Development of a Severity Calculator of a Sweet Pepper Cercospora Leaf Spot Disease Through Digital Image using Marker-Controlled Watershed and Contour Finding Algorithm

Bon Jovi Lapiceros, Michael Anthony Jay B. Regis* & Lucia M. Borines 
Visayas State University, Visca, Baybay City, Leyte, Philippines

Abstract
The study of disease severity on crops would visually gauge the observable inter-related patterns. However, manual assessment of disease severity is subjective, thus decreasing the accuracy of the specified evaluation. Due to this, computer vision is necessary to objectively supplement the human capacity to perform image analysis. This study would develop a disease severity calculator of Cercospora leaf spot on sweet pepper. Eighteen leaf samples of sweet pepper were collected from the field and photographed under fixed camera settings with white background. Digital images were resized to 640 x 480 pixels and pre-processed using the Gimp™ image editor to remove unimportant infections and leaf shadow. Watershed transform was used to extract the foreground (i.e. leaf image) and then the contour-finding algorithm was used to compute the area of the leaf. Afterward, the leaf spots were isolated from the leaf using inverse thresholding and contour-finding algorithm. The area of the isolated leaf spots was then computed. Finally, the severity of the disease was calculated by dividing the disease area over the leaf area. Quantitative performance evaluation utilized the detection percentage, branching factor and receiver operating characteristics (ROC). After testing the system on the specified dataset, experimental results show a detection percentage of 72.46% and branching factor of 37.97%. Moreover, the area under the curve (AUC) of the ROC graph is 0.988 (i.e. >0.5) which means that the study is effective on its detection capability.

Keywords: Watershed transformation; Contour finding; inverse thresholding

Introduction
Sweet peppers are perennials but grow as annuals in temperate climate. They contain an impressive list of plant nutrients that are known to have disease preventing and health promoting properties (Marín, Ferreres, Tomás-Barberán, & Gil, 2004). Early laboratory studies on experimental mammals suggest that alkaloid capsaicin found on sweet pepper has anti-bacterial, anti-carcinogenic, analgesic and anti-diabetic properties (Ludy, Moore, & Mattes, 2012).

The disease spot that is caused by bacterial infection (Cercospora capsici) of the plant during its maturation stage (Jun, Zhiheng, Xinyang, Hong, & Yukun, 2009) is the focus for this study. This disease entails to poor yield of the crop (Cheewawiriyakul, Conn, Gabor, Kao, & Salati, 2006).

Researchers at Plant Disease Diagnostic Laboratory (PDDL) of the Visayas State University (VSU) conducts manual severity assessment of the Cercospora leaf spot disease. This is done by visually identifying, estimating the size then finally counting the specified leaf spot. However, manual assessment is subjective, thus decreasing the accuracy of the specified evaluation (Kreiman, Koch, & Fried, 2002). So, it is necessary to have a computational method that would objectively augment the human capacity to perform image analysis.

Computer vision is a branch of computer...
science that utilizes algorithms to transform still images or video data into either a decision or a new representation (Forsyth & Ponce, 2012). Since digital leaf images are data from a still camera, computer vision can be utilized to perform an objective assessment to address the specified problem.

**Methodology**

This study used the features of OpenCV 3.0 application program interface (API) and Qt version 5.5 development framework in building the application. During image acquisition, a Sony DSC-W620 Digital Camera with a total pixel number of approximately 14.5 megapixels would be used.

**A. Image Acquisition and Preprocessing**

Sweet pepper leaf images were acquired using 14.5 megapixels Sony Digital Camera. A plain white bond paper was placed at the back of the leaf, ensuring an excellent contrast during foreground separation from the background. The camera was adjusted to capture the required leaf images with the background. The original images were resized to 640 pixels horizontally by 480 pixels vertically and exported as .png images. Shadows and any unwanted objects that were taken during image acquisition were cleaned using the Gimp™ image editor in Linux operating system.

**B. Image Processing Procedure**

A marker would be defined from the cleaned image. This marker is referenced on the top, bottom, left, right and center rectangle of the photo. A watershed transformation (Roerdink & Meijster, 2001) would make use of this particular marker to create a boundary between the foreground and the background. The extracted leaf image (i.e. foreground) was thresholded using Otsu’s algorithm (Otsu, 1979). Finally, leaf area (in pixels) were calculated using the contour finding technique (Arbelaez, Maire, Fowlkes, & Malik, 2010). The extracted foreground was then converted to its grayscale representation. The grayscale image was thresholded using binary inverse thresholding. After thresholding, binary erosion would be applied to clean unnecessary noise. The contour-finding algorithm would now segment the leaf spots, count the number of leaf spots and compute the total area (in pixels) of the leaf spot. Figure 1 summarizes image processing procedure of the system.

