Automating the segmentation of necrotized regions in cassava root images

Flavia Delmira Ninsiima[1] and Godliver Owomugisha[2] and Ernest Mwebaze[3]
[1,2,3]Makerere University, School of Computing & Informatics Technology
P.O. Box 7062 Kampala, Uganda
[2]University of Groningen
Johann Bernoulli Institute for Mathematics and Computer Science,
P.O. Box 407, 9700 AK Groningen, The Netherlands
[2]Busitema University, Faculty of Engineering
P. O. Box 236, Tororo, Uganda
Email: delmira91@gmail.com, g.owomugisha@rug.nl, emwebaze@cit.ac.ug

Abstract— *The most common form of symptom measurement in assessment of crop health is visual inspection of the plants. In this paper we discuss a prototypical symptom assessment method for assessing the amount of rot or necrosis in the cross-section of a cassava root tuber. Necrosis is largely associated with the presence of the viral disease Cassava Brown Streak Disease (CBSD), the most dangerous viral cassava disease in Sub-Saharan Africa now. Subjective assessment of symptoms tends to be sub-optimal leading to inaccurate estimates. We propose to automate this task by use of computer vision techniques to standardise the assessment of necrosis. Our work extends previous work in this area in two ways; (i) through the development of algorithms that segment out roots from noisy backgrounds and (ii) we provide improved techniques for identifying necrotic regions in the root as well as evaluating how accurate these methods are.*

Keywords: Segmentation, computer vision, image processing, necrosis

1. Introduction

Cassava Brown Streak Disease (CBSD) is the top viral disease devastating farms and gardens in Sub-Saharan Africa. Estimated losses are in the millions of dollars annually [1]. It affects cassava root tubers, the edible part of the cassava plant, causing brown dead areas (necrosis) that make the roots unsuitable for consumption. A key area of intervention is carrying out surveillance visits and breeding of more resistant varieties of the cassava crop.

To carry out the surveillance or breeding tasks, agricultural experts need to be able to access the extend of necrosis in the cross-section of a cassava tuber. The current methods of assessing the presence and severity of CBSD are based on visual inspection of the root tuber of the plant. To assess a cassava plant, the experts will usually uproot a plant and slice off clean disks from its root tuber. A visual assessment of the root tuber then ensues and results are recorded for the field. A huge challenge with this subjective analysis of disease is that the erroneous diagnosis of disease and its severity can have adverse effects on management decisions such as the inappropriate use of pesticides hence the need for automated techniques of symptom measurement.

This paper focussed on the segmentation and identification of necrotic regions from images taken of cassava roots having noisy backgrounds i.e. in the garden. In the sections that follow, we present an image processing pipeline that measures the percentage of necrosis in a cassava root. We structure our work in two phases; (i) the first phase deals with how to identify the best method for segmenting out a root tuber from its background, (ii) the second phase focuses on identifying the best method for estimating necrotic regions in cassava.

The rest of the paper expands on these two phases and is organized as follows. We begin with a section on related work followed by a section which presents the details of the image dataset used. This is followed by a description of the development of the necrosis segmentation pipeline detailing two broad methods of doing root segmentation. Further sections detail methods for necrosis detection, evaluation and deployment of the pipeline. We conclude with a discussion of the results and outlook to future work.

2. Related work

The use of computer vision and machine learning techniques for automatic symptom measurement in plants has been investigated in many contexts. A common feature in most of the work is there is an increase in efficiency and performance of the task being automated. Some researchers have studied the use of crop images to diagnose plant diseases e.g. [2] and [3] where images are used in the diagnosis of wheat rust disease. Another study looked at identification of grape leaf disease [4] using neural networks.

A more related study is the use of images to assess the severity of brown leaf spot disease in cassava in Thailand [5]. As intimated already, most of these studies aimed at increasing throughput and reducing subjectivity arising from human experts in detecting the plant diseases.

