On Developing and Enhancing Plant-Level Disease Rating Systems in Real Fields

Article *in* Pattern Recognition · December 2015

DOI: 10.1016/j.patcog.2015.11.021

Xiaoming Liu
Michigan State University
218 PUBLICATIONS **12,541** CITATIONS
SEE PROFILE

J. Mitchell McGrath
Michigan State University
121 PUBLICATIONS **2,502** CITATIONS
SEE PROFILE

Linda E. Hanson
United States Department of Agriculture
132 PUBLICATIONS **2,049** CITATIONS
SEE PROFILE

On Developing and Enhancing Plant Level Disease Rating Systems in Real Fields

Yousef Atoum[a], Muhammad Jamal Afridi[b], Xiaoming Liu[b,*], J. Mitchell McGrath[c]

[a]*Department of Electrical and Computer Engineering, Michigan State University*
[b]*Department of Computer Science and Engineering, Michigan State University*
[c]*ARS Sugar Beet and Bean Research Unit, U.S. Department of Agriculture*

Abstract

Cercospora leaf spot (CLS) is one of the most serious disease in sugar beet plants causing an enormous decrease in the sugar production throughout the world. Agricultural researchers are continuously seeking CLS-resistant sugar beet cultivars. Normally human experts manually observe and rate the resistance of a large variety of sugar beet plants over a period of a few months. Unfortunately, this procedure is laborious and subjective from one expert to another resulting in large disagreements on the level of resistance. Therefore, we propose a novel computer vision system, CLS Rater, to automatically and accurately rate 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 encodes 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 further address the issue of the noisy label by experts in the field, and propose a method to enhance the performance of the CLS Rater by automatically modifying the experts ratings. We test our system on the field data collected from two years over a two-month period for each year, under different lighting and weather conditions. Experimental results show that both the CLS Rater and the enhanced CLS Rater to be highly consis-

[*] Corresponding author
Email addresses: `atoumyou@msu.edu` (Yousef Atoum), `afridimu@msu.edu` (Muhammad Jamal Afridi), `liuxm@cse.msu.edu` (Xiaoming Liu), `mitchmcg@msu.edu` (J. Mitchell McGrath)

tent with the rating errors of 0.65 and 0.59 respectively, which demonstrates a higher consistency than the rating standard deviation of 1.31 by the human experts.

Keywords: CLS Rater, Histogram of Importance(HoI), Bag of Words(BoW), LBP

1. Introduction

The United States is one of the top leaders in sugar production due to the pivotal role of sizable and sophisticated sugarcane and sugar beet industries. More than one half of the total U.S. sugar production is from sugar beet [1]. However, sugar beet diseases have contributed to decline the worldwide sugar production in which Cercospora leaf spot (CLS) is the most serious one. This disease accounts for a significant reduction in sucrose production from sugar beets roots while increasing impurities concentration, which results in higher operation costs [2]. Giving the high cost and environmental effect on applying fungicides methods to overcome sugar beet diseases, planting resistance cultivators using advanced precision farming techniques is the most common and practical method to battle this disease [3]. To do this, on the weekly basis over a course of a few months, the domain experts walk through the field, visually observe the diseased plants, and rate the level of cultivators resistance using the rating system adopted by U.S. Department of Agriculture (USDA), named “USDA scale”. However, this manual rating system has three critical drawbacks: *subjective* where multiple experts may 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 leaf appearances. Therefore, an improved rating system addressing these drawbacks is highly desired.

Considering the popularity and ever-reducing cost of 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 work focuses only on detecting or classifying CLS from the *zoom-in* and *well-controlled* view of the leaf images [4, 5, 6, 7]. Although such *leaf-level* approaches simplify the classification problem, they are prac-

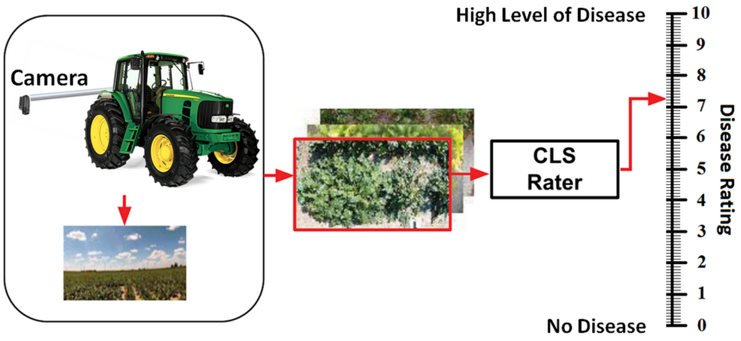
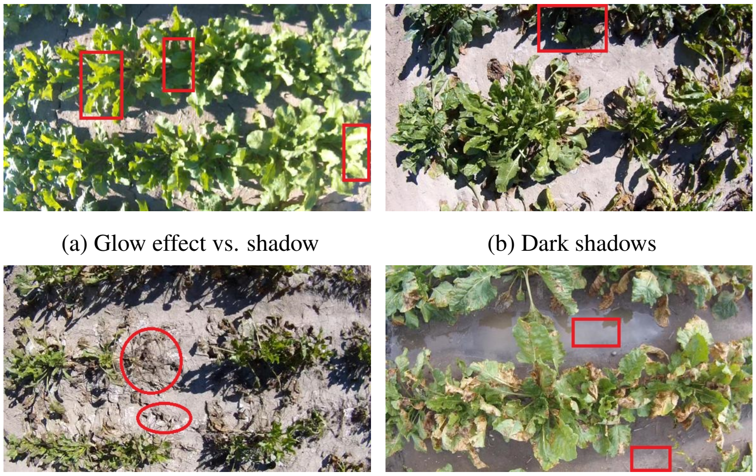


Figure 1: A camera mounted to a field tractor records the plant videos. CLS Rater performs automated analysis and assigns a rating of “USDA scale” to each video frame.

tically hard to adopt due to the stringent requirement on the image acquisition.

Alternatively, the *plant-level* images can be more conveniently acquired in the real field via a fly-over UAV or a drive-through tractor (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 difficult 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, increasing the potential of confusing them.

In order 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 tractor-mounted camera. Notably, this application demands a *global* rating estimate of a plant image by analyzing diverse appearance patterns of disease in its *local* regions. We tackle this challenge by 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 these features for learning image-level regression models. Although superpixels are frequently used in image segmentation [8, 9, 10], they have not been explicitly used to learn image-level regression models. Given an input image, we first extract superpixels at a



(a) Glow effect vs. shadow

(b) Dark shadows

(c) Dead plants mix up with the soil

(d) Variations in soil

Figure 2: Appearance variations of real-world plant images in the field.

pre-defined scale, e.g., M superpixels. Since each superpixel is a collection of neighboring pixels with similar appearance, a D-dim feature vector, e.g., a color histogram, is extracted to represent the local appearance 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-dim histogram for each *column* of this matrix. This results in a DT-dim 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 regions of diseased leaves can have diverse sizes, 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 rating (or label) in USDA scale.

