An Automated System for Plant-level Disease Rating in Real Fields

Muhammad Jamal Afridi[1], Xiaoming Liu[1], and J. Mitchell McGrath[2]

[1]Department of Computer Science and Engineering, Michigan State University
[2]ARS Sugar Beet and Bean Research Unit, U.S. Department of Agriculture
[1,2]{afridimu,liuxm,mitchmcg}@msu.edu

***Abstract*—Cercospora leaf spot (CLS) is the most serious disease in sugar beet plants that significantly reduces the sugar yield throughout the world. Therefore the current focus of the researchers in agricultural domain is to find sugar beet cultivars that are highly resistant to CLS. To measure their resistance, CLS is manually observed and rated in a large variety of sugar beet by different human experts over a period of a few months. Unfortunately, this procedure is laborious and subjective. Therefore, we propose a novel computer vision system, CLS Rater, to automatically and accurately rate CLS of plant images in the real field to the "USDA scale" of 0 to 10. Given a set of plant images captured by a tractor-mounted camera, CLS Rater extracts multi-scale superpixels, where in each scale a novel histogram of importances feature representation is proposed to encode both the within-superpixel local and across-superpixel global appearance variations. These features at different superpixel scales are then fused for learning a bagging M5P regressor that estimates the rating for each plant image. We test our system on the field data collected over a period of two months under different day lighting and weather conditions. Experimental results show CLS Rater to be highly consistent with a rating error of 0.65, which demonstrates higher consistency than the rating standard deviation of 1.31 by the human experts.**

I. Introduction

As a plant with a high concentration of sucrose in its roots, sugar beet accounts for 55% of the total sugar produced in the U.S. [1]. However, despite using control strategies, the yield from this crop is significantly reduced by various diseases, among which Cercospora Leaf Spot (CLS) [2] is the most serious one. This disease adversely affects about 33% of the world's sugar beet cultivation area [3]. One effective way to tackle CLS is to plant more *resistant* cultivars of sugar beet. To study the resistance property, scientists grow a large number of genetically selected cultivars of sugar beet in the real field. The resistance of each cultivar to CLS is observed and rated by experts over a course of a few months. To facilitate rating, Ruppel and Gaskill define a well-known 11-level rating system adopted by U.S. Department of Agriculture (USDA) [4], named "USDA scale". However, this manual rating system has three critical drawbacks: *subjective* where multiple experts can have different ratings for the same plant, *laborious* where it requires enormous time from experts for frequent and large-scale rating, and *relatively insensitive* where the human eye is not sufficiently sensitive to the subtle variation of the leaf appearance. Therefore, an improved rating system addressing these drawbacks is highly desired.

Given the popularity and low cost of visual sensors such as cameras, a computer vision based system can be an excellent choice for the rating system where the images of plants are analyzed and rated in an *automated*, *consistent*, and *efficient* manner. Unfortunately, the agricultural industry appears to lack such types of commercial systems. In the research community, most of the prior works focus only on detecting or classifying CLS from the *zoom-in* and *well-controlled* view of the leaf images [5]–[8]. Although such leaf-level approaches simplify the classification problem, they are practically hard to adopt due to the stringent requirement on image acquisition.

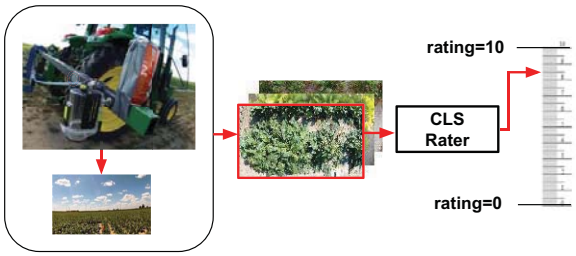
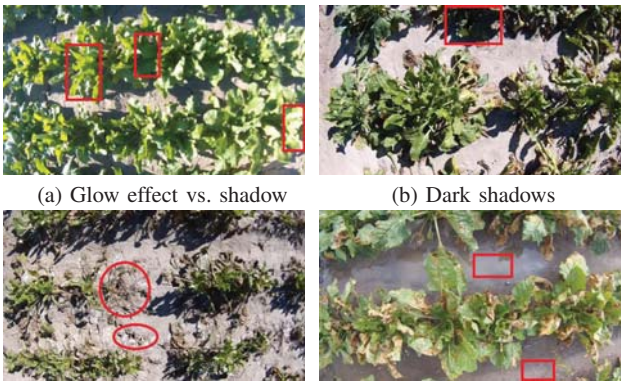


Fig. 1: A camera mounted to a field tractor records the plant videos from top view. CLS Rater performs automated analysis and assigns a rating of "USDA scale" to each video frame.