A particularly related piece of work that our work builds on is the study by Jovia et. al. [6] where the analysis of necrosis in cassava is done at a pixel level in a very controlled environment with controlled background and lighting conditions. For this paper, we look at assessment of the roots captured in the fields, *in situ*. This is a more challenging task because we have to take care of the background, the lighting effects, occlusion, etc. This however is also a more realistic experiment because the roots are primarily assessed in the field. Later in this work we discuss how to deploy our methods to a smartphone. Our work also presents a better evaluation framework for this problem.

3. Image acquisition

Building a suitable model to solve this task requires a good but realistic dataset, in this case images taken under ordinary conditions of the field. For this study a total of 40 images of cross-sectional cuttings of CBSD infected cassava tubers were obtained from the National Crops Resources Research Institute (NaCRRI), the government organisation mandated to do research in cassava in Uganda, using the camera application of a Techno C5 mobile phone at a resolution of 8.0 Megapixels. The image data set collected consisted of clean roots that exhibited no symptoms (Fig. 1a), and roots that exhibited the necrosis symptoms of CBSD (Fig. 1b).

(a) Clean root (b) Necrotic root

Fig. 1: Examples of clean and necrotic roots collected.

The preferred method of capturing images was to position the camera close to the root so that its cross-section was the focus of the camera lens. Majority of images captured were taken in the field in natural lighting conditions and had a noisy background. For experimental and testing purposes too, a small percentage of the images were captured in a controlled environment i.e, with the background of roots as black cardboard.

4. Development of a necrosis segmentation pipeline

Our goal is to develop a pipeline that can be used to automatically detect a root cross-section from an image of the root with a noisy background and perform a segmentation of the necrotic part of the root providing a score representing how necrotic the cross-section of the root is. The three key phases in this pipeline are; (i) root segmentation from an image of the root *in situ*, i.e. with a noisy background, (ii) segmentation of the necrotic part of an image of the cross-section of the root and (iii) calculation of the area of the root cross-section that is necrotised. Most of the computer vision algorithms used were standard implementations from the OpenCV library [7] or the SciKit-Image library [8]. Details of the different phases of the pipeline follow.

4.1 Root segmentation

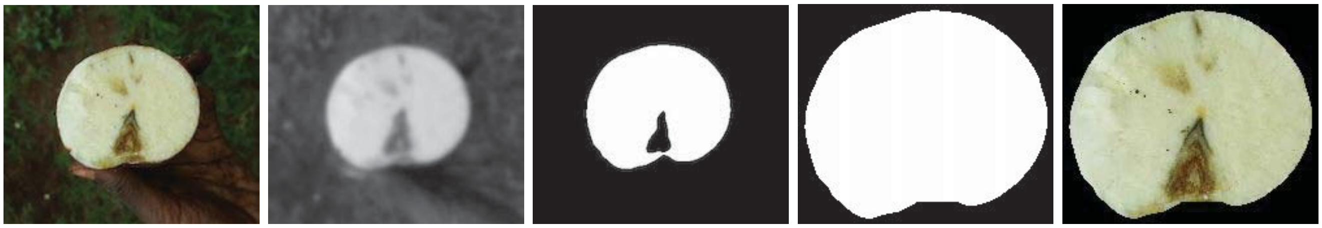
Typically for images of roots taken *in situ*, one has to deal with the noisy background. Our data collection specifically focuses on such images because the down stream usage of this sort of application will be in the field. It is also a harder problem that extends some of the early work [6] which was done with carefully segmented root images.

To state the problem in this phase: given an image of a cross-section of a cassava root we want to be able to accurately segment out the area of the root from the noisy background. To do this we employ two methods, one based on region/blob detection and thresholding and one based on the Watershed algorithm and connected components. In later sections, we evaluate these two methods and contrast their relative strengths. For both methods we apply some preprocessing to resize the images and smooth them using a Gaussian kernel. Resizing is important to improve the efficiency of the method. This is particularly important if one is considering deployment of this application on a mobile device.