Using our novel CLS Rater, we have the capability to address some of the existing drawbacks such as laborious and subjectiveness, simply by driving the tractor through the field and producing ratings for every plant in USDA scale. Unfortunately, the drawback of insensitive has not been well tackled since the manual ratings, based on which

CLS Rater is trained, are generated with insensitive eyes. Furthermore, the manual ratings are known to be noisy, as evidenced by the large variations among multiple experts. For example, in the ratings from 3 experts over a two-month period, the level of disagreement in ratings was considerably high with a standard deviation of 1.31. Hence, it is reasonable to conclude that the CLS Rater learned from the noisy ground truth still desires further improvement. Finally we hypothesize that enhancing the manual ratings of training samples is able to produce a more consistent and accurate CLS Rater. To validate this hypothesis, we propose a method to enhance human rating for better distinguishing different rating levels by the means of maximizing the separation in the feature space. This label enhancement module (LEM) takes all expert ratings as input, and strives to find the maximum separation value by iteratively adjusting each rating. The separation is measured using the multiclass Linear Discriminant Analysis (LDA), which is known to be indicative of the linear separability among multiple classes. After LEM is applied to the training set, an enhanced CLS Rater can be trained with the new rating. We experimentally show that the enhanced CLS Rater is able to further improve rating consistency on the unseen test data.

Extensive experiments are conducted by using the video data captured in the real field under different outdoor weather conditions, for consecutive two years (2013 and 2014). First, we test the CLS Rater based on the ground truth manual ratings on the 2013 dataset. 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. Furthermore, when applying the LEM, the enhanced CLS Rater can reduce the rating error to 0.59. Finally, cross-year experiments are performed by testing the CLS Rater learned in 2013 on the unseen data in 2014.

A preliminary version of this work was published in the International Conference on Pattern Recognition 2014 [11]. We have extended it in a number of ways: (i) developed the label enhancement module to address the issue of noisy labels; (ii) further reduced the rating error of CLS Rater; (iii) conducted experiments using real-world data from consecutive two years.

In summary, this paper makes four main contributions:

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

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

⋄ We address the problem of noisy labels by proposing a LEM, and experimentally show the superior performance of applying LEM over the one using the noisy labels obtained from the experts in this field.

⋄ We collect a Real-World Sugar Beet Database with various degrees of CLS disease and the associated manual ratings in the USDA scale, over a two-month period in both 2013 and 2014. This dataset is publicly available to the research community [1].

2. Prior Work

Considering the contributions of our work, we review relevant prior work in the three areas, disease rating, feature representation, and the handling of noisy labels, respectively.

There have been a number of prior work focusing on detecting or classifying CLS disease in sugar beet [4, 5, 6]. 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 acquire leaf-level detail of each plant in a large field. For instance, in [4], authors classify different diseases in sugar beat leaves, where the plants are grown under controlled laboratorial conditions. In [5], the authors use leaf images to differentiate a CLS leaf from a healthy one by a SVM classifier. Similarly, [6] and [7] also use leaf images and utilize a threshold-based strategy to monitor the diseased part of a leaf. In contrast, we collect plant-level images in a real field under diverse 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

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

scale. To the best of our knowledge, this is the first study to utilize the plant-level real field images and automatically predict the fine-grained severity of a disease.

Since our feature representation builds upon the superpixel, we provide a brief overview of the related work in superpixels. With time, superpixel-based methods are becoming more advanced. For example, authors of [12] discuss how the superpixels resulted from different techniques can be combined to improve 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 [8, 9, 10]. 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 one novelty of our technical approach.

Label noise is a well-studied problem over the last few decades, due to its negative impact on any patter recognition problems. Having noisy labels will effect the classification model, increase the complexity, and ultimately reduce the accuracy [14]. Some researchers attempt to learn models that are robust when training data has label noise [15, 16]. An alternative approach is to detect noisy labels, correct, or remove them [17, 18]. A third type of approach is to use classification filtering as a preprocessing step [19, 20, 21, 22, 23]. For example, Adaboost is used to filter mislabeled samples in [19], by eliminating a group of the samples with the highest weights. However, most prior work eliminates mislabeled data instead of correcting them, which reduces the number of samples. Also, the majority of them use synthetic data with injected noise [14], rather than real world data in our case. It is worthy to note, that the noisy CLS rating is not caused by mistakenly assigning an incorrect class label, instead it is due to the difficult nature of assigning disease ratings that may vary from one person to another, or one field to another. The limited information provided from the USDA scale of each rating class is one reason for this problem. This task is highly subjective based on how the expert interprets the different ratings from $0 - 10$. For example, in our dataset, the standard deviation of CLS ratings among multiple experts can be as high as 1.31. Therefore, given the fact that label noise is presented in almost all samples in our dataset, it is important to be able to correct or enhance the labels, which is the main goal of our LEM.

3. Proposed Approach of CLS Rater

The input data to the proposed CLS Rater is the plant-level imagery captured by a face-down camera mounted on either a fly-over UAV or a horizontal pole on a regular field tractor. Specifically in this paper we adopt the latter, as illustrated in Fig. 1. Given the captured plant images, we use a superpixel-based approach to extract features that best describes the local characteristics. The superpixel is well suited for our given problem, because it concisely and efficiently represents local appearances at a diverse range of scales, by grouping pixels with locally uniform color and texture. After superpixel extraction, there are many types of features to represent a local region. We focus on the color and texture based feature representation. Color features are the most important in this problem, since it is the core indication of CLS disease on the leaves of the plant. A CLS infected plant exhibits more yellow color in comparison to a healthy one, where the amount of yellow indicates the disease severity. When a plant is going through different stages of CLS infection, the color as well as the amount of healthy leaf, diseased leaf, and visible soil regions in plant images are changing accordingly. Therefore, color can be very useful in discriminating these three types of regions and further contributing substantially to the prediction of the rating. Similarly, texture also exhibits distinct patterns on these different regions. Healthy leaves can be described to be smoother, where diseased ones can be characterized to have dried and rough surfaces. Thus, texture is also a good candidate to discriminate between healthy and non-healthy plants.

Similar to any learning-based computer vision system, CLS Rater has a training stage and a testing stage. During the training stage, a regressor is learned from a set of plant images and their ratings in “USDA scale”, with the goal that the predicted rating from the regressor is as close to the manually labeled rating as possible. While in the testing stage, the learned regressor is applied to an unseen plant image for automatically

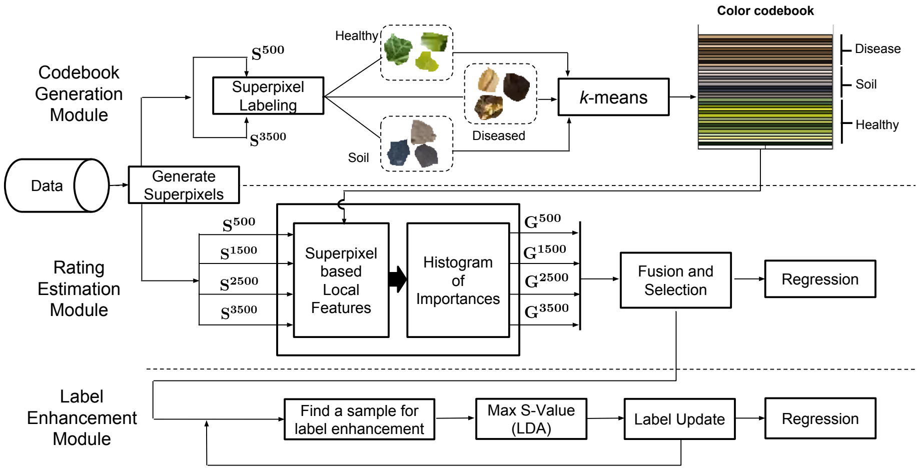


Figure 3: The high-level architecture of our CLS Rater system.

predicting its disease rating. As shown in Fig. 3, the training stage includes three modules: codebook generation module (CGM) and rating estimation module (REM), label enhancement module (LEM), while the testing stage only includes the REM.

