

Autonomous Herbicide Spraying System using AI and IoT

Anirudha S Tadpatri ¹
Electronics and Communication
Engineering
Jyothy Institute of Technology
Bengaluru, India

Prajwal B ¹
Electronics and Communication
Engineering
Jyothy Institute of Technology
Bengaluru, India

Pramod P Ugargol ¹
Electronics and Communication
Engineering
Jyothy Institute of Technology
Bengaluru, India

Rakesh Shridhar ¹
Electronics and Communication
Engineering
Jyothy Institute of Technology
Bengaluru, India

Ajjaiah H B M ¹
Electronics and Communication
Engineering
Jyothy Institute of Technology
Bengaluru, India

Abstract - The solution developed in this paper addresses the common health hazards that are generally seen in the traditional method of spraying herbicides. To maintain the good health of the plants as they grow by eradicating unnecessary plants or weeds, it is important to spray herbicides and other chemicals effectively so that they will not hinder the growth of the plant. The solution to this is by building and modeling an Autonomous Herbicide Spraying System using Artificial intelligence and the Internet of Things (IoT). This paper emphasizes the method of filtering Global Positioning System (GPS) data using a moving average filter to improvise on the autonomous navigation system of the robot. Furthermore, the robot is capable of evaluating the count of weeds whilst spraying the herbicides when the threshold is crossed. Additionally, the robot incorporates a liquid-level sensor for measuring the amount of herbicide in real-time and transmitting the same using IoT.

Keywords: Smart agriculture; Autonomous navigation; GPS data filtering; Image processing; IoT; YOLOv5; Moving average filter.

I. INTRODUCTION

Technological advancements in the field of agriculture has increased exponentially over the Past years , which has led to rise in new scientific field improving the precision and the agricultural productivity using the data effectively ,keeping the impact on the environment to minimum [1]. Agricultural practices have been made autonomous and the navigation system plays a pivotal role in making route decisions for an autonomous vehicle. In order to achieve autonomous navigation in this application, a basic differential-drive robot is designed using Global Positioning System (GPS) and magnetometer sensors which will work in tandem to make the robot reach the desired destination. GPS sends the coordinates and way points to the receiver in the robot in a prefixed order. When there is a clear optical axis, the receiver is connected to four GPS satellites. The latitude and the longitude coordinates received are said to have accuracy of about 2.5m - 3m under

clear environmental conditions [2]. TinyGPS plus Arduino library is used for reading the GPS data via the serial port [3]. Since the robots available in the market currently are limited by their high initial cost, an alternative low cost means for agriculture application is necessary [4].

Image processing comes under signal processing and it is commonly is used to perform several modifications to an image to enhance the image or to extract the required data or information from the image captured. The process is straightforward beginning from the image acquisition through the camera to using machine vision systems like deep learning to process the image acquired to obtain the needed information from the same [5]. Image processing is done using the steps given below:

- i. Image acquisition through image acquiring tools.
- ii. Recognizing and manipulating the image captured.
- iii. Output of the same can be a modified image or extracted information from it.

Analog image processing and digital image processing are the two types of image processing techniques used in signal processing domain. The former one is used in photographs and printouts while the latter is generally used in digital computers. There are 3 main steps in any image processing technique. They include pre-processing, enhancement of images and depiction and extraction of information. This paper focuses on the You Only Look Once (YOLOv5) technique and the results of the same are discussed. The images captured by the acquisition tools should be pre-processed to segment the images. This is called image segmentation. Any image processing algorithm incorporates normalizing process, resizing of images, colour space transforming process, enhancement of contrast and eliminating the noise. [6].

II. RELATED WORK

Autonomous herbicide spraying system is an agriculturally based robot that makes use of image processing techniques and autonomous navigation algorithms to navigate itself down the rows of a field by spraying herbicides effectively only based on the count of weeds detected in a particular image frame

captured by the camera. This section discusses about the algorithms used in the two main areas of interest and also the transmission of real time data using IoT.

A. NAVIGATION SYSTEM

The agriculture robot is expected to cut down some of the major traditional routines performed by the farmers in the agriculture field, it would also improve their task’s momentum and its preciseness [7]. Fig. 1 depicts the pictorial representation of the navigation system. The main components that constitute the navigation system are the controller board, GPS and compass modules for retrieving GPS coordinates and heading data, motor driver and motors for navigation. The liquid level sensor is also interfaced to track real time data such as the herbicide level and sends it across to a developed android mobile application. Based on the data acquired from the GPS and compass modules, the controller shall be controlling the motors attached to the wheels for autonomous navigation.