Alternatively, the *plant-level* images can be more conveniently acquired in the field via a fly-over UAV or drive-through tractor as shown in Fig. 1. However, automatic rating on plant-level images is challenging, as illustrated in Fig. 2. The varying light conditions in different weather contribute to a large amount of appearance variations in the images. Dark shadows tend to hide the details making it tough to analyze the appearance patterns of diseased spots. In the higher ratings of CLS, the dead plants mix up with the soil and hence not confusing them with soil is challenging. Similarly a bright glow in healthy leaves due to sunlight displays a yellowish color that is normally present around the diseased leaves.

To fulfill the application needs and address the technical challenges, we propose a novel system, CLS Rater, for the automated rating of CLS disease in plant-level images captured by a conventional tractor-mounted camera. Notably, this application requires our system to make a global rating estimate of a plant image by analyzing diverse appearance patterns of disease in its local regions. We tackle this challenge with our novel technical contribution of superpixel-based *Histogram of Importances (HoI)* features that describe the local patterns of each superpixel at the global image level. We then utilize



(a) Glow effect vs. shadow (b) Dark shadows

(c) Plants mix up with the soil (d) Variations in soil

Fig. 2: Appearance variations of plant images.

these features for learning image-level regression models. Although superpixels are frequently used in image segmentation problems [9]–[11], they have not been explicitly used to learn image-level regression models. Given an input image, we first extract superpixels at a pre-defined scale, e.g., M superpixels. Since each superpixel is a collection of neighboring pixels with similar appearance, a D-dimensional feature vector, e.g., a color histogram, is extracted to capture the local appearance variations of a superpixel. Given the $M \times D$ feature matrix extracted from all superpixels of an image, we describe the appearance variations across all superpixels, by computing a T-dimensional histogram for each *column* of this matrix. This results in a DT-dimensional vector, where each element describes the distribution of *relative importance* of one feature, e.g., one representative color, among all individual superpixels.

Furthermore, depending on the rating of a plant, the distinctive region of diseased leaves can have large-scale variations, from a tiny spot to an extensive area of dead leaves. Hence, the superpixels extraction is conducted at multiple scales, ranging from hundreds to thousands of superpixels, and the proposed HoI feature is extracted at each scale. Finally the features from multiple scales are fused, from which a regressor is learned based on a set of images and their manual labels of USDA scale. We test CLS Rater on a dataset collected over a two-month period in a real field under different outdoor weather conditions. Experimental results show that our system is more consistent in comparison to the human rating. CLS Rater can predict rating with an average rating error of 0.65, while the rating standard deviation of human is 1.31.

In summary, our paper makes three main contributions:

$\diamond$ We design a practical computer vision system that conveniently consumes plant-level images of a real field and automatically rate the CLS resistance in the USDA scale.

$\diamond$ We propose a novel histogram of importances feature over the multi-scale superpixels representation, and demonstrate its effectiveness in the regressor learning.

$\diamond$ We collect an image dataset of 458 sugar beet cultivars with various degrees of CLS disease and the associated manual labels in the USDA scale, over a two-month period. This dataset is publicly available to the research community[1].

[1]http://www.cse.msu.edu/~liuxm/precisionAgriculture.html

II. Prior Work

There are a number of prior works focusing on detecting or classifying CLS disease in sugar beet [5]–[7]. These approaches utilize zoom-in leaf-level images to detect the diseased segments and classify a leaf as diseased or healthy. Such approaches address a less challenging problem than ours due to the leaf-level images and a two-class classification task, while we perform regression from plant-level images. Furthermore, these approaches are hard to adopt in practices since it is inconvenient to obtain leaf-level detail of each plant in a large field. For instance, in [5], authors classify different diseases in sugar beat leaves, where the plants are grown under controlled laboratory conditions. In [6], the authors use leaf images to differentiate a CLS leaf from a healthy one by a SVM classifier. Similarly, [7] and [8] also use leaf images and utilize a threshold-based strategy to monitor the diseased part of the leaf. In contrast, we collected plant-level images in a real field under different weather conditions, which exposes our system to all kinds of *real-world* challenges. Further, our system learns a regression model that predicts the *continuous* severity of CLS disease in the 11-level USDA scale. To the best of our knowledge, this is the first study to utilize the plant-level real field images of sugar beet and automatically predict the fine-grained severity of CLS disease.