The goal of CGM is to model the representative colors in three different types of regions, i.e., healthy, soil and disease. 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 the same as REM except that it takes only one image as input. In LEM, we perform label enhancement on the manual ratings obtained from the experts in the field in order to reduce the amount of label noise and better distinguish all possible ratings. This is accomplished by iteratively adjusting the existing rating of each sample, with the goal of achieving the maximum separation among the training samples of different ratings in the feature space. The separation is measured using the multiclass Linear Discriminant Analysis (LDA), which explicitly models the linear separability among the data of multiple classes. We describe the key components

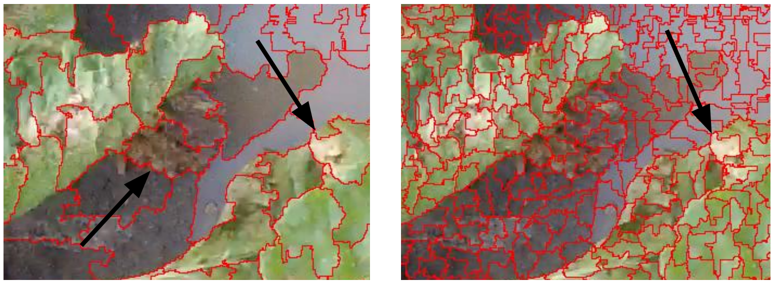


Figure 4: Superpixels at $M = 500$ (left) and 3500 (right).

of the training stage starting from superpixel extraction, to a detailed explanation for all three modules as follows.

3.1. Superpixel Extraction

CLS disease in its early stages appears as very small spots located on the leaves of the sugar beet plant. As the disease progresses to higher levels, the spot of the disease grows in size and changes in color accordingly. Therefore, the disease segments show large variations of scales ranging from a tiny spot to a large segment depending on the level of CLS disease. 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 [24], we observe that superpixels at each scale cover local image 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, 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). Combining all the features obtained from superpixel of various M scales will effectively describe all rating levels of the disease.

3.2. Codebook Generation Module

For an arbitrary image, the color of pixels may not have a priori distribution. However for domain-specific images such as sugar beet plant images, it is safe to assume that a distribution of pixel color exists and can be learned for efficient feature representation. Therefore, motivated by the Bag of Words (BoW) approaches [25], we first learn a color codebook to estimate the representative colors (codewords) in the plant images as illustrated in Fig. 3, so that they can be used later for feature representation. From our dataset we manually 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 extracted. To facilitate the labeling for CCM, we develop a GUI where the superpixels $\mathbf{S}_i^M$ of image $\mathbf{I}_i$ is displayed on the screen and a user may select superpixels belonging to healthy, diseased or 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. This superpixel selection procedure is performed at two scales only: $\{\mathbf{S}_i^{3500}\}$ containing smaller superpixels for selecting diseased spots, and $\{\mathbf{S}_i^{500}\}$ 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 each for the disease and soil categories, and denote them as $\mathbf{C}_E$ and $\mathbf{C}_S$ respectively. Since the healthy part shows larger 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 form 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 described 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 a biased codebook.

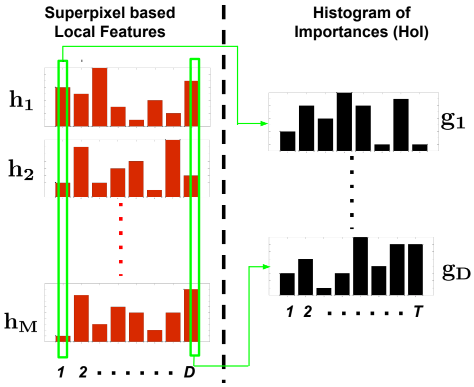


Figure 5: From the histograms of individual superpixels to the Histogram of Importances (HoI).

3.3. Rating Estimation Module

Given the color codewords from the CGM, as well as the superpixels of an image set, this rating estimation module performs two main tasks: 1) feature representation, and 2) feature selection and regressor learning. We now discuss them as follows.

3.3.1. Feature Representation

Feature representation is critical for any computer vision system. Classifying local regions in superpixel segments into diseased or healthy may seem to be a trivial task. However, it is unclear how to generalize this task to consider a global 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. Moreover, a global fine-grained continuous rating needs to be learned from the feature representation of images. These considerations lead to the proposed novel histogram of importances feature, computed in two steps.

In the first step, a histogram feature is extracted to represent the color variation of all pixels within each superpixel based on the color codewords. 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 $\mathbf{H} = [\mathbf{h}_1^\intercal; \mathbf{h}_2^\intercal; ...; \mathbf{h}_M^\intercal]$. 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

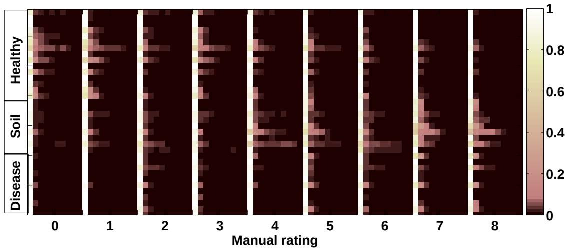


Figure 6: Color-based HoI of nine images with different ratings.

to $\mathbf{c}_d$ among all D 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 may not correspond to each other, and the numbers of superpixels M can be different too. 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-dim 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 [26] $\mathbf{L}^M$, 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 importance in healthy features and a slight increase of importance in soil features, as we move to higher ratings.

3.3.2. 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 enjoy the benefits from every scale, we compute the color and LBP based HoI, $\mathbf{G}^M$ and $\mathbf{L}^M$, at all four scales for 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 [27], which is based on two measures: the high predictive ability and the low correlation with already selected features. We then pass the selected set of 162 discriminative features, $\{\check{\mathbf{G}}^M, \check{\mathbf{L}}^M\}$, to the bagging M5P regressor [28, 29]. M5P decision tree learns different regression functions for each leaf node of the tree. Experiments in Section 5 provide a comparative study of different regression schemes on our features. Our results show that bagging M5P to be superior to other well-known regression paradigms.

3.4. Label Enhancement Module

So far we have presented a carefully designed learning-based approach to automatically estimate or mimic the disease rating manually labeled by domain experts. However, such manual ratings, either from one expert or the average of multiple experts, are inevitably noisy. For example, Fig. 7 shows that the disagreement among experts is almost everywhere on an entire dataset, with especially large variation for some images (Fig. 7 (a)). As mentioned before, the noisy label is caused by a number of factors, including the insensitive eyes of the human, the vague definition of USDA scale, the existence of multiple plants within one image, etc. For these reasons, this issue cannot be solved by the experts, and thus an automatic method to enhance the noisy labels of a dataset is desired, which is exactly the objective of LEM.

One potential approach of LEM is to adopt unsupervised learning to learn 11 clusters, each corresponding to one level of CLS disease. However, our preliminary experiment shows that without supervision it is difficulty to ensure that the clusters are indeed defined based on the CLS disease. Therefore, we make the following assumption: the noisy-free rating of a data sample is in close approximation to its manual rating, and it is thus possible to obtain the former by making a small adjustment to the latter. Based

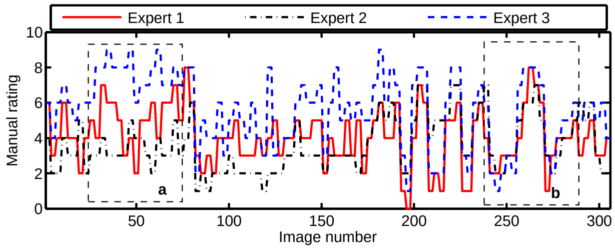


Figure 7: Assigning disease rates to images from the real-world field is challenging. There are large variations (a) and small variations (b) in manual ratings of three experts.

on this assumption, we take the manual ratings as the starting point, and improve them in a systematic manner, with the goal that the enhanced labels will make the different rating levels more discriminative in the feature space. This will in turn result in an enhanced CLS Rater, when trained from the enhanced labels. Specifically, given a dataset and its manual ratings as input, after feature extraction from the REM, the LEM iteratively updates one rating at a time in order to maximize the separation among samples of different ratings. A simple illustration of our proposed LEM is shown in Fig. 3, and a more detailed explanation is in Algorithm 1.