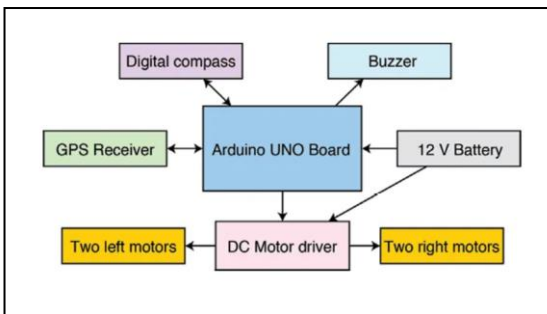


Fig. 1. Navigation system block diagram contains all the necessary components that are required for autonomous navigation of the robot.

The working flow of the agriculture robot is displayed in Fig. 2. In order to achieve autonomous navigation, amongst many things, distance and angle play a major role. Since GPS deals with latitude and longitude points, normal distance formulae like the Euclidean distance formula will not provide desired results. Therefore, haversine formula is being used in order to determine the spherical length to reach the destination. Meanwhile, heading angle formula is used to determine the angle made between the two points using magnetic north as reference. If the spherical length returned by the haversine formula is found to be large compared to the pre-set distance error of 1m; the robot should be instructed to move in the forward direction. In addition to this, if the angle made by the robot with respect to the destination point is found to be exceeding the heading tolerance error of ± 5 degrees, the robot should be instructed to change its course and rotate towards the desired destination and traverse accordingly until it reaches latest destination point. The steps should be repeated until the last waypoint is encountered and should stop when there is no waypoint available in the waypoint array. The agriculture robot should calculate the route data at every instance. because distance and angles keep changing with the slightest of movements of the robot and thus the maneuvers should be done accordingly.

Distances generally vary when we move from a 2D surface to a 3D surface. Conventional distance formula holds good for distances that are measured on a 2D plane while Haversine formula is used to measure distances on a 3D surface. The 3D surface in this context is Earth surface [8]. Spherical distance

of any two points on the face of the earth can be evaluated by considering the latitude and longitude of the respected points selected. Haversine formula has two forms – Generalized Haversine and Haversine that employs trigonometric identity. [9].

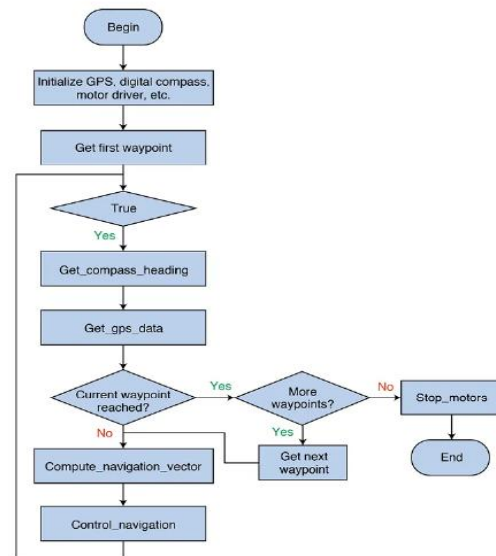


Fig. 2. Navigation Algorithm gives the flow of acquiring the GPS coordinates to successfully navigating the robot autonomously.

The radial distance or the angle between two lines can be calculated using the heading angle formula. Heading angle formula for calculating the heading is given in equation (1) and coordinates of the target point and the position of the robot are needed. The heading angle is denoted by θ.

$$\theta = a \cdot \tan 2(\sin \Delta \lambda \cdot \cos \phi 2 \cdot \cos \phi 1 \cdot \sin \phi 2 - \sin \phi 1 \cdot \cos \phi 2 \cdot \cos \Delta \lambda) \quad (1)$$

The heading error is another parameter that needs to be considered. It is defined as the difference between the heading angle and the direction of robot in which it is moving with reference to magnetic north. The data is obtained from a magnetometer. Equation (2) represents this mathematically.

$$\text{Error_heading} = \text{Angle_waypoint} - \text{Heading_robot} \quad (2)$$

B. NAVIGATION ROUTE CALCULATION

Navigation route distance and the heading are defined graphically as shown in Fig. 3. The two parameters, navigation route distance and the heading are computed from three parameters that are known: the coordinates of the destination, the coordinates of the robot and the heading of the robot relative to magnetic north. The control algorithm reduced the distance error and the heading errors. A linear velocity can help reduce the distance error and on the same lines, the angular velocity can help reduce the angle errors. The distance error “d” can be calculated using the Haversine formula and the azimuth angle of the waypoint by using the “Forward Azimuth Formula”. This can calculate the heading error “α”.

