International Journal of Engineering Technology, Management and Applied Sciences

www.ijetmas.com April 2017, Volume 5, Issue 4, ISSN 2349-4476

Image Processing For Weed Detection

R.Anirudh Reddy
Assistant Professor,
B.V.Raju Institute of Technology, Narsapur, Medak.

G.Laasya,
PG Student,
B.V.Raju Institute of Technology, Narsapur, Medak.

T.Sowmya, UG Student,
B.V.Raju Institute of Technology, Narsapur, Medak.

P.Sindhuja, UG Student,
B.V.Raju Institute of Technology, Narsapur, Medak.

Mudasar Basha
Assistant Professor,
B.V.Raju Institute of Technology, Narsapur, Medak

Abstract: This paper introduces the system implementation of image processing technique for weed detection and removal. It involves simple edge detection technique using various filters such as Gaussian and Laplacian. It finally concludes with the feature extraction results that implement ORB algorithm. An RGB image is taken as a sample in order to demonstrate the difference between weed and the crop. This RGB image is further processed for detecting the weeds. We used Python 3.4.1 version for processing the sample image. After certain steps, we get an output where the weeds are separated from the crop that has been taken in the sample image.

Keywords: Python, Image Processing, Feature extraction, Laplacian filter, ORB algorithm, weeds.

I LITERATURE SURVEY

Faisal Ahmed et. Al., have investigated the use of support Vector Machine (SVM) and Bayesian classifier as machine learning algorithms for the effective classification of crops and weeds in digital images. From the performance comparison, it is reported that SVM classifier has outperformed Bayesian classifier. Young plants that did not mutually overlap with other plants are used in the study. Robert Bosch designed a system for weed detection which runs with the help of solar panels for power and uses a camera which is fixed at the bottom for continuous processing of the captured images. This is implemented in the fields of Germany. In the Eastern European countries, students have developed the robot for crushing the weeds as and when detected. Countries like China, Japan are under the process of developing a system which sweeps off all the unwanted materials like weeds, pebbles and stones.

II INTRODUCTION

In olden days weed detection was done by employing some men, especially for weed removal purpose. They will detect the weeds by checking each and every plant field. Then they will pluck them out manually using their hands or spades. Later with the advancement in the technology they started using the herbicides to regulate the growth of the weeds. But to detect the weeds they are still using manual power in many parts of the world. Weeds are the plants growing in a wrong place which compete with crop for water, light, nutrients and space, causing reduction in yield and effective use of machinery and can cause a disturbance in agriculture. Weeds can also host pests and diseases that can spread to cultivated crops. We are using Image processing technique for detecting the weeds and by Image processing, we extract the features that distinguish between crop leaves and weed leaves.

III SOFTWARE REQUIREMENT

The software tool we used here is Python3.4.1. Python is one of the prominent languages used for Image processing. It includes certain packages that make Image processing easy to implement. We took help of OpenCV which is an open platform for certain programming languages like C, C++, JAVA, Python. We have installed certain packages such as Numpy, PyWavelet, Matplotlib, etc. All this work was done on a Windows operating system with an inbuilt Microsoft Visual Studio (any version, here 2015).

IV BLOCK DIAGRAM

This system performs the following tasks (refer to figure 1). The important task here is to identify the weeds in the taken image.

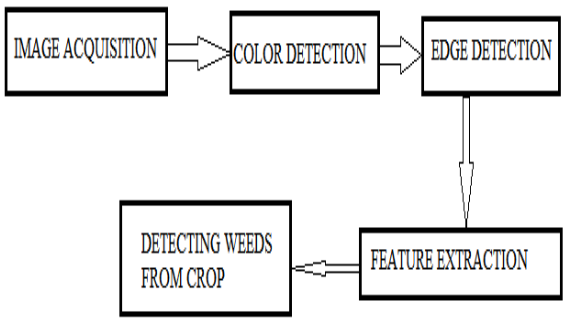

Figure 1: Block diagram

V DESIGN FLOW

Design flow explains the various steps followed in the process of achieving the final output with the help of certain algorithms that generates accurate results. The following design flow shows the order of the analyses that are observed on the image to determine the absolute results.

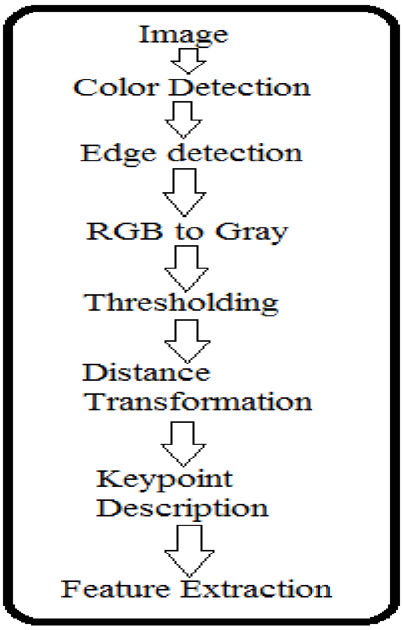

Figure 2: Design Flow

VI SYSTEM DESIGN AND IMPLEMENTATION

As mentioned earlier, the first step here is acquiring an image. Here, we considered a field image in a cotton field with top view in a broad day light.