It is obvious that the order of samples being processed within an input dataset of N samples affects the final enhanced labels. Thus, we denote the method for selecting which sample to update label by $F : \mathbf{Y} \mapsto j, j \in 1, 2, .., N$, where j is the index of the candidate sample. In this work we explore three options for implementing this function: (i) Random Label Selection: This function randomly select one sample from the input dataset, without considering any prior knowledge about the label. (ii) Maximum Disagreement First: This function ranks all samples in the descending order of the disagreement among the experts. It basically first selects samples with the most confusing labels (i.e., the largest disagreement). (iii) Maximum Offset First: Given a dataset and the current labels, a M5P regression-based CLS Rater is learned and applied to the training dataset. The sample with the maximum difference between the current label and the rating predicted by CLS Rater is selected as the sample to be processed.

After finding the candidate sample, we assume its label $\tilde{\mathbf{Y}}_j$ can make a small ad-

Algorithm 1: Label enhancement module.

```
Data: $\mathbf{Y}, \mathbf{X} = \{\check{\mathbf{G}}^M, \check{\mathbf{L}}^M\}$
Result: $\tilde{\mathbf{Y}}$
$\tilde{\mathbf{Y}} = \mathbf{Y}, S = 0$;
do
    $j = F(\tilde{\mathbf{Y}})$;
    $\tilde{\mathbf{Y}}_j^{set} = [\lceil \tilde{\mathbf{Y}}_j + 0.5 \rceil, \lfloor \tilde{\mathbf{Y}}_j \rfloor, \lfloor \tilde{\mathbf{Y}}_j - 0.5 \rfloor]$;
    for $i = 1 : 3$ do
        Partition $N$ samples into $c$ classes based on $\tilde{\mathbf{Y}}$ and $i^{th}$ element of $\tilde{\mathbf{Y}}_j^{set}$;
        $\mathbf{\Sigma} = \sum_{m=1}^{c} \sum_{n=1}^{N_m} (\mathbf{x}_n^m - \mu_m)(\mathbf{x}_n^m - \mu_m)^\mathsf{T}$;
        $\mathbf{\Sigma}_b = \sum_{m=1}^{c} (\mu_m - \mu)(\mu_m - \mu)^\mathsf{T}$;
        $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots] = eig(\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_b)$ ;
        $S^{set}(i) = \sum_{m=1}^{c-1} \frac{\mathbf{w}_m^\mathsf{T} \mathbf{\Sigma}_b \mathbf{w}_m}{\mathbf{w}_m^\mathsf{T} \mathbf{\Sigma} \mathbf{w}_m}$;
    end
    $i = \underset{i}{\operatorname{argmax}}\, S^{set}$;
    $\tilde{\mathbf{Y}}_j = \tilde{\mathbf{Y}}_j^{set}(i)$;
    $S_{pre} = S, S = S^{set}(i)$;
while $S > S_{pre}$;
```

justment to one of the following neighboring ratings $[\lceil \tilde{\mathbf{Y}}_j + 0.5 \rceil, \lfloor \tilde{\mathbf{Y}}_j \rfloor, \lfloor \tilde{\mathbf{Y}}_j - 0.5 \rfloor]$. Therefore, we consider the possibility of either modifying this label to one of the neighboring ratings, or maintaining its current label. For each possibility, we compute the S value of the feature set $\mathbf{X}$ given the updated labels $\tilde{\mathbf{Y}}_j$, where S is the class separability computed via a multiclass Linear Discriminant Analysis (LDA). Specifically, we compute the $\mathbf{\Sigma}_b$, $\mathbf{\Sigma}$, $\mathbf{W}$, and S is the summation of eigenvalues that are indicative of linear separability among multiple ratings. Finally, we update $\tilde{\mathbf{Y}}$ to the possibility that produces the maximum S value. Note that the label of a sample can be modified more than once, when other samples in the dataset are modified. While making modification on the labels of the samples, it is important to preserve the range of the labels because the S value will shrink if the range is reduced. Therefore, we have a constraint to enforce that no label modification is performed for samples with the maximum rating or minimum rating of a particular dataset. The enhancement process will continue until there is no increase in S values. This means that all data samples are well separated into rating clusters with the minimum overlap among the clusters, and a regressor will then be learned based on the enhanced labels $\tilde{\mathbf{Y}}$.

4. Real-World Sugar Beet Database

Although there are prior works on applying computer vision for agriculture applications, there is very few public databases of plant images that are captured in the real field. Hence, one task of our work is to acquire a sugar beet plant database in two consecutive years with the same imaging setup, and making this database publicly available is also one contribution of our work.

A conventional RGB camera was attached to a tractor pointing downwards at a height of 1.2 meters. The tractor drives through the sugar beet field while maintaining a constant speed of $\sim$20 miles per hour, capturing videos at a frame size of $1,080 \times 1,920$ and 30 frames per second for the entire field. We reduce the frame size of all images to 540×960 for improved computational efficiency. To record the progress of CLS disease, we collect videos periodically during the sugar beet season, across a period of two months capturing a wide range of disease severity. Our sugar beet field is of a rectangular shape at 135×168 meters. Each section of the field corresponds to a known sugar beet cultivar, with a total of 458 cultivars over the entire field. Hence, the CLS rating study provides many insights to the domain experts regarding the CLS resistances of various cultivars. Along the short edge of this rectangle there are 22 parallel field lines with equal distances between them, where our tractor drives along each of the field lines for data collection.

The first part of the database was captured from July 30, 2013 to September 12, 2013 on 10 different dates. We collect 220 total videos, i.e., 22 videos per day. Each video is about 3 minutes long and covers one field line. We select a diverse set of 306 images from this dataset ranging through all dates to capture all possible disease ratings. Using the USDA scale, three experts separately provide manual ratings to all

Table 1: Overall distribution of all labeled images across different ratings.

Manual rating	0	1	2	3	4	5	6	7	8	9	10
# of images in 2013	1	11	43	60	49	46	47	24	21	4	0
# of images in 2014	0	34	575	630	878	1121	663	121	2	0	0

these images. The overall distribution of all labeled images across different ratings is tabulated in Tab. 1. The ratings provided from three experts for all 306 images are also shown in Fig. 7, illustrating the variations in the ratings.

The second part of the database was captured from August 15, 2014 to September 12, 2014 on 7 different dates. This part used the exact same imaging setup as the first part, where the only differences are in the capturing and labeling procedure. Instead of capturing every line in the field separately, the entire field was captured in a total of 2 videos. A GPS system, as an integrated component of the tractor, was utilized to record exact longitude and latitude coordinates while capturing videos. For this part, only one expert provides manual rating to the plants on 4 out of 7 dates, while she walks through the field, and the manual ratings are recorded w.r.t. the locations of cultivators. Since we aim to have labels for all 2014 dataset, we did not ask the expert to manually label a small subset of images. Instead, using the GPS data, we map all manual ratings in the field to specific video frames, as shown in Fig. 8. However, due to imprecise GPS data, the manual ratings in 2014 dataset is not as ideal as the one in 2013 dataset.

