In Images Of Plant Shoots," Machine Vision And Applications, v. 6, p224, 1993
- [4] Otsu, N., "A Threshold Selection Method From Gray-Scale Histograms," IEEE Transaction on System, Man, and Cybernetics, Vol. SMC-9, 1979, p62-66.
- [5] Moore, A. *General Memory Based Learning, A Brief Explanation of the concept, the software and the C interface* Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15217, 1994.