Figure 3: Sample Image

The weed leaves and the crop leaves are green in color. So, color detection cannot be used here. That is why, we opted for edge detection. But, before going to edge detection we need to separate the green colored part from the image. For this, we need to adjust the HSV values which are specific for each color in HSV table. Masking is the process to setup required Hue, Saturation values for obtaining the required colored part in an image. HSV values in general, ranges from 0 to 255, 0 to 360 in HSV table. Opencv takes saturation values from 0 to 180 (half the original HSV values). We adjusted the saturation values for separating the green colored part in an image.

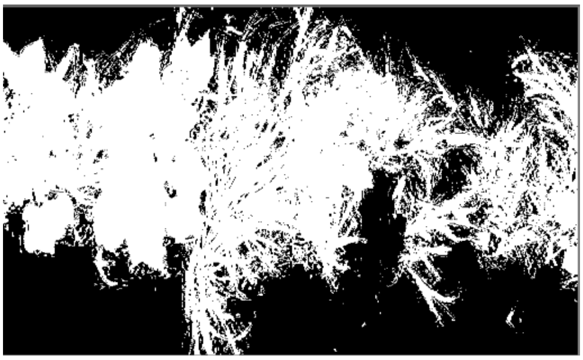

Figure 4: Masked Image

After this the green colored part is obtained leaving behind the unnecessary soil part. Thus, the obtained green colored part of the image is used for further analysis.

Figure 5: Image after Color Detection

As mentioned earlier, since the color detection cannot be utilized, we went for edge detection. We have many edge detection filters such as, Canny, SobelX, SobelY, Laplacian, etc. After verifying the edge detection filters we opted for Laplacian filter which gave the following result, But, before passing an image through Laplacian filter the noise in an image must be removed. This is done using Gaussian blur. It not only removes the noise, but also smoothen the image. The Gaussian blur is a type of image blurring filter which can be used in one or more than one dimensions.

Figure 6: Image after Edge Detection

The equation of Gaussian functions in two dimensions and one dimension respectively are as follows:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Thresholding is the next major step in this process. It is the simplest method of image segmentation which is basically used to create binary images. Thresholding must be done on gray images in order to attain high efficiency.

Figure 7: Image after RGB to Gray Conversion

The various thresholding algorithms that can be implemented in Python using OpenCV are Watershed algorithm, Otsu's algorithm, etc. Otsu's algorithm performs cluster based image reduction where as Watershed goes for linear reduction. For computational efficiency, we used Otsu's algorithm. It assumes that image contains two classes of pixels following bimodal histogram. Histogram refers to a bar diagram showing different pixel intensity values. It then calculates the optimum threshold separating the two classes so that their class variance is minimal or inter class variance is maximal.

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Where w_0 and w_1 are the probabilities of two classes while σ_0 and σ_1 are the variances of two classes.

The following images represent the sure foreground and sure background of the image. Where, sure background is unwanted. So, the further analysis is completely dependent on the sure foreground image.

Figure 8: Sure foreground image

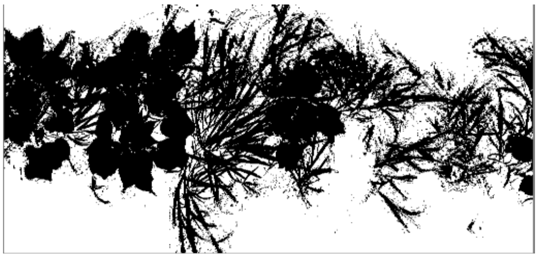

Figure 9: Sure background image

The sure foreground image is distance transformed. The distance transformation is an operator normally applied only to binary images. The result of this transform is a gray level image that looks similar to input image except that the gray level intensities of points inside the foreground regions are changed to show the distance to the closest boundary from each point.

Figure 10: Distance transformed image

We now take the help of keypoints for feature extraction. Keypoints acts as accurate points of interest which helps in determining the feature of a desired object. This keypoint identification is a key step because, in real time the image may get rotated, shrink, translated, or subject to distortion. Python provides a simple command for this keypoint identification in which certain parameters have to be adjusted as per the requirement.