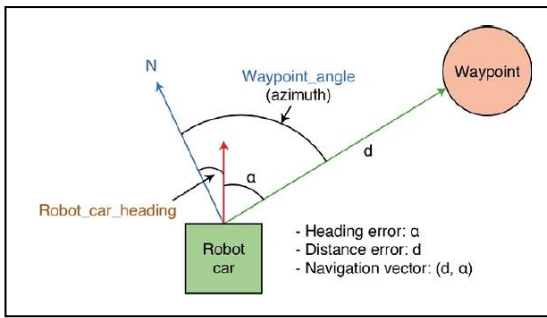


Fig. 3. Navigation route calculation

In spherical analysis of objects, the cartesian formula of measuring distances doesn't hold good and hence the Haversine formula is generally deployed when distances between two points are to be calculated on spherical surfaces such as the Earth [8]. This makes use of the latitudes and longitudes of the two points considered that are on the Earth and the equations for the same are below in equations numbered 3 and 4 respectively [9].

$$\text{Hav}\left(\frac{d}{r}\right) = \text{Hav}(\phi_2 - \phi_1) + \cos(\phi_1) * \cos(\phi_2) * \text{Hav}(\lambda_1 - \lambda_2) \quad (3)$$

$$\text{Hav}(\theta) = \sin^2\left(\frac{\Delta\phi}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (4)$$

- φ1: Latitude 1; λ1: Longitude 1
- φ2: Latitude 2; λ2: Longitude 2
- d: arc(distance); r: Earth's radius
- Δφ = φ2 - φ1; Δλ = λ1 - λ2

C. DESIRED DIRECTION CONTROL

The heading error can be reduced by the solution developed in this paper. The developed solution assists the robot in making turns based on the sign of the heading error. In other terms, when the heading error is greater than zero, the robot turns right and when the heading error is less than zero, the robot turns left. Highly ideal case would be to make the robot move forward when the heading error is only zero. But practically, a tolerance error of ±5 degrees is given. The robot continues to move forward as long as the heading error is in that range. Meanwhile, the robot command itself to move forward by reducing the distance error in the tentative direction towards the destination waypoint. This induces higher angle error in both the directions that is, positive and negative. The tolerance errors are due to various factors such as the GPS, compass, actuators and also environmental conditions. Hence, the robot should iterate over the pre-defined calculations of the newer navigation route and should minimize both the errors until it connects to the destination waypoint

The heading error determines the direction of turns. If the value of the heading error lies between 5 and 10 degrees, the robot automatically turns right and this principle is very similar to that of the stepped proportional controllers as it divides the entire space into three ranges: -5 and +5, -180 and -5, +5 and +180. The developed algorithm makes use of a concept of proportional correction control signal that are "hardcoded" based on the velocities of three corresponding directions of robot movement. Those are: Turn left, right and move forward.

D. IMAGE PROCESSING SYSTEM

Fig. 4 is a blueprint of the image processing system containing a camera that captures images of plants and weeds

and feeds as input to the processor that helps calculate the number of weeds in a particular frame. Weed detection can be done in 4 ways - Location analysis, color analysis, shape analysis and texture analysis [10]. Any unwanted plant grown around a necessary one is considered to be a weed. Weeds are considered to be unwanted where it is not necessary and totally depends on the context. Hence, they possess no botanical classification. [11].

Weeds can be distinguished based on the edge frequencies present in them. Therefore, narrow leaves and wide leaves are the two categories of weeds that are generally considered. [12]. If the count of weeds crosses the threshold value that is preset, a control signal is sent from the processor to the spraying system to spray herbicides on the weeds through the nozzles attached to the legs of the chassis.