5. Experimental Results

In this section, based on the Real-World Sugar Beet Database, we design experiments to answer the following questions: 1) how does the CLS Rater perform in comparison to manual expert rating? 2) how do different regression schemes perform at different superpixel scales? 3) how do our discriminative features vary across different CLS ratings? 4) Does maximizing the separation value in the LEM indeed change labels according to disease levels? 5) How do we evaluate the performance of the enhanced labels? We now discuss different aspects of our experiments to answer these questions.

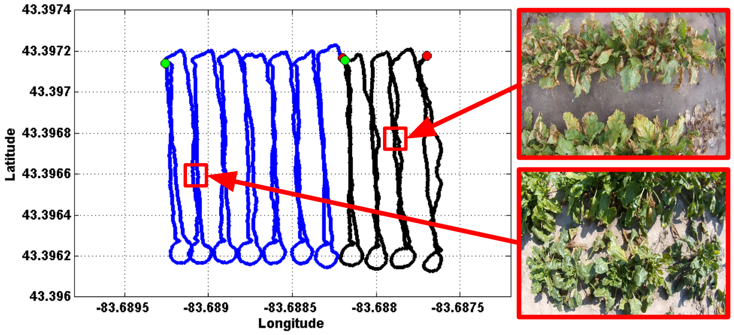


Figure 8: Mapping GPS coordinates to specific video frames. The blue and black lines represent two video sequences captured on August 21, 2014. The green and red circles represent the start and end of each video sequence, respectively.

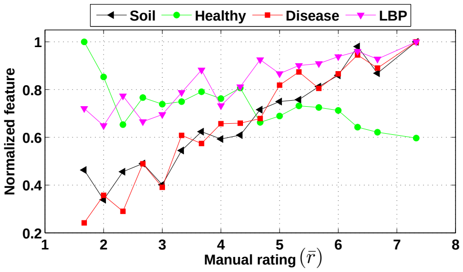


Figure 9: Top hierarchy features of bagging M5P regressor.

Experimental Setup. Most of our experiments are based on the 2013 dataset, where we randomly split the 306-image set into two equal parts and use one for regressor training and the other for testing. This is also repeated to generate multiple partitions of training and testing sets. 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 $e = (\frac{1}{K}\sum_i ||\bar{r}_i - \hat{r}_i||^2)^{\frac{1}{2}}$.

Feature Analysis. We start by analyzing the performance of the proposed HoI features and the selected features by one of the best performing classifiers, M5P regressor, during the training stage. Specifically, we evaluate the effectiveness of the selected features and compute their feature value across different unseen testing images with varying CLS disease ratings. Note that the M5P is a tree-based regressor, where each node is associated with a selected feature. From the M5P hierarchy, we select top four nodes (features) that represent different type of features, i.e., the color features from the disease, soil and healthy region and one LBP-based texture feature. In order to see how effective these four selected features are on the testing images, 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, which illustrates a clear trend of the four features. We notice a proportional relationship among soil, disease and LBP features with a high correlation in the behavior across ratings. Whereas, the healthy leaves tends to have an inverse relationship with all other features. This is highly expected, since at higher ratings, the amount of green leaves in the frame decreases, which are typically replaced with disease leaves and soil. This study also provides an insight on how the HoI feature element extracted from various regions contributes to CLS rating.

CLS Rater Prediction Analysis. While Fig. 9 indicates the strong correlation between the novel HoI features and the rating, the ability of CLS Rater to predict rating is more important. Our CLS Rater is design to predict ratings based on the USDA scale with 11 different levels of disease ratings. To analyze the predictions of our rater based on the novel HoI features, we attempt to test the discriminative ability of the rater across a large variety of ratings. Using the experimental setup on the 2013 dataset, the predictions on one testing set are illustrated in Fig. 10. The narrow line-like plot shows that the rating error is evenly distributed across the entire rating range, and also our CLS Rater is able to predict labels very similar to the human labels on the unseen data, which is desired for practical applications.

Label Enhancement Results. We now study the LEM and its contribution to CLS Rater. First, we explore the various methods for selecting which sample to update the label, which is the function F with three options: random label selection, maximum dis-

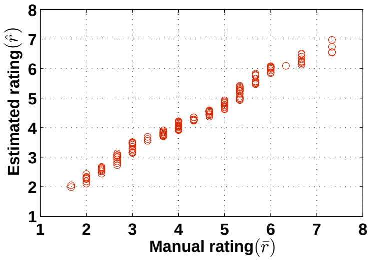


Figure 10: Ground truth manual rating vs. the estimated rating of CLS Rater.

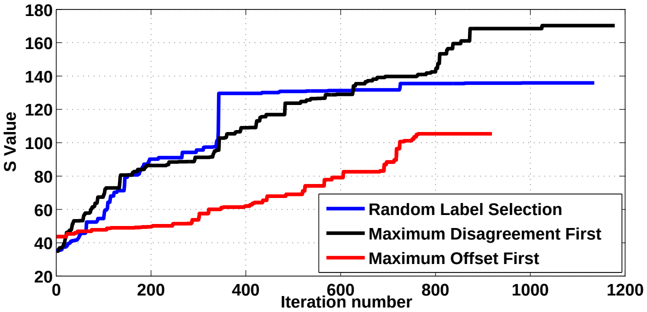


Figure 11: S value comparison of the three sample selection functions.

agreement first or maximum offset first. We attempt to enhance the ground truth label of the training set of 2013 dataset with a total of 153 images. Fig. 11 shows a comparison of all three functions during the iterative process of selecting the candidate sample. It is worthy to note, that all three methods start at low S values meaning that the ground truth ratings are not well separated among different ratings. The best resulting S value is produced using the maximum disagreement first method, which converges at $S = 172$ after a total number of $1,982$ iterations. This method selects the sample with rating that has the highest inconsistency among multiple experts.

After the label enhancement converges, we can compare the original ground truth ratings (average of three manual ratings) with the enhanced ratings generated from the maximum disagreement first method, as shown in Fig. 12. On one hand, although on average each sample has been modified its rating for 13 ($\approx \frac{1982}{153}$) times, the differences

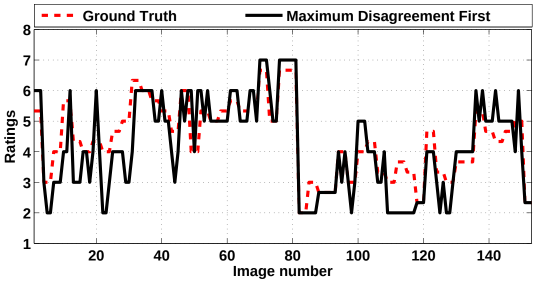


Figure 12: Comparing the enhanced ratings generated from the maximum disagreement first method with the ground truth ratings.

between the original and final ratings are very minimal, where the absolute difference has a distribution of $\mathcal{N}(0.56, 0.62)$. This is a good indication of our assumption that noisy-free label of a sample is in close approximation to its manual label. On the other hand, even with a small modification on the ratings, a much larger S value is achieved which indicates improved separability among different ratings.

Since the LEM operates on a particular dataset, it is possible that one sample might converge to *different* enhanced rating when it is a member of different dataset. Obviously this is not desired, and therefore we design experiments to explore this potential issue. On the 2013 dataset, we generate five random subsets of data with different number of images, and apply LEM based on maximum disagreement first to each subset. Fig. 13 shows the label enhancement results for all five subsets, and the bottom row shows measure the standard deviation of the enhanced ratings of common samples across five sets. An average standard deviation of 0.41 is obtained over all common samples. Therefore, we can observe that the dependency of enhanced ratings to a particular dataset composition is relatively low, and it seems that the enhanced ratings are moving toward the noisy-free labels of the samples.