4.1.1 Method 1: Otsu algorithm + blob detection

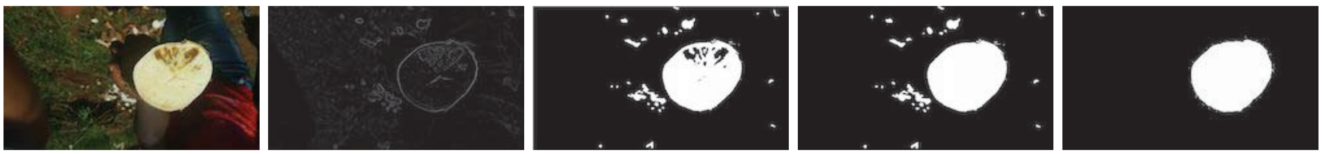
In this method, we rely on the fact that an image of a freshly cut root taken in a garden will have pixel values that are relatively different from the background. We thus apply the Otsu thresholding algorithm [9] to the image to separate the background from the root image and them look for contiguous regions in the image of a particular size that represent the root cross-section. Otsu's method has the advantage that we don't have to manually pick out the intensity threshold to segment upon. Figure 2 illustrates the different steps in this method. It is also enumerated here.

1) Given an image, resize the image based on whether its in portrait or landscape mode



(a) Input image (b) Gaussian smoothing (c) Otsu thresholding (d) Binary closing (e) Segmentation output

Fig. 2: Method 1: Workflow of image segmentation



(a) Input image (b) Sobel edge detection (c) Watershed tranform (d) Binary closing (e) Segmentation output

Fig. 3: Method 2: Workflow for root segmentation.

2) Using a Gaussian filter, smooth the image to remove random noise. (Fig. 2b).
3) Apply Otsu's method to segment out the foreground (root cross-section) from the background. (Fig. 2c).
4) Apply a mathematical morphological operation *binary closing* to remove small areas of inconsistency in the image, which could represent small pieces of dirt in the image.
5) Apply a region/blob detection computer vision algorithm to identify contigous areas in the image and return their areas.
6) Filter out only identified regions where the area is greater than a certain threshold. For this method the threshold was 5000 pixels, the size being influenced by the resizing operation in step 1.
7) Output a binary image/mask (Fig. 2d) where pixels having the value 0 represented the noisy background, and pixels having the value 1 represented our foreground/cassava root(s). This mask, when applied to the original image, has the effect of segmenting out the area of interest, the root cross-section (Fig. 2e).

4.1.2 Method 2: Watershed algorithm + connected components

This method relies on the Watershed algorithm [10] to segment out the root cross-section from the noisy background. The Watershed algorithm considers the image as a topographic surface with the high intensity areas (root) representing the peaks in the image and the low intensity areas (background) representing the valleys. We employ the OpenCV implementation of the Watershed algorithm which users markers of the peaks and valleys to do the image segmentation. From this segmentation, we look for connected components in form of contours and by selecting the largest contour in the image we can segment out the root cross-section from the image. Figure 3 illustrates how this method works. We present the step by step working of the method here.

1) Given an image, resize it based on its orientation.
2) Apply the Sobel filter as a preprocessing step to detect edges, i.e. areas in the image where there is a strong gradient change.
3) Select out markers representing the background and the root(s) based on the extreme parts of the histogram of grey values. These are important inputs to the Watershed algorithm.
4) Apply the Watershed transform to fill regions of the elevation map. The result of this is a clear distinction of the root cross-section. (Fig. 3c).
5) Apply a mathematical morphological operator, *binary closing* to reduce the noise by filling up small areas in the image. Fig. 3d is an illustration of the application of binary closing on image 3c.
6) Extract the connected components, by finding the contours in the image.
7) Select out the contour with the largest area. This will be the root cross-section.
8) Output a mask where pixels having the value '0' represented the noisy background, and pixels having the value '1' represent the foreground/cassava root(s). Applying this mask to the original image results in the segmentation of the root cross-section.