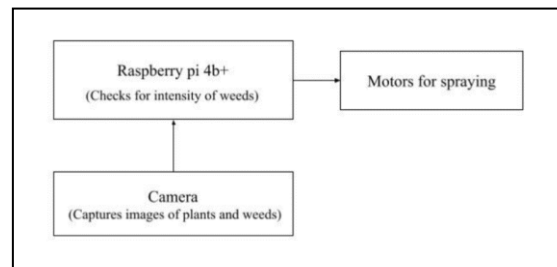


Fig. 4. Image processing block diagram

Image processing is done using the YOLO algorithm. YOLO is an object detection model that is dynamic and compact and provides high performance considering its size and it has been continually improving. Object detection makes use of an object detector model that takes input images and extracts the features of train images subsequently feeding it to the prediction system to detect necessary objects of required classes. There are two classes considered; Healthy plants and weeds. The trained model is capable of detecting healthy plants as well as weeds. The features are bounded by coloured boxes. The YOLO employs an end-to-end differential network to predict the boxes bounding the object and labelling them accordingly. The network of the YOLO algorithm constitutes 3 main components:

- i) Backbone – Image features can be extracted using a CNN or a convolutional neural network at different granularities.
- ii) Neck – A cascade connection of various layers to merge the extracted attributes to deliver it to the prediction system.
- iii) Head – Takes input from the neck and processes class prediction steps.

Furthermore, the YOLOv4 and the version 5 have proven that as a group, the performance of object detection can be improved. The training of model and the number of images captured and used to train the model are very important as they totally affect the end results of prediction.

E. SPRAYING SYSTEM

The spraying system takes the input from the image processing system as a control 1 or 0. Control 1 is sent when the number of weeds in a particular frame crosses the threshold and control 0 when the count is less than the threshold. If the returned value from the image processing system is 1, then the motors connected to the processor will be triggered therefore activating the pump for spraying. Otherwise, the control 0 would deactivate the system and hence does not spray. The spraying can be done using a fogging sprayer [13] which ensures effective area coverage of the herbicides.

Fig. 5 represents the block diagram of the spraying system taking input from the image processing system to initialize spraying of herbicides based on weed count.

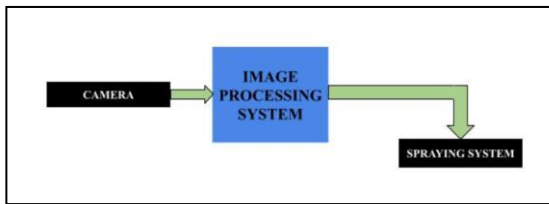


Fig. 5. Spraying system block diagram

III. SIMULATION AND RESULTS

Low accuracy and poor precision of the GPS coordinates retrieved result in abundant noise and affect the navigation performance of the robot. To mitigate the same, a moving average filter that is designed to be a FIR or a finite impulse response filter is more commonly a low pass filter. This filter takes “N readings from the GPS module and calculates the average of all the considered values. If “N” values are considered, Out(0) is the average of all the values from input(0_ to input(N-1) and similarly Out(1) is calculated and so-on. Equation (5) gives the generalised filter response and equation (6) gives an illustration when i=21.

$$y[i] = \frac{1}{N} \sum_{j=0}^{N-1} x[i + j] \tag{5}$$

$$y[21] = \frac{x[21]+x[22]+x[23]+x[24]+x[25]}{5} \tag{6}$$

Fig. 6 depicts the retrieved reading and the filtered reading using the moving average filter on the same plot.

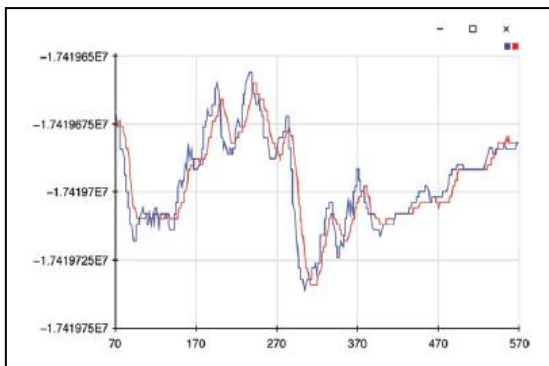


Fig.6 Actual reading vs Filtered reading

There always exists latency in data transmission from the satellites. This causes a drift in the received data from the actual data. This can be eliminated with the help of various filters such as Kalman filter, mean average filter and moving

average filter. Consequently, this paper uses moving average filter that would eliminate errors in autonomous navigation of the robot. Fig. 7 depicts the tabulated data of GPS coordinates retrieved in real time and the same is used for autonomous navigation of the robot.