Since our feature representation builds upon the superpixel, we provide a brief overview of the related works in superpixels. With time, superpixel-based methods are becoming more advanced. For example, authors of [12] discuss how the superpixels resulting from different techniques can be combined to achieve better image segmentation. Similarly, various studies utilize superpixels for classifying local image segments. In [13], authors use a multi-scale superpixel classification approach for tumor segmentation. Furthermore, superpixels have been utilized in various other applications as shown in [9]–[11]. Note that in our study, CLS rating needs to be conducted *globally* for an entire image, while superpixels only capture *local* characteristics of an image. Hence, to fill in the gap, we need to address *how* the local characteristics of superpixels can be summarized as an image-level representation, which unfortunately has not been explicitly studied before and is the main novelty of our technical approach.

III. Our Approach of CLS Rater

The input data to our system is the plant-level imagery captured by a camera mounted on either a fly-over UAV or a horizontal pole on a regular field tractor. Specifically in this paper we use the latter, as illustrated in Fig. 1. Given a plant image, superpixels are extracted and the pixels within a superpixel are used to describe the local characteristics. Although there are many types of features for representing a local region, we decide to focus on the color and texture based features. The reason is that, when a plant is going through different stages of CLS infection, the amount as well as the color of healthy leaf, diseased leaf, and visible soil regions in plant images are changing accordingly. Thus, color can be very useful in discriminating these three types of regions and further predicting the rating. Similarly, texture also exhibits distinct patterns on these different regions.

Like any learning-based computer vision system, CLS Rater has a training stage and a testing stage. During the

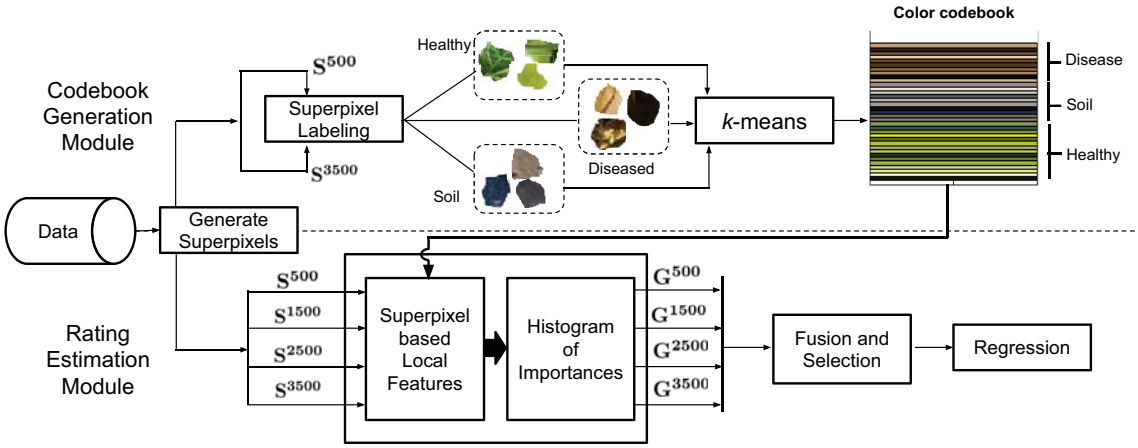


Fig. 3: The high-level architecture of our CLS Rater system.

training stage, a regressor is learned from a set of plant images with manual labels of the "USDA scale", with the goal that the predicted rating from the regressor is as close to the labeled rating as possible. While in the testing stage, the learned regressor is applied to an unseen plant image for automatically predicting its rating. As shown in Fig. 3, the training stage includes two modules: codebook generation module (CGM) and rating estimation module (REM). The goal of CGM is to model the representative colors in three different types of regions. In CGM, we manually label diverse sets of superpixels into each of the three regions, to which *k*-means clustering is applied independently for generating the codewords of these three regions. In REM, superpixels are extracted from a set of images at four scales, where at each scale a novel feature representation is used to describe both the local and global image characteristics. Features at all scales are then fused and a regressor is learned from the selected features. Processing in the testing stage is similar to REM except that it takes only one image as input. We describe the key components of the training stage as follows.