4.2 Identification of necrotic regions

The next stage in our segmentation pipeline is the identification of the necrotic part of the root. Basically the

Fig. 4: Clustering with EM algorithm

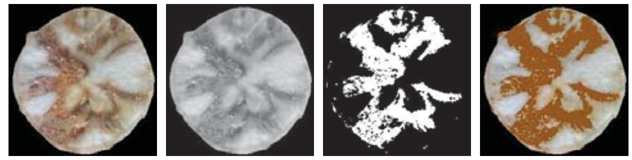
problem here is, given a properly segmented out root cross-section, identify the areas that represent necrosis and those that represent a healthy root cross-section. We employed two methods for this task as well. One based on clustering using the Expectation-Maximization (EM) algorithm and one based on thresholding using the pixel intensities.

4.2.1 Clustering using the EM algorithm

For this method, given an image, we define the number of clusters, $k = 2$ representing the necrotized and the clean areas of the root. We then apply the EM algorithm to segment the image. The EM clustering scheme generates probabilistic descriptions of the clusters in terms of the mean and standard deviation. Using these properties, each of the pixels in the image can be clustered into the two defined clusters. For this problem, the clean part of the root that has mainly white pixels is represented by the cluster with the lowest standard deviation, whereas the necrotic region having brown-black pixels is represented by the cluster with the highest standard deviation. Figure 4 is a depiction of pixels in a root classified as necrotic or non-necrotic using EM Clustering.

4.2.2 Thresholding based on pixel intensity

Given a properly segmented root cross-section, it is possible to define a minimum and maximum intensity that can be used for thresholding the pixels in the image into the clean and the necrotic parts of the root. For this particular problem, by empirically examing the images in the sample we collected, we were able to determine the minimum and maximum thresholds as 0.1 and 0.57 respectively. Any pixel whose intensity value was in the range of the min and max was identified as being necrotic. The output of thresholding is a mask whose white pixels represent necrotic areas and black pixels represented non-necrosis. The image depicted in Figure 5c shows the output of thresholding the floating point image 5b using the specified minimum and maximum thresholds.



(a) Input root (b) Grayscale (c) Output mask (d) Detection

Fig. 5: Thresholding based on pixel intensity.

4.3 Scoring necrosis

The overall goal of this paper is to provide a better, automated and non-subjective method of segmenting out necrotized areas of a cassava root and providing a score that represents the level of necrosis. This is very important for breeders who are developing disease resistant varieties and need to cross different necrosis resistant varieties of cassava. The current method is to assign a subjective score in the range 1 to 5 representing how necrotized the root is. Here we provide a more robust score which is a percentage score of the number of pixels in the root cross-section that are determined to be necrotized.

To calculate this area, we calculate the average of the total number of pixels in the cross-section of the root that represent the necrotized area of the root. The number of pixels representing the necrotized area of the root are obtained from the previous methods; clustering with the EM algorithm or thresholding with the min and max intensity values.

5. Evaluation of pipeline

Our pipeline consists of two distinct phases and as such to evaluate it, we evaluate the individual phases and using this evaluation select out from the different methods proposed, the one that performs best. This section thus presents evaluation of the root cross-section segmentation from an image with a noisy background and evaluation of the accuracy of detecting the area of necrosis from a segmented out root cross-section. To do this, we selected a total of 20 new images and evaluated our proposed methods on these 20 images compared with manual ground truth data obtained by manually segmenting out the root and the necrotized area of the root.

5.1 Root segmentation evaluation

Here we want to determine how accurate our methods identify a root cross-section in an image with a noisy background. For the 20 test images, we obtained ground truth data by manually segmenting out the roots and the necrotized areas using PhotoScissors software[1]. For each of the 20 images the ground truthing produces two outputs, a

[1]https://www.photoscissors.com

ISBN: 1-60132-485-5, CSREA Press ©

mask representing the area of the image that is the root cross-section and a contour representing the boundary between the foreground and the background.

To evaluate segmentation, three differently oriented methods were used. Each method takes as inputs two segmented images, in our case the ground truth segmentation and the segmentation from our methods, and returns a score that represents how well our segmentation matches the ground truth. The three evaluaton methods include.