Fig. 11 shows that the larger separability can be achieved using the enhanced ratings on the dataset where LEM is applied. The next step is to validate that if we learn an enhanced CLS Rater from the enhanced ratings and apply it to an unseen dataset, whether a larger separability can still be observed. To test this generalization capability, using the training set we learn four CLS Raters based on four labels, the ground

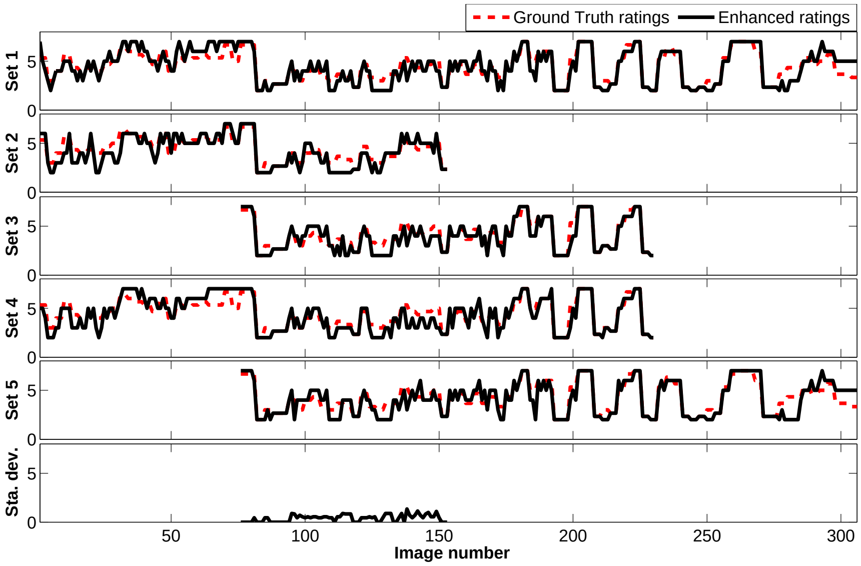


Figure 13: Convergence analysis of LEM. Rows $1 - 4$ represent the results of applying LEM to different subsets of 2013 dataset. Row 5 is the standard deviation of the enhanced ratings of common samples in five subsets.

truth ratings and the enhanced labels with each of three sample selection functions, respectively. Each CLS Rater is applied to the testing set, and based on the estimated ratings all testing samples can be grouped into multiple classes. Then we compute the eigenvalues of the matrix $\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_b$, where $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_b$ are computed as in Algorithm 1. By repeating this experiment on ten random partitions of training and testing sets, we show the distribution of top eigenvalues in Fig. 14. Since larger eigenvalues indicate high linear separability among the classes, the result demonstrates that the enhanced CLS Rater is able to make the unseen testing set more separable and less confusing between consecutive rating levels. Also, among the three sample selection functions, the maximum disagreement first method seems to have a minor advantage over the others.

Regression Results. Using the 2013 dataset, we evaluate a diverse set of regression methods belonging to three categories: (1) functional regression (SVM [30], Least Median Squared Linear (LMS) [31], and Linear), (2) decision tree learning-based re-

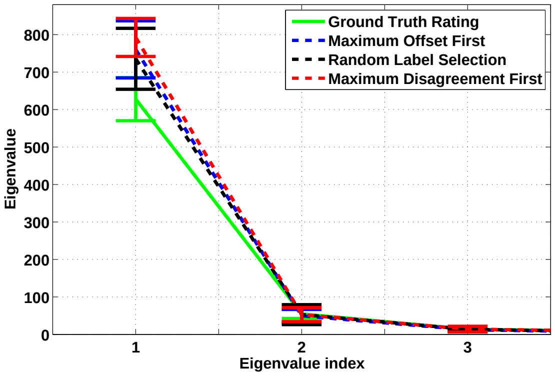


Figure 14: Eigenvalues of the LDA on the testing set when the CLS Rater is trained with different labels.

Table 2: Rating error (e) at different superpixel scales.

Regression	LEM	$\mathbf{S}^{500}$	$\mathbf{S}^{1500}$	$\mathbf{S}^{2500}$	$\mathbf{S}^{3500}$	BoW	$\mathbf{S}^{all}$
M5P	No	0.90 ± 0.03	0.91 ± 0.04	0.88 ± 0.03	0.69 ± 0.04	0.73 ± 0.02	$\mathbf{0.65 \pm 0.03}$
	Yes	0.72 ± 0.03	0.73 ± 0.06	0.75 ± 0.02	0.62 ± 0.02	0.72 ± 0.02	$\mathbf{0.59 \pm 0.04}$
SVM	No	1.10 ± 0.09	1.12 ± 0.05	1.05 ± 0.09	0.81 ± 0.08	0.83 ± 0.03	0.75 ± 0.04
	Yes	0.69 ± 0.03	0.79 ± 0.05	0.79 ± 0.07	0.63 ± 0.02	0.79 ± 0.08	0.60 ± 0.02
Linear	No	1.46 ± 0.17	1.40 ± 0.11	1.06 ± 0.13	0.91 ± 0.03	0.83 ± 0.04	0.82 ± 0.06
	Yes	0.67 ± 0.02	0.78 ± 0.02	0.75 ± 0.02	0.64 ± 0.01	0.79 ± 0.03	0.62 ± 0.01
M5Rules	No	0.92 ± 0.04	0.92 ± 0.05	0.89 ± 0.03	0.70 ± 0.03	0.74 ± 0.03	0.66 ± 0.05
	Yes	0.66 ± 0.03	0.73 ± 0.03	0.76 ± 0.01	0.65 ± 0.01	0.78 ± 0.04	0.61 ± 0.02
LMS	No	1.35 ± 0.42	1.41 ± 0.17	0.95 ± 0.04	0.94 ± 0.12	0.85 ± 0.03	0.70 ± 0.04
	Yes	0.66 ± 0.01	0.81 ± 0.08	0.76 ± 0.02	0.64 ± 0.01	0.88 ± 0.03	0.62 ± 0.05

gression (M5P) [29], and (3) rule learning-based regression (M5Rules) [32]. 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 [27]. Tab. 2 shows the results where the mean and standard deviation of rating errors are computed from five random partitions of the 2013 dataset. When no "LEM" is used, both the training and testing are based on the ground truth ratings, i.e., the average of three ratings.

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. The baseline method to compare with our HoI feature is the well-known BoW features [25] based on the 35 color codewords and 256 LBP codewords of each image. As shown in the *BoW* column of Tab. 2, none of the regression methods based on BoW is superior to CLS Rater.

By using the LEM, we evaluate the performance of enhanced CLS Rater. Since the enhanced CLS Rater is trained on the enhanced labels, its rating error is also computed w.r.t. the enhanced labels on the testing set, which is obtained by apply the LEM to the testing set. It can be seen that the fused feature with a M5P regressor is still superior to other regressors or other features. Also, in almost all cases, the enhanced CLS Rater has smaller rating errors than the original CLS Rater, with the minimum error reduced from 0.65 to 0.59. On one hand, this superiority indicates that the enhanced CLS Rater can predict rating more consistently and with less confusion. On the other hand, the reduced error is especially encouraging when the labels in the training set and testing set are *idenpendently* enhanced via LEM. Furthermore, note that the improvement margin of the enhanced CLS Rater is larger for linear regressors (LMS or Linear). Finally, we also explore the scenario of evaluating the enhanced CLS Rater w.r.t. the ground truth ratings. Trained with M5P regressor on the enhanced labels from the maximum disagreement first function has an average rating error of 0.83. Clearly this is an unfair scenario since training and testing are based on different types of labels. However, this relatively small rating error is a good indication that the LEM is indeed updating labels according to disease levels.