A. Superpixel Extraction

CLS appears as diseased segments in sugar beet plants and depending on the extent of disease, these segments show large-scale variations ranging from a tiny spot to a large segment. As a popular middle-level representation, a superpixel is a local segment in an image containing a group of neighboring pixels with similar appearance. Normally a scale is specified so that a pre-determined number M of superpixels can be generated for one image. To capture the local characteristics of diseased spots at all rating levels, we generate superpixels $\mathbf{S}^M = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_M\}$ of an image at four different scales where $M = \{500, 1500, 2500, 3500\}$. Using the standard implementation of [14], we observe that superpixels at each scale cover image local characteristics in a unique way, as shown in the zoom-in views of the smallest and largest scales in Fig. 4. For example, small sized superpixels, obtained with a large M, can completely fit to a small diseased spot developed in the early CLS stage. Although a larger sized superpixel cannot restrict its boundary to a small segment present in low rating images, it covers the surrounding of such a small spot and hence provides useful neighborhood contextual information,

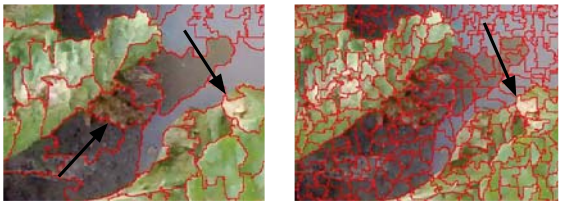


Fig. 4: Superpixels at $M = 500$ (left) and 3500 (right).

as indicated by the two parallel arrows in Fig. 4. On the other hand, in high rating images, larger superpixels can cover an entire large spot and provide a more confident indication of the severity of CLS (the leftmost arrow in Fig. 4).

B. Codebook Generation Module

Motivated by the Bag of Words (BoW) approaches [15], we first learn a color codebook to estimate the representative colors (codewords) in the plant images. From our dataset we select a diverse set of $B = 33$ images with various severities of CLS. For each image, $\mathbf{I}_i$, superpixels at multiple scales $\{\mathbf{S}_i^M\}$ are generated. Using our GUI, the superpixels $\mathbf{S}_i^M$ of image $\mathbf{I}_i$ is displayed on the screen, where a user may select superpixels belonging to healthy, diseased and soil regions via mouse clicks. The selected subsets are denoted as $\mathbf{S}_i^h$, $\mathbf{S}_i^e$, and $\mathbf{S}_i^s$ respectively. We perform this step for all B images to form $\mathbf{S}_H = \{\mathbf{S}_1^h, \mathbf{S}_2^h, \cdots, \mathbf{S}_B^h\}$, $\mathbf{S}_E = \{\mathbf{S}_1^e, \mathbf{S}_2^e, \cdots, \mathbf{S}_B^e\}$ and $\mathbf{S}_S = \{\mathbf{S}_1^s, \mathbf{S}_2^s, \cdots, \mathbf{S}_B^s\}$. We collect about 150 superpixels for each of three categories. We perform this superpixel selection procedure at two scales only. To select clean diseased spots, we use $\{\mathbf{S}_i^{3500}\}$ containing smaller superpixels, whereas it is convenient to use $\{\mathbf{S}_i^{500}\}$ for healthy plants and soil.

The RGB pixel values of all pixels within the superpixels of $\mathbf{S}_H$, $\mathbf{S}_E$, and $\mathbf{S}_S$ are fed to the *k*-means clustering for extracting codewords of each category. We extract 10 codewords for disease and soil respectively, and denote them as $\mathbf{C}_E$ and $\mathbf{C}_S$. Since the healthy part shows high variations and also responds with lighter green in regions around the diseased part, we select 15 codewords $\mathbf{C}_H$. We combine $\mathbf{C}_H$, $\mathbf{C}_E$, and $\mathbf{C}_S$ to obtain a codebook with $D = 35$ codewords $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{35}\}$, which will be used in the rating estimation module explained

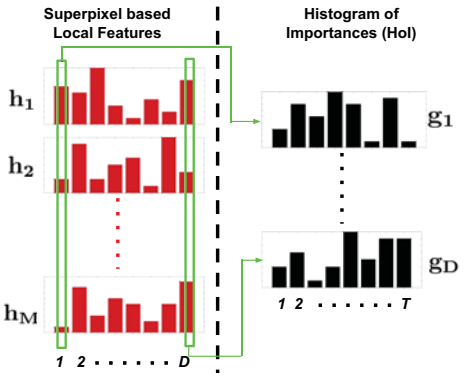


Fig. 5: Computing the Histogram of Importances (HoI).