(a) **Boundary Displacement Error (BDE):** This method [11] measures the average displacement error of one boundary pixel in one segmentation and the closest boundary pixels in the other segmentation. The error of each boundary pixel is defined as the distance between the pixel and the closest pixel in the other boundary image. The smaller this value, the better the performance of the segmentation technique.

- **Global Consistency Error (GCE):** This method [12] measures the extent to which one segmentation can be viewed as a refinement of the other. Segmentations which are related in this manner are considered to be consistent, since they could represent the same natural image segmented at different scales. It takes two segmentations as input, and produces a real valued output in the range [0,1] where zero signifies no error. The smaller the value for GCE, the better the performance. The GCE score is obtained using this formula:

$$GCE(S_1, S_2) = \frac{1}{n} min\{\sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i)\}$$

where E is a segmentation error measure and S_1 and S_2 are the two input segmentations.

- **Jaccard Index (JI):** This method measures the region coincidence between the segmentation result and the ground truth. It is defined as the intersection between two sets divided by their union. It outputs a score in the range $[0, 1]$, the larger the value, the better the performance. If region S_1 is the ground-truth foreground object and the region S_2 be the foreground segment derived from the segmentation result, region-based segmentation accuracy can be defined as:

$$JI(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

The same operation is repeated on the ground-truth background segment and the region S_2 of the background segment derived from the segmentation result, and an average value for the JI is gotten.

5.2 Detection of necrosis

To evaluate how well our methods are doing on the necrosis identification task, we compare pixels identified as necrotic in the ground truth and those identified by our methods. Using the ground truth data from the 20 images we calculate for each pixel in the image how many of those are identified correctly as necrotized and how many are identified wrongly. Using these values the precision and recall of the different methods can be calculated and a more definitive score the Area Under the Curve (AUC) is obtained. We apply this measure for the 20 roots and average over them to get a single score per method employed, i.e. clustering or thresholding based on pixel values.

6. Results

In this section we present results of using our segmentation pipeline on a test set of 20 cassava root cross-sectional images taken *in situ*. We present results for the segmentation phase of the pipeline as well as for the necrosis identification phase of the pipeline. The results depict how well our methods perform compared to a carefully created ground truth dataset, where we manually segment out the areas of interest from the test images.

Table 1 shows the performance of our root segmentation methods in identifying a root cross-section in an image with a noisy background. Particularly it shows the BDE, GCE and JI values that represent how well the segmentation happens in each case. The table shows method 1 out performing method 2 for the segmentation task on average. We notice for some particular images, method 2 has better performance however.

Table 1: The BDE, GCE and JI scores of each model on 20 test images. Bold figures show winning score.

	BDE		**GCE**		**JI**	
Image	*Method1*	*Method2*	*Method1*	*Method2*	*Method1*	*Method2*
1	4.3	**3.85**	**0.05**	**0.05**	**0.94**	0.93
2	3.23	**2.91**	**0.03**	0.04	**0.95**	0.94
3	**3.96**	5.13	**0.04**	0.05	**0.94**	0.92
4	**1.7**	2.38	**0.01**	0.02	**0.96**	0.94
5	**1.25**	31.24	**0.01**	0.11	**0.97**	0.67
6	3.63	**3.08**	**0.04**	**0.04**	**0.96**	0.95
7	**2**	2.13	**0.01**	0.02	**0.95**	0.94
8	**2.06**	255.44	**0.01**	0.02	**0.94**	0.47
9	**1.86**	1.92	**0.02**	**0.02**	**0.96**	0.95
10	2.59	2.41	0.04	**0.03**	0.91	**0.95**
11	**1.73**	1.96	**0.01**	**0.01**	**0.96**	0.93
12	**1.62**	2.05	**0.01**	**0.01**	**0.95**	0.92
13	4.31	**4.24**	**0.05**	0.06	**0.94**	0.93
14	**1.51**	2.11	**0.01**	0.02	**0.97**	0.95
15	2.56	**2.12**	**0.02**	**0.02**	**0.95**	0.94
16	2.73	**2.55**	**0.03**	**0.03**	**0.95**	0.94
17	**2.64**	7.03	**0.02**	0.04	**0.95**	0.91
18	**1.64**	4.17	**0.01**	0.02	**0.95**	0.89
19	**3.36**	13.68	**0.03**	0.07	**0.95**	0.89
20	2.29	**2**	0.03	**0.02**	**0.95**	**0.95**
Mean	**2.55**	**16.12**	**0.023**	**0.035**	**0.95**	**0.896**