Latitude	Longitude	Target distance	Target heading	Current heading	Error	Status
12.91711	77.575294	4.76	159.66	193	-33.34	
12.91711	77.575309	4.27	180	197	-17	
12.9171	77.575309	3.89	192.28	200	-7.72	
12.9171	77.575309	3.71	206.46	205	1.46	
12.9171	77.575309	3.29	210.14	209	1.14	
12.91709	77.575309	3.13	211.89	210	1.89	
12.91709	77.575309	2.97	213.82	211	2.82	
12.91709	77.575309	2.59	219.68	214	5.68	
12.91709	77.575309	2.59	219.68	218	1.68	
12.91709	77.575325	2.45	222.52	217	5.52	
12.91708	77.575325	2.18	229.27	223	6.27	
12.91708	77.575325	2.06	233.27	227	6.27	
12.91708	77.575332	2.69	247.17	222	25.17	
12.91708	77.575332	2.62	250.99	221	29.99	
12.91708	77.575332	2.57	255	220	35	
12.91707	77.575325	1.68	260.23	210	50.23	
12.91707	77.575325	1.66	273.29	210	63.29	
12.91707	77.575325	1.66	273.29	210	63.29	
12.91707	77.575325	1.67	276.55	211	65.55	
12.91707	77.575325	1.68	279.77	210	69.77	
12.91707	77.575325	1.75	289.01	210	79.01	
12.91706	77.575325	1.91	299.86	210	89.86	
12.91706	77.575332	1.85	296.45	209	87.45	
12.91706	77.575332	1.79	289.01	262	27.01	
12.91708	77.575325	1.68	242.68	341	-98.32	
12.91709	77.575317	1.5	199.21	314	-114.79	
12.91709	77.575302	1.42	162.12	256	-93.88	
12.91709	77.575302	1.33	162.72	205	-42.28	
12.91709	77.575294	1.25	135.94	210	-74.06	
12.91708	77.575294	1.25	128.79	210	-81.21	
12.91708	77.575294	1.15	122.27	211	-88.73	
12.91708	77.575302	1.12	132.57	210	-77.43	
12.91707	77.575302	0.91				STOP

Fig. 7 GPS data

Fig. 8 and Fig. 9 represent the images captured by the camera for processing and calculating the count of weeds in that image. The two images depict 4 and 3 weeds respectively. The threshold was preset to 3 and anything above 3 would trigger the spraying system for spraying herbicides. The program developed for image processing using the YOLOv5 algorithm will help do that using OpenCV. The python script will return a response signal to the spraying system function that consequently triggers the motor interfaced with the processor board. Based on the tests done, the weeds were effectively detected and spraying was done as needed. The image processing system used a Pi camera to capture the required images and the same was processed using a Raspberry Pi processor

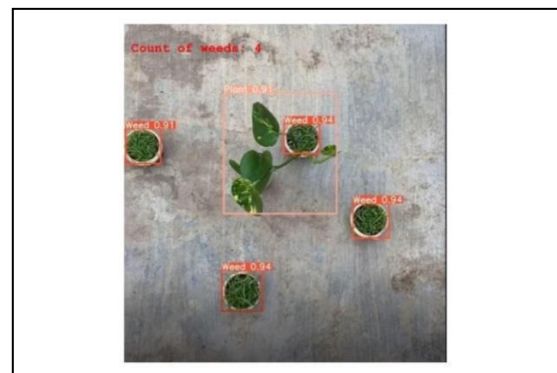


Fig.8. Weed count of 4 detected in image



Fig.9 Weed count of 3 detected in image

An android mobile application was developed to retrieve real time data such as the liquid level from the robot. The IoT system used a nodeMCU microcontroller board with an ESP8266 Wi-Fi module to transmit data over Wi-Fi. Fig. 10 shows the login page of the application that provides access only to the authentic user and the data can be seen once the login is done. Fig. 11 shows the device providing data in milliliters and the farmers can take necessary action by refueling the chemicals whenever needed. This helps farmers keep a track on the expensive chemicals and since the navigation of the robot is autonomous, the welfare of farmers is not compromised.