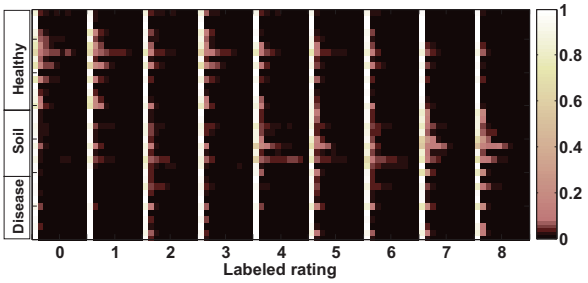


Fig. 6: Color-based HoI of 9 images with different ratings.

below. An alternative approach to our codebook learning is to directly learn the color codewords from the images, which is not preferred because the resulting codewords will mainly cover the variations in healthy and soil parts, hence creating an unbalanced codebook.

C. Rating Estimation Module

Based on the superpixels of an image set, this module performs two main tasks: 1) feature representation, and 2) feature selection and regressor learning.

1) Feature Representation: Feature representation is critical for any computer vision system. In our work, we strive to design an image-level feature that captures both the *local* pixel statistics, such as the small diseased spots, and the *global* image regularity, such as a large region of dead leaves. This leads to our proposed approach to compute our novel histogram of importances feature in two steps.

In the first step, a histogram feature is extracted to represent the color variation of all pixels within each superpixel. Given that an image $\mathbf{I}$ contains a set of M superpixels $\mathbf{S}^M = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_M\}$, we compute a set of color histograms $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_M]$. For each superpixel $\mathbf{s}_m \in \mathbf{S}^M$, we have $\mathbf{h}_m(d) = \frac{h_d}{|\mathbf{h}_m|}$, where h_d indicates the number of pixels $\mathbf{u}$ within $\mathbf{s}_m$ whose color is most similar to $\mathbf{c}_d$ among all 35 codewords, i.e., $h_d = \sum_{\mathbf{u}\in\mathbf{s}_m} \delta(d = \arg\min_d \|\mathbf{I}(\mathbf{u}) - \mathbf{c}_d\|_2)$, and $\delta()$ is the indicator function.

Although $\mathbf{h}_m$ is a good descriptor of local appearance at each superpixel, it cannot be applied to regression learning directly because superpixels between two images are not matching with each other, as well as it depends on the superpixel scale M. Hence, we aim to extract an image-level feature independent to superpixel locations or M. Specifically, by observing the matrix $\mathbf{H}$ of an image, each element $\mathbf{h}_m(d)$ indicates the relative importance of the color feature $\mathbf{c}_d$ within the superpixel $\mathbf{s}_m$. Such an importance value can vary between 0 and 1. By collecting all the importance values corresponding to the same feature $\mathbf{c}_d$, i.e., one column of $\mathbf{H}$, we can form a T-dimensional histogram of importance (HoI) $\mathbf{g}_d$, where $\mathbf{g}_d(t) = \sum_m \delta(\frac{t-1}{T} \leqslant \mathbf{h}_m(d) < \frac{t}{T})$, $1 \leqslant t \leqslant T$, and both t and T are integers. We show this procedure diagrammatically in Fig. 5. By collecting the HoI of all D color codewords, we have a $D \times T$ feature representation $\mathbf{G}^M = \{\mathbf{g}_d\}$ for one superpixel scale M.

Similar HoI features are also computed for the LBP-based texture features [16] $\mathbf{L}^M$, where $D = 256$. In our study, we use $T = 10$ for color features and $T = 5$ for LBP features. Thus, for each image at one superpixel scale, we have a total of $1,630$ features. To visualize the HoI features, Fig. 6 plots $\mathbf{G}^M$ of 9 randomly selected images at $M = 500$. We can clearly see a decrease of importances in healthy features and a slight increase of importances in soil features, as we move to higher ratings.

2) Feature Fusion, Selection and Regression: As mentioned before, superpixels at different scale cover local characteristics in different ways and provide different advantages over each other. Therefore, to get the best from every scale, we compute the color and LBP based HoI, $\mathbf{G}^M$ and $\mathbf{L}^M$, at all four scales of each image, which results in a feature vector with the length of $1,630 \times 4$. However, since not all feature elements have a high discriminative power, we perform feature selection by the correlation-based approach [17], which is based on two measures: the high predictive ability and low correlation with already selected features. We then pass the selected set of 162 discriminative features to the bagging M5P regressor [18], [19]. M5P decision tree learns different regression functions for each leaf node of the tree. Experiments in the next section provide a comparative study of different regression schemes on our features. Our results show bagging M5P to be superior to other well-known regression paradigms.