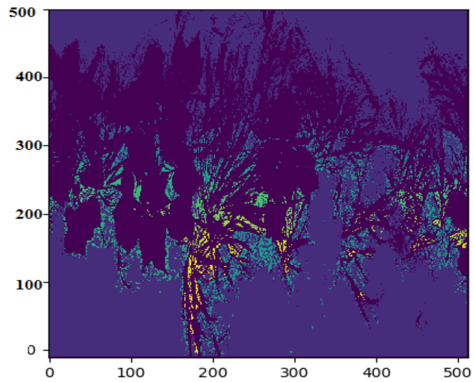

Figure 11: Image after Keypoint Description

The final and the crucial step here in weed detection is the feature extraction. The prominent feature extraction algorithm in use are SIFT (Scale Invariant Feature Transform) and SURF (Speeded Up Robust Features). But, the main drawback with these algorithms is highest computation time. FAST (Feature Accelerated Segmented Test) is a matching algorithm which performs image matching in a linear manner i.e., feature matching cannot be done with orientation but, it is famous for its high speed. Similarly, BRIEF (Binary Robust Independent Elementary Features) is feature extraction algorithm which is fast in both building and matching. Combining the desirable features of both FAST and BRIEF, Python supports ORB algorithm which stands for Oriented FAST Rotated BRIEF. This ORB algorithm performs feature extraction along with orientation. It computes the intensity weighted centroid of the patch with located corner at centre.

The direction of the vector from this corner point to centroid gives the orientation.

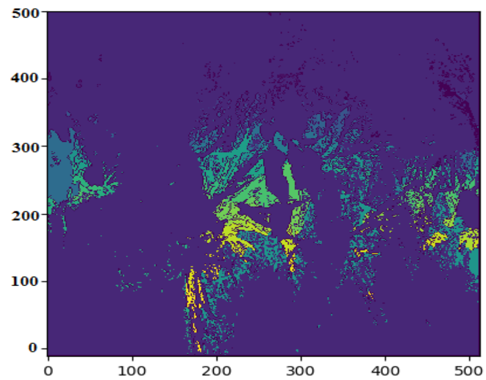

Figure 12: Feature Extracted Image

www.ijetmas.com April 2017, Volume 5, Issue 4, ISSN 2349-4476

To improve the rotation variance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the path.

VII RESULTS

The sample image and the oriented final image with detected weeds are as follows.

Figure 13: Sample Image taken with weed leaves

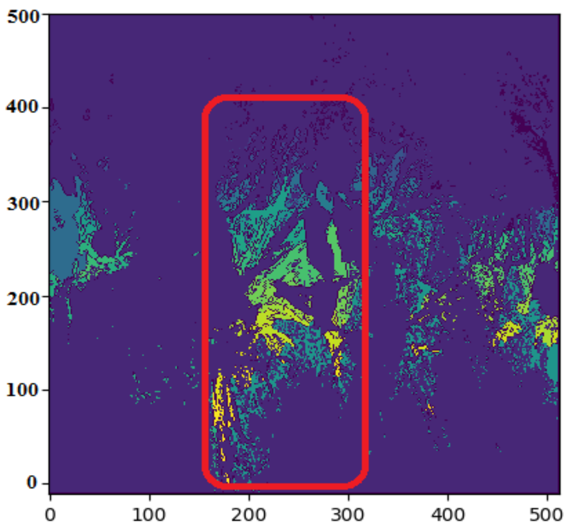

Figure 14: Image after weed detection

VIII CONCLUSION

Prior work has documented the high accuracy considering various other parameters such as texture, genes, etc. Weed detection is an important factor for their removal and regulation. Appropriate weed detection algorithm must be used to avoid damaging the crop plants. The proposed system considers the simple edge detection algorithms. In reality, the proposed system may compromise in accuracy but not in efficiency.

IX REFERENCES

[1] Florian Knoll, Tim Haltorf and steephan Hussman, "Vegetation Index Determination method based on Color room processing for weed control applications in Organic Farming", IEEE.

[2] Faisal Ahmed, A.S.M. Hossian Bari, Emam Hossian, "Performance analysis of Support Vector Machine and Bayesian Classifier for Crop and Weed classification from digital Images", World Applied Sciences Journal, 2011.

[3] Riya Desai, Kruti Desai, Zinal Solanki, "Removal of weeds using Image Processing: A Technical review" in International Journal of Advanced Computer Technology (IJACT).

[4] Latha, Poojith, Amarnath Reddy, "Image Processing in Agriculture" in International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, June 2014.

[5] Florian Johannes Knoll, Tim Holtrof, Stephen Hussmann, "Investigation of Different Sensor Systems to Classify Plant and Weed in Organic Farming Applications".

[6]Tsampikos Kounalakis, Georgios A. Triantafyllidis, Lazaros Nalpantidis, "Weed Recognition Framework for Robotic Precision Farming" in IEEE 2016.

[7] Keun Ha Choi, Sang Kwon Han, Kwang-Ho Park, Kyung-Soo Kim and Soohyun Kim, "Vision based Guidance Line Extraction for Autonomous Weed Control Robot in Paddy Field" in IEEE Conference, Zhuhai, China, December 2015.