Detected disease region are displayed in two windows. One is on a background with white foreground region (i.e. disease region) and the other in a Red-Green-Blue (RGB) representation with red-colored contour boundaries as detected leaf spots of the image. Equation 1 calculates the disease severity percentage.

\[
S(\%) = \frac{AD}{AL} \times 100 \quad (1)
\]

where:

- **DS** = disease severity percentage
- **AD** = the number of pixels for the disease
- **AL** = the number of pixels for the leaf

The computed disease severity value is then matched and categorized with the severity value using the disease severity scale (Horsfall & Henberger, 1942) as shown in Table 1:

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>SEVERITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Apparently infected</td>
</tr>
<tr>
<td>1</td>
<td>0 – 25% leaf area infected</td>
</tr>
<tr>
<td>2</td>
<td>26 – 50% leaf area infected</td>
</tr>
<tr>
<td>3</td>
<td>51 – 75% leaf area infected</td>
</tr>
<tr>
<td>4</td>
<td>&gt; 75% leaf area infected</td>
</tr>
</tbody>
</table>

**C. System Evaluation**

Eighteen leaf samples from 3 different varieties were collected at the Department of
Horticulture field area. A visual counting of an expert plant pathologist would act as the source of ground truth for the evaluation. Comparisons were made between manual counting and for automated detection of leaf spots. The following parameters considered were as follow:

- **TP (True Positive)** – an object detected both manually and automatically
- **FP (False Positive)** – an object undetected manually but is detected automatically, also termed as
- **FN (False Negative)** – an object detected manually but is not detected automatically

The performance is evaluated by adopting the performance computations of Lin and Nevatia (1998) which were the following:

\[
DetectionRate(DR) = \frac{TP}{(TP + FN)} \quad (2)
\]

\[
BranchingFactor(BF) = \frac{FP}{(TP + FN)} \quad (3)
\]

The Receiver Operating Characteristic (ROC) would further analyze the significance of the results (Goncalves, Subtil, Oliveira, & Bermudez, 2014). This computation is done by plotting the detection rate (i.e. sensitivity otherwise known as false positive rate) and branching factor (i.e. 1-specificity also identified as true positive rate). The area under the curve (AUC) can be calculated using:

\[
A = \int_{\infty}^{\infty} TPR(T)FPR'(T)dT \quad (4)
\]

Where:

- **TPR** – True positive rate
- **FPR** – False positive rate

When the ROC plot passes through the uppermost left corner (i.e. 100% sensitivity, 100% specificity) a perfect discrimination is concluded. So, the closer the ROC plot is to the upper left corner, the higher the overall accuracy of the test.
Results and Discussion

Watershed segmentation needs marker (i.e. local minima) to avoid over segmentation of the image. The marker (Figure 2) is referenced from the top, bottom, left, right and center position.

![Markers of the image](image1.png)

Watershed segmentation would use this local minima to segment the leaf area from the white background (Figure 3).

![Segmented leaf image](image2.png)

The binary inverse thresholding on the leaf area. Binary erosion would be applied to clean the image since thresholding would produce noise. The structuring element for this operation is a square with a 5x5 pixel size anchored at coordinates (2,2). A contour-finding algorithm would now isolate, count, and compute the area of the detected disease spots (Figure 5).

![Cercospora disease leaf spots](image3.png)

In calculating leaf area, this study implemented the contour finding technique. In which the algorithms loop throughout the extracted foreground image (i.e. leaf green pixel) and finds the biggest contour (i.e. the leaf area). Refer to Figure 4 for the output.

![Leaf Contour](image4.png)

The initial stage to segment disease applies the binary inverse thresholding on the leaf area. Binary erosion would be applied to clean the image since thresholding would produce noise. The structuring element for this operation is a square with a 5x5 pixel size anchored at coordinates (2,2). A contour-finding algorithm would now isolate, count, and compute the area of the detected disease spots (Figure 5).

The final results is displayed in two windows (Figure 6). One represents the isolated leaf spots and the other is the superimposed leaf spots on the leaf. The disease severity information text field displays the total leaf area(in pixels), total disease area(in pixels), total number of spots, severity percentage and the severity category. Eighteen leaf
samples were loaded and analyzed by the system after image pre-processing. Using the knowledge of a plant pathology expert, the true positive (TP), false positive (FP) and false negative (FN) was taken from each test image. Individual detection percentages and branching factor were recorded based on the TP, FP, and FN.

Further analysis shows that the area under the curve (AUC) is 0.988 and a standard error of .013 which means the overall accuracy of the study is high and the system considered is effective.

**Summary, Conclusion, and Recommendation**

A method for detecting leaf spots on sweet pepper digital images using the combination of watershed segmentation and contour finding is done in this study. The system accepts the RGB representation of sweet pepper leaf images. The images were cleaned from any unnecessary objects attached such as shadows captured during image acquisition. The original image was also scaled and resized from its original resolution to 640 x 480 pixels.

Test data are processed in two phases. First, the leaf extraction automatically creates a marker of the image. This marker is used during watershed segmentation to completely separate the foreground object (i.e. green leaf area) from the background. The contour-finding technique calculates the leaf area. Second, a binary inverse thresholding is applied to the leaf then binary erosion reduced the noise. Contour finding method would count the segmented leaf spot and compute the leaf spot area.
### Table 2: Overall evaluation results of the calculator

<table>
<thead>
<tr>
<th>TEST DATA</th>
<th>PERFORMANCE MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N =18)</td>
<td>TP</td>
</tr>
<tr>
<td>Overall</td>
<td>13.89</td>
</tr>
</tbody>
</table>

Most of the time the systems detect more spots than the manual count. The factors attributes to the presence of unrelated objects attached to the leaf image, such as pest manifestation and/or object shadows. However, since the area under the curve (AUC) is 0.988 (i.e. high relative to the cut-off line) the system is considered effective. This makes the system a suitable alternative to the manual method done by the PDDL.

The system cannot completely detect early leaf spots and shadows must be removed manually. We recommend that further studies must be done in order automatically remove shadows using specialized segmentation techniques. Furthermore, machine learning could be employed to classifying the stages of a leaf spots. Finally, for a tidier counting of detected leaf spots, a bounding box within a detected contour should be implemented.

### References