We further explore how the regression methods perform w.r.t. different types of appearance features, i.e., color and LBP. As shown in Fig. 15, when learning the regressor with ground truth ratings, fusing color and LBP features improves the system performance for various regression methods. Note that the enhanced CLS Rater also uses the combined color and LBP features. However, M5P and M5Rules perform well using color alone, and fusing with LBP has no noticeable improvement in the rating error. Moreover, when combining color and LPB using the enhanced CLS Rater, all regressors have substantially improved to almost the same high performance, i.e., around 0.6 error rate. In other words, when using enhanced CLS Rater, the choice of regression

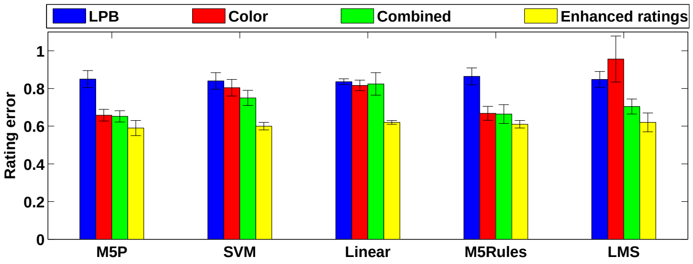


Figure 15: Regression performance with different feature types and labels.

methods is less important, which allows us to use a more efficient and simple regressor, yet still achieving the high performance.

CLS Rater vs. Expert Rating. In general, it takes about five seasons to train an unskilled individual for rating CLS disease and at least one season to train a pathologist. However, it is well known that human experts tend to provide *inconsistent* rating for CLS as discussed earlier. Hence, it is interesting to compare the rating error of CLS Rater to the error observed in human expert rating. The minimum rating error is 0.65 for CLS Rater, and 0.59 for enhanced CLS Rater, as shown in Tab. 2. For comparison, we calculate the standard deviation of expert rating using the *same* equation as our system error e, i.e., $e^h = (\frac{1}{3K}\sum_i \sum_j ||\bar{r}_i - r_i^j||^2)^{\frac{1}{2}}$. Based on the same five partitions in computing e, the standard deviation of expert rating e^h is 1.31 ± 0.08. The superior consistency of our system, i.e., with or without the LEM, over the human experts indicates the great potential of applying CLS Rater in practices.

CLS Rater Across the Years. Ideally the CLS Rater learned from data samples of one year can be repeatedly utilized in the real field in subsequent years. Therefore, it is important to evaluate the generalization capability of CLS Rater on a testing set that is collected from a different year as the training set. For this purpose, we use the 2013 dataset as the training set and the 2014 dataset as the testing set. The labels for the training set is either from one expert (who also labels 2014 dataset), or the enhanced labels by LEM based on the maximum offset first function, which result in the CLS Rater and the enhanced CLS Rater respectively. Similarly, two types of labels exist for the testing set. As shown in Fig. 16, each box represents the manual ratings of the field at a specific day, which is made of 22×46 subunits, where 22 is the number of field lines and 46 is the number of evenly sampled images along each field line. Note that only one expert provides ratings for the four chosen days to record various disease ratings. The second row shows the enhanced ratings after applying LEM using the maximum offset first function.

By applying CLS Rater and the enhanced CLS Rater on the testing set, we obtain the rating results in Fig. 17, respectively. We can see that the CLS Rater was not very successful at predicting very high or low rating in "Sept 3" and "Aug 15", respectively. Moreover, it appears that cultivators located on lines 7 and 8 have relatively higher resistance to the CLS disease in comparison to other lines. An average rating error of the CLS Rater is 1.26 w.r.t. the manual ratings, while an average rating error of 1.05 is achieved for the enhanced CLS Rater w.r.t. the enhanced ratings. Therefore, similar to Tab. 2, we see again that the enhanced CLS Rater provides more consistent rating, even for across-year experiments. The reason for observing higher rating errors than Tab. 2 is twofold: (i) the appearance variation between the years; (ii) the imprecision of mapping GPS data to video frames, and hence assigning manual rating to frames. Nevertheless, the rating error in this challenging across-year experiment is still smaller than the standard deviation of expert rating. Some of the images with predicted labels from the enhanced CLS Rater is illustrated in Fig. 18. Note how the amount of green leaves decrease among four days, while more soil becomes visible. We can also observe the challenging problem of varying weather and light conditions across all days. Our enhanced CLS Rater was successful at assigning the correct rating for all images in this figure.

6. Conclusions

This paper introduced a novel computer vision system, CLS Rater, which uses real field plant images for the automated rating of the CLS disease in sugar beet plants. Our 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 0.59, which

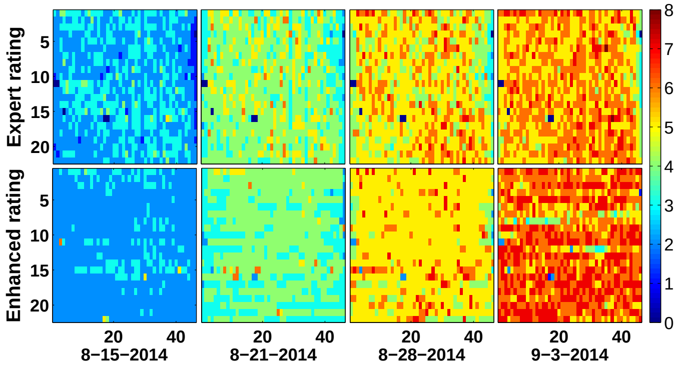


Figure 16: Ratings of a plant field in four days of 2014. The first row shows manual ratings from one expert. The second row shows the enhanced ratings by LEM.

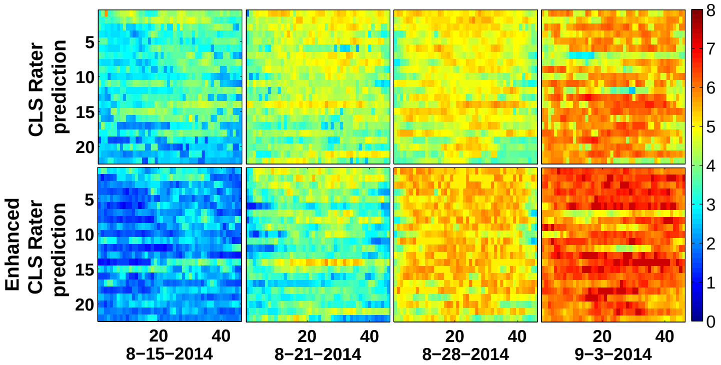


Figure 17: Automatically predicted ratings of the plant field in four days of 2014. The First row is the predictions of the CLS Rater trained on the manual ratings. The second row is the predictions of the enhanced CLS Rater trained on the enhanced ratings.

is substantially more consistent in comparison to manual ratings performed by human experts. We tested our system on a real field of sugar beet plants under different lighting and weather conditions for consecutive two years. We also addressed the issue of the noisy expert labels by developing the LEM to enhance the labels. One future direction is to learn CLS Rater from a set of image pair each *ranked* by their disease severity,