For the necrosis identification task, we compare pixels from the ground truth with those from our methods. We obtain the AUC as an average over the 20 images for each method. The thresholding method based on pixel intensity obtains an AUC of 0.94 while the clustering algorithm based on EM obtains an AUC of 0.93. On average we notice a slight advantage in the performance of the thresholding method based on pixel intensity values compared to the clustering method with the EM algorithm.

Empirical evaluation of the methods was also done using hold out test images. These images were of three categories, a healthy/clean root, a dirty/muddy root and a CBSD infected root. The goal is to run our pipeline on these images and visually inspect if they are doing the right thing.

Analysis of a clean root

For a clean root, we expect our pipeline to return a score close to 0 %. For this experiment, the image shown in Figure 6 was subjected to the two necrosis detection algorithms, thresholding and clustering. The thresholding method returns a score of 0.03 % while the clustering algorithm returns a score of 0.9 % indicating again that the thresholding algorithm is superior.

(a) Clean root (b) Thresholding=0.03% (c) Clustering=9.0%

Fig. 6: Empirical scoring of a clean root cross-section.

Analysis of a dirty/muddy root

Here we are interested in understanding if the different methods will be able to distinguish between dirt and necrosis. This is an important quality because very often in the process of harvesting and cutting roots in the field, dirt may get onto the root cross-section. We expect the ideal score to be less than 1% in this case. Figure 7 depicts the results for doing this experiment with both methods of necrosis identification. Both algorithms get above 1% however with the thresholding algorithm getting a much smaller score of 1.31% compared with 18.0% for the clustering algorithm.

(a) Dirty root (b) Thresholding=1.31% (c) Clustering=18.0%

Fig. 7: Empirical scoring of a dirty/muddy root cross-section.

Analysis of a CBSD affected root

For a CBSD affected root with clear display of symptoms we expect a proper segmentation and a non-zero percentage score. Figure 8 shows results for applying thresholding and clustering methods to this problem. Again we observe the thresholding method obtaining a result closer to the ground truth.

(a) CBSD root (b) Threshold=12.97% (c) Clustering=34.0%

Fig. 8: Empirical scoring of a CBSD affected root cross-section.

7. Deployment

The problem of determination of Cassava Brown Streak Disease (CBSD) which causes this necrosis is an important problem for several regions of Sub-Saharan Africa. This paper highlights work that was carried out with a view of deploying the outcome methods so that breeders and other agricultural workers working to intervene on this disease can have better and more standardized tools. A key aspect of this work is how to deploy the solution in such a way that it can be easily accessed.

Fig. 9: Smartphone deployment of tool

To this end, we prototyped three deployment options for this tool; as a standalone desktop tool, as a smartphone app shown in Figure 9, and as a web application. The different deployments offer slightly varying functionalities and are targetted for different use cases of such a tool. Specifically the mobile application is more suited for people collecting data in the field that need to get a quick understanding of the state of health of the crops, these would typically be regular surveillance personnel. We also have the more accurate desktop version of the tool that allows interactively with the tool to improve the accuracy of the system, a deployment option more suited for breeders who need more accurate and detailed analysis of different cassava varieties. The web version allows anyone with images to uplaod them and receive a scoring of the images. This can be used by anyone doing research in necrosis determination.