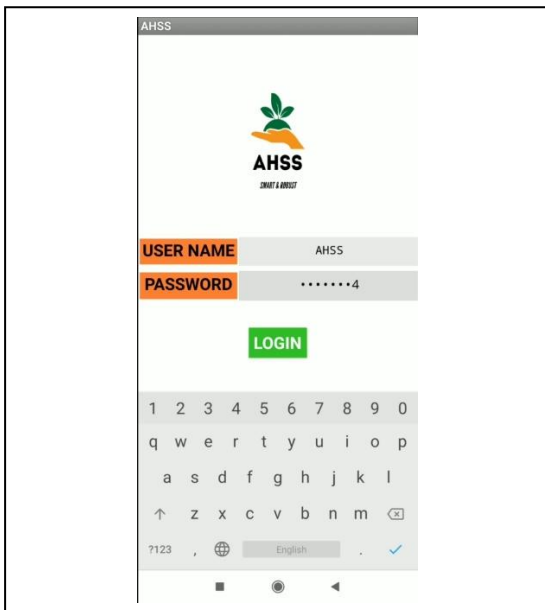


Fig.10. Login page of application

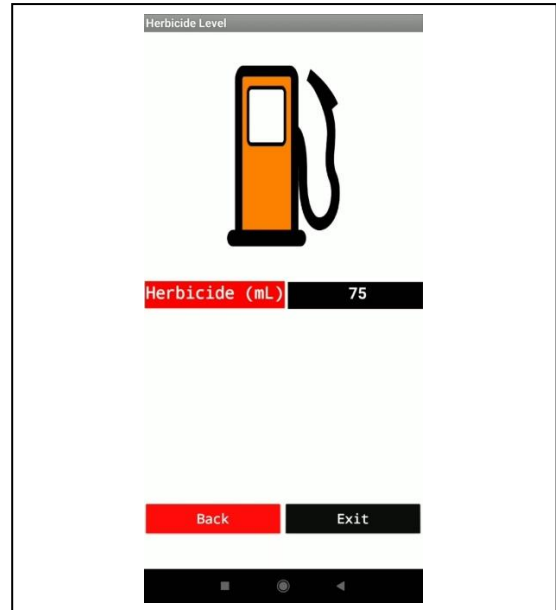


Fig. 11 Real-time data updated in application

IV. CONCLUSION

The traditional methods of spraying herbicides have shown tremendous health effects to the farmers and it also increases the labor costs. As a solution to these problems faced, an Autonomous Herbicide Spraying System has been developed that can navigate itself down the rows of a field thereby increasing the efficiency and ensuring the welfare of farmers. There is always a scope for improvement and many additional technologies can be added to the existing idea to make the system more robust and efficient.

V. REFERENCES

- [1] Machine Learning in Agriculture: A Review Konstantinos G. Liakos 1 , Patrizia Busato 2 , Dimitrios Moshou 1,3, Simon Pearson 4 ID and Dionysis Bochtis 1,*
- [2] Arduino Powered GPS Motor Vehicle Saurav Kumar Verma, C. Vairavel
- [3] Tiny GPS plus library: <http://arduinoiana.org/libraries/tinygpsplus/>
- [4] Jindal, H., Stair Climbing Robot. coordinates, 1, p.2
- [5] Y. Shen et al., "Detection of stored-grain insects using deep learning," Computers and Electronics in Agriculture, vol. 145, pp. 319 – 325, 2018.
- [6] A review on weed detection using ground-based machine vision and image processing techniques Aichen Wanga,c,* , Wen Zhangb , Xinhua Weia,c
- [7] Development of Smart Pesticide Spraying Robot Pvr Chaitanya, Dileep Kotte, A. Srinath, K. B. Kalyan
- [8] Navigation, Guidance & Control Program for GPS based Autonomous Ground Vehicle Amit Yadav1 , Ajeet Gaur1 , and D K Chaturvedi2 Department of Electrical Engineering, DEI, Dayalbagh1 Scientist- D ADRDE- Lab DRDO Agra1
- [9] Haversine formula: https://en.wikipedia.org/wiki/Haversine_formula
- [10] EFFECTIVE CRITERIA FOR WEED IDENTIFICATION IN WHEAT FIELDS USING MACHINE VISION N. Zhang C. Chaisattapagon
- [11] Image Processing in Agriculture Mrs. Latha1, A Poojith2 , B V Amarnath Reddy3, G Vittal Kumar4
- [12] WEED DETECTION USING IMAGE PROCESSING Ajinkya Paikekar1, Vrushali Ghule2, Rani Meshram3, V.B. Raskar4
- [13] Austerweil M., Grinstein A., (1