IV. Experimental Results

In this section we design experiments to answer the following questions: 1) how does CLS Rater perform in comparison to manual human rating? 2) how do different regression schemes perform at different superpixel scales? 3) is there any relation between the type of the regression methods and the type of feature representation? 4) how do our discriminative features vary across different CLS ratings? We now discuss different aspects of our experiments to answer these questions.

Dataset To record the progress of CLS disease, we collected 220 total videos of a sugar beet field from July 30, 2013 to September 12, 2013 on 10 different dates. Our field is of a rectangular shape at 135×168 meters, and plants a total of 458 sugar beet cultivars. Along the short edge of this rectangle there are 22 field lines, where our tractor drives along each of the field lines for data collection. For each field line, our system captures a video of about 3 minutes at 30 FPS, with a frame size of $1,080 \times 1,920$. We reduce the frame size to 540×960 for improved computational efficiency. A conventional RGB camera captures these videos from a height of 1.2 meters. For

TABLE I: Rating distribution of individual labels $\{r_i^1, r_i^2, r_i^3\}$.

Labeled rating	0	1	2	3	4	5	6	7	8	9	10
# of images	3	33	129	180	147	138	141	72	63	12	0

performance evaluation, we utilize a diverse set of 306 images extracted from all videos as the training and testing set. Using the USDA scale, three experts separately provide labels to all these images. The overall distribution of all labeled images across different ratings is tabulated in Tab. I.

In our experiments, we randomly split the 306-image set into two equal parts and use one for training and the other for testing. This is repeated to generate five partitions of training and testing sets. To address the inconsistency of human labeling, for each image $\mathbf{I}_i$ in our dataset, the manual ratings from three experts are averaged to generate the ground truth rating $\bar{r}_i$. Given $\bar{r}_i$ and the estimated rating of $\hat{r}_i$ from CLS Rater, we compute the *rating error* of our system on a K-image testing set as $e = (\frac{1}{K}\sum_i ||\bar{r}_i - \hat{r}_i||^2)^{\frac{1}{2}}$.

Regression Results Using our data, we evaluate a diverse set of regression methods belonging to three categories: (1) functional regression (SVM [20], Least Median Squared Linear (LMS) [21], Linear), (2) decision tree learning-based regression (M5P) [19], and (3) rule learning-based regression (M5Rules) [22]. We use bagging with each of these methods to enhance their predictive abilities. To remove the bias in coding, we utilize the standard regression implementations in [17]. Regressor training and testing are conducted on all five partitions and the average results are reported in Tab. II. We observe that while features at different superpixel scales are preferred by different regression methods, the fused feature ($\mathbf{S}^{all}$) achieves the best performance regardless of the method. Also in general M5P performs the best among all regression methods. Therefore, our CLS Rater utilizes the fused feature with a M5P regressor. One baseline method to compare with our HoI feature is the BoW features [15] based on the 35 color codewords and 256 LBP codewords of each image. As shown in the *BoW* column of Tab. II, none of the regression methods based on BoW is superior to CLS Rater.

We further explore how the regression methods perform with different types of appearance features, i.e., color and LBP. As shown in Fig. 7, in general fusing color and LBP features improves the system performance for various regression methods. However, M5P and M5Rules perform well using color alone, and fusing with LBP has no noticeable improvement.

To study how CLS Rater performs across different ground truth ratings, we plot the estimation results of all testing images in one of the partitions in Fig. 8. The narrow line-like plot shows that the rating error is evenly distributed across the entire rating range, which is desired for practical applications.

Feature Analysis We now analyze the features selected by M5P during its training process. Note that M5P is a tree-based regressor where each node is associated with a selected feature. Due to limited space, we only analyze the top hierarchy nodes (features) and select four nodes, each with a different type of feature, i.e., the color features from the disease, soil and healthy region and one LBP-based texture feature. In order

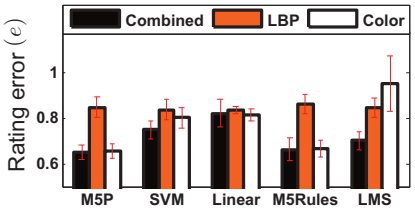


Fig. 7: Regression performance with different feature types.