8. Discussion

This paper has presented an non-subjective pipeline for automating the segmentation of necrotized parts of the cross-section of a cassava root tuber. This work builds on previous work where images used were carefully curated and the evaluation was not as rigorous. In this previous work it was shown that automated methods outperformed the current human methods of analysis, where even experts in the field could not agree on a score for a particular root cross-section. This work provides a more accurate method and presents several deployment options for the developed pipeline. We also evaluate different methods for executing the different phases of the pipeline.

Results in this paper are drawn from a test set of 20 images which were carefully manually segmented so that a proper ground truth can be obtained. The evaluation points us towards a set of methods that provide on average better performance than the alternate methods presented. However, we notice that for some of the test images, the alternate methods provide better performance. This inconsistency while problematic is also expected. Doing image analysis based on tehcniques that have to determine decision boundaries based on pixel values is inherently problematic because of the occlusion that naturally abounds when dealing with images. In our experimental set up we try as much as possible to relate the conditions underwhich the images are taken to the actual field conditions under which a technician using our tool would take the same images. This offers some robustness to our methods, however we are cognizant of the inherent dependence of the performance of the algorithms on the nature and context around which the image was taken.

On the performance of the actual methods, we notice that the threshold based methods tend to outperform the alternate methods albeit only slightly in some cases. Threshold methods tend to work best with good preprocessing of the images. The silver lining here is that these methods would be guaranteed to perform well if the protocol for taking images was made more rigorous.

9. Conclusion

The work presented in this paper offers an alternative method for scoring necrosis in cassava roots. We attempt to show its superiority to existing methods presently. We have also prototyped the different deployment options for this pipeline. However the evaluation on 20 images is a bit limited. Future work will involve evaluation on more images which could be picked more deliberately to reflect the different contexts under which images can be captured for example images with severe occlusion, images of varying scale levels, etc.

Acknowledgement

We would like to acknowledge the team at the National Crops Resources Research Institute, NaCRRI in Uganda who provided the experimental data for this work. We also acknowledge our funders, the Bill and Melinda Gates Foundation who provided the grant for this work.

References

[1] E. Nuwamanya, Y. Baguma, E. Atwijukire, S. Acheng, and T. Alicai, “Competitive commercial agriculture in sub saharan africa,” *International Journal of Plant Physiology and Biochemistry*, vol. 7(2), pp. 12–22, 2015.

[2] J. K. Patil and R. Kumar, “Advances in image processing for detection of plant diseases,” *Journal of Advanced Bioinformatics Applications and Research*, vol. 2, no. 2, pp. 135–141, 2011.

[3] D. Majumdar, D. K. Kole, A. Chakraborty, and D. D. Majumder, “An integrated digital image analysis system for detection, recognition and diagnosis of disease in wheat leaves,” in *Proceedings of the Third International Symposium on Women in Computing and Informatics*. ACM, 2015, pp. 400–405.

[4] S. S. Sannakki, V. S. Rajpurohit, V. Nargund, and P. Kulkarni, “Diagnosis and classification of grape leaf diseases using neural networks,” in *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*. IEEE, 2013, pp. 1–5.

[5] K. Powbunthorn, W. Abudullakasim, and J. Unartngam, “Assessment of the severity of brown leaf spot disease in cassava using image analysis,” in *The International Conference of the Thai Society of Agricultural Engineering, Chiangmai, Thailand*, 2012.

[6] J. Tuhaise, J. A. Quinn, and E. Mwebaze, “Pixel classification methods for automatic symptom measurement of cassava brown streak disease.” Citeseer.

[7] G. Bradski, “The OpenCV Library,” *Dr. Dobb's Journal of Software Tools*, 2000.

[8] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in python,” *PeerJ*, vol. 2, p. e453, 2014.

[9] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[10] J. M. Gauch, “Image segmentation and analysis via multiscale gradient watershed hierarchies,” *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 69–79, Jan 1999.

[11] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, “Yet another survey on image segmentation: Region and boundary information integration,” in *European conference on computer vision*. Springer, 2002, pp. 408–422.

[12] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.