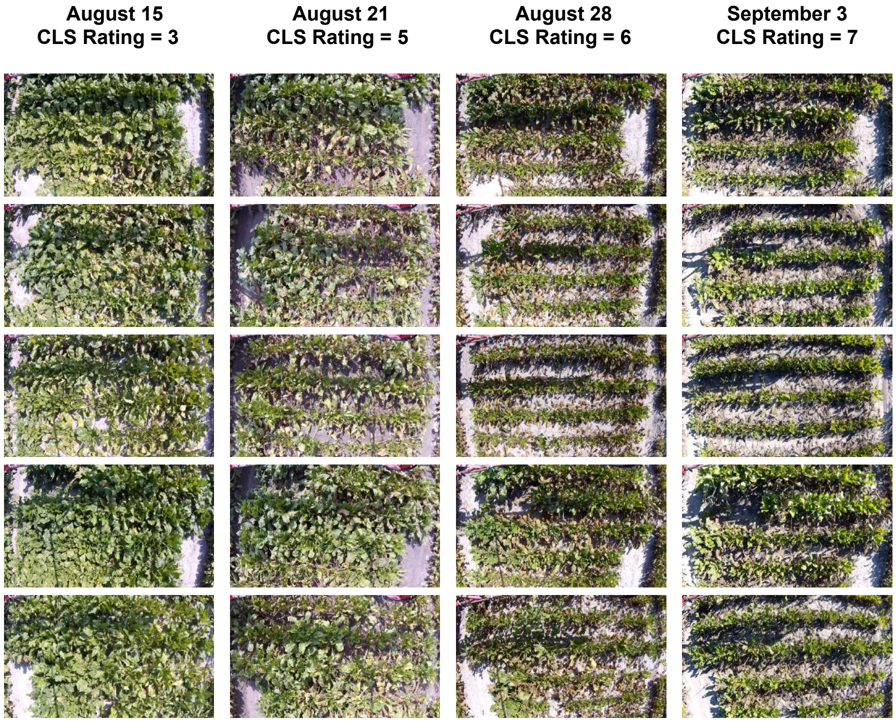


Figure 18: Predicted ratings from the enhanced CLS Rater for images captured in four days. All images in each row have the same GPS coordinates. The predicted ratings in every column are the same for all four days.

using approaches such as boosted rank learning [33]. 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.

7. 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] J. Khan, L. d. Rio, R. Nelson, V. Rivera-Varas, G. Secor, M. Khan, Survival, dispersal, and primary infection site for cercospora beticola in sugar beet, Plant Disease 92 (5) (2008) 741–745.

[3] A.-K. Mahlein, U. Steiner, H.-W. Dehne, E.-C. Oerke, Spectral signatures of sugar beet leaves for the detection and differentiation of diseases, European Conf. Precision Agriculture 11 (2010) 413–431.

[4] S. Bauer, F. Korc, W. Förstner, Investigation into the classification of diseases of sugar beet leaves using multispectral images, European Conf. Precision Agriculture 9 (2009) 229–238.

[5] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, L. Plümer, Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance, Computers and Electronics in Agriculture 74 (1) (2010) 91–99.

[6] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik, Z. ALRahamneh, Fast and accurate detection and classification of plant diseases, Int. J. Computer Applications 17 (1) (2011) 31–38.

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

[8] B. Fulkerson, A. Vedaldi, S. Soatto, Class segmentation and object localization with superpixel neighborhoods, in: Proc. Int. Conf. Computer Vision (ICCV), Kyoto, Japan, pp. 670–677.

[9] J. Tighe, S. Lazebnik, Superparsing: scalable nonparametric image parsing with superpixels, in: Proc. European Conf. Computer Vision (ECCV), 2010, pp. 352–365.

[10] H. Liu, Y. Qu, Y. Wu, H. Wang, Class-specified segmentation with multi-scale superpixels, in: Proc. Asian Conf. Computer Vision Workshops (ACCV), Springer, 2013, pp. 158–169.

[11] M. J. Afridi, X. Liu, J. M. McGrath, An automated system for plant-level disease rating in real fields, in: Proc. Int. Conf. Pattern Recognition (ICPR), Stockholm, Sweden, 2014, pp. 148 – 153.

[12] Z. Li, X.-M. Wu, S.-F. Chang, Segmentation using superpixels: A bipartite graph partitioning approach, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), IEEE, Providence, RI, 2012, pp. 789–796.

[13] Z. Hao, Q. Wang, H. Ren, K. Xu, Y. K. Seong, J. Kim, Multiscale superpixel classification for tumor segmentation in breast ultrasound images, in: Proc. Int. Conf. Image Processing (ICIP), IEEE, 2012, pp. 2817–2820.

[14] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. and Learning Syst. 25 (5) (2014) 845–869.

[15] N. Manwani, P. Sastry, Noise tolerance under risk minimization, IEEE Trans. Cyber. 43 (3) (2013) 1146–1151.

[16] F. A. Breve, L. Zhao, M. G. Quiles, Semi-supervised learning from imperfect data through particle cooperation and competition, in: Int. Joint Conf. Neural Netw., IEEE, 2010, pp. 1–8.

[17] J.-w. Sun, F.-y. Zhao, C.-j. Wang, S.-f. Chen, Identifying and correcting mislabeled training instances, in: Future Generation Communication and Networking, Vol. 1, IEEE, 2007, pp. 244–250.

[18] D. Gamberger, N. Lavrac, S. Dzeroski, Noise detection and elimination in data preprocessing: experiments in medical domains, Applied Artificial Intelligence 14 (2) (2000) 205–223.

[19] S. Verbaeten, A. Van Assche, Ensemble methods for noise elimination in classification problems, in: Multiple Classifier Syst., Springer, 2003, pp. 317–325.

[20] A. Malossini, E. Blanzieri, R. T. Ng, Detecting potential labeling errors in microarrays by data perturbation, Bioinformatics 22 (17) (2006) 2114–2121.

[21] J. Thongkam, G. Xu, Y. Zhang, F. Huang, Support vector machine for outlier detection in breast cancer survivability prediction, in: Advanced Web and Network Technologies, and Applications, Springer, 2008, pp. 99–109.

[22] X. Zeng, T. Martinez, A noise filtering method using neural networks, in: IEEE Int. Workshop Soft Computing Techniques in Instrumentation, Measurement and Related Applications, IEEE, 2003, pp. 26–31.

[23] D. Wang, X. Tan, Robust distance metric learning in the presence of label noise, in: Twenty-Eighth AAAI Conf. Artificial Intelligence, 2014.

[24] M.-Y. Liu, O. Tuzel, S. Ramalingam, R. Chellappa, Entropy rate superpixel segmentation, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), IEEE, Colorado Springs, CO, 2011, pp. 2097–2104.

[25] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Vol. 2, IEEE, San Diego, CA, 2005, pp. 524–531.

[26] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition 29 (1) (1996) 51–59.

[27] I. H. Witten, E. Frank, Data mining: Practical machine learning tools and techniques, Diane Cerra (2005) 187–99.

[28] R. J. Quinlan, Learning with continuous classes, in: Proc. fifth Australian Joint Conf. Artificial Intelligence, Vol. 92, 1992, pp. 343–348.

[29] Y. Wang, I. H. Witten, Induction of model trees for predicting continuous classes, in: European Conf. Machine Learning, 1996.

[30] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, K. R. K. Murthy, Improvements to the SMO algorithm for SVM regression, IEEE Trans. Neural Netw. 11 (5) (2000) 1188–1193.

[31] P. J. Rousseeuw, A. M. Leroy, Robust regression and outlier detection, Vol. 589, Wiley, 2005.

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

[33] H. Wu, X. Liu, G. Doretto, Face alignment via boosted ranking models, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, 2008, pp. 1–8.