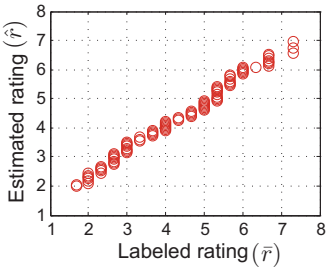


Fig. 8: Labeled rating vs. the estimated rating of CLS Rater.

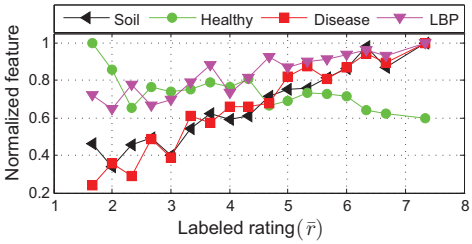


Fig. 9: Top hierarchy features of bagging M5P regressor.

to see how effective these four selected features are on the testing images, we perform the following computation. Using the same testing set as Fig. 8, we allocate the testing images with the same ground truth rating into one group. For each of the four selected features, we compute its average feature values from images within the same group. This leads to a vector for each selected feature, which is further normalized by dividing with the maximal element in the vector. We plot the resulting four vectors in Fig. 9. We see a clear trend for each of the four features. For example, there is an overall increase in the values of the disease feature and a notable decrease in healthy feature as moving to higher ratings, which is consistent with the observation of human rater in the field. This study also provides an insight on how the HoI feature element extracted from various regions contributes to CLS rating.

CLS Rater vs. Human Rating In general, it takes about 5 seasons to train an unskilled individual for rating CLS disease and at least 1 season to train a pathologist. However, it is well known that human experts tend to provide *inconsistent* rating for CLS. Hence, it is interesting to compare the rating error of CLS Rater to the error observed in human rating. The minimal estimation error for CLS Rater is 0.65, as shown in Tab. II. For comparison, we calculate the standard deviation of human rating using the *same* equation as our system error e, i.e., $e^h = (\frac{1}{3K}\sum_i\sum_j ||\bar{r}_i - \hat{r}_i^j||^2)^{\frac{1}{2}}$. Based on the same five partitions in computing e, the standard deviation of human

TABLE II: Rating error (e) at different superpixel scales.

Regression	$\mathbf{S}^{500}$	$\mathbf{S}^{1500}$	$\mathbf{S}^{2500}$	$\mathbf{S}^{3500}$	BoW	$\mathbf{S}^{all}$
M5P	0.90 ± 0.03	0.91 ± 0.04	0.88 ± 0.03	0.69 ± 0.04	0.73 ± 0.02	$\mathbf{0.65 \pm 0.03}$
SVM	1.10 ± 0.09	1.12 ± 0.05	1.05 ± 0.09	0.81 ± 0.08	0.83 ± 0.03	0.75 ± 0.04
Linear	1.46 ± 0.17	1.40 ± 0.11	1.06 ± 0.13	0.91 ± 0.03	0.83 ± 0.04	0.82 ± 0.06
M5Rules	0.92 ± 0.04	0.92 ± 0.05	0.89 ± 0.03	0.70 ± 0.03	0.74 ± 0.03	0.66 ± 0.05
LMS	1.35 ± 0.42	1.41 ± 0.17	0.95 ± 0.04	0.94 ± 0.12	0.85 ± 0.03	0.70 ± 0.04

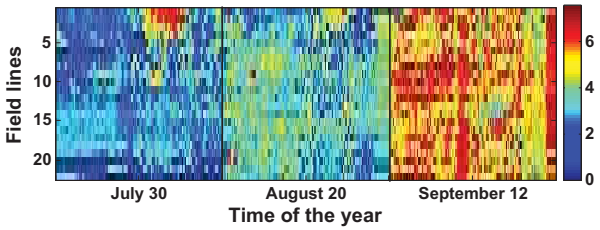


Fig. 10: CLS ratings of the entire field over two months.

rating e^h is 1.31 ± 0.08. The superior consistency of our system over the human experts indicates the great potential of applying CLS Rater in practices.

CLS Resistance Patterns in the Field Using our CLS Rater system, we can also study how the disease grows in different sections of the field over the two-month period. Since each section corresponds to a known sugar beet cultivar, this study provides many insights to the domain experts about the CLS resistances of various cultivars. As shown in Fig. 10, each box is made of 22×140 subunits where 22 is the number of field lines and 140 is the number of evenly sampled images along each field line. Note the high rating region on "July 30" is the result of almost all soil and little plant, which is likely due to the delayed plant growth at that region. We observe that cultivars at the left of the middle field lines show far less resistance to CLS, while some cultivars at the field lines 13 and 14 are very resistant even till the end of the season.

V. Conclusions

This paper introduced a novel computer vision system, CLS Rater, which uses plant-level images for the automated rating of the CLS disease in sugar beet plants. We tested our system on a real field under different lighting and weather conditions over a period of two months. CLS Rater utilizes a novel HoI feature to represent the local characteristics of superpixels at the image level and predicts the rating with an error of 0.65, which is substantially more consistent in comparison to manual rating performed by human experts. One future direction is to learn CLS Rater from a set of image pair each *ranked* by their disease severity, using approaches such as boosted rank learning [23]. Furthermore, since the technical approach of CLS Rater is very general, we will apply it to disease monitoring of other plants and a variety of precision agriculture applications in the real field.

VI. Acknowledgements

This project was sponsored in part by Michigan Sugar Company Competitive Grant (#MSC-13-08) and Project GREEEN (#GR14-034). The authors thank Tom Goodwill, Linda Hanson and Bill Niehaus for labeling images with CLS.

References

[1] http://www.ers.usda.gov/topics/crops/sugar-sweeteners/background.aspx#.UpfNwuIwBZ8.

[2] G. N. Skaracis, O. I. Pavli, and E. Biancardi, "Cercospora leaf spot disease of sugar beet," *Sugar Tech*, vol. 12, no. 3-4, pp. 220–228, 2010.

[3] B. Holtschulte, "Cercospora beticola-worldwide distribution and incidence," *Cercospora beticola Sacc. biology, agronomic influence and control measures in sugar beet*, pp. 5–16, 2000.

[4] E. Ruppel and J. Gaskill, "Techniques for evaluating sugarbeet for resistance to Cercospora beticola in the field," *J. Am. Soc. Sugar Beet Technol.*, 1971.

[5] S. Bauer, F. Korc, and W. Förstner, "Investigation into the classification of diseases of sugar beet leaves using multispectral images," *Precision agriculture*, pp. 229–238, 2009.

[6] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," *Computers and Electronics in Agriculture*, vol. 74, no. 1, pp. 91–99, 2010.

[7] H. Al Hiary, S. Bani Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, "Fast and accurate detection and classification of plant diseases," *IJCA*, vol. 17, no. 1, pp. 31–38, 2011.

[8] W. Shen, Y. Wu, Z. Chen, and H. Wei, "Grading method of leaf spot disease based on image processing," in *Int. Conf. Computer Science and Software Engineering*, vol. 6, pp. 491–494, 2008.

[9] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *ICCV*, 2009.

[10] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," in *ECCV*, pp. 352–365, 2010.

[11] H. Liu, Y. Qu, Y. Wu, and H. Wang, "Class-specified segmentation with multi-scale superpixels," in *ACCV Workshops*, pp. 158–169, 2013.

[12] Z. Li, X.-M. Wu, and S. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *CVPR*, 2012.

[13] Z. Hao, Q. Wang, H. Ren, K. Xu, Y. K. Seong, and J. Kim, "Multiscale superpixel classification for tumor segmentation in breast ultrasound images," in *ICIP*, pp. 2817–2820, 2012.

[14] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *CVPR*, pp. 2097–2104, 2011.

[15] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, vol. 2, pp. 524–531, 2005.

[16] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[17] I. H. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques," *Diane Cerra*, pp. 187–99, 2005.

[18] R. J. Quinlan, "Learning with continuous classes," in *Proc. of the 5th Australian Joint Conf. on Artificial Intelligence*, pp. 343–348, 1992.

[19] Y. Wang and I. H. Witten, "Induction of model trees for predicting continuous classes," in *ECML*, 1997.

[20] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," *IEEE Trans. Neural Networks*, vol. 11, no. 5, pp. 1188–1193, 2000.

[21] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, vol. 589. Wiley, 2005.

[22] G. Holmes, M. Hall, and E. Prank, *Generating rule sets from model trees*. Springer, 1999.

[23] H. Wu, X. Liu, and G. Doretto, "Face alignment via boosted ranking models," in *CVPR*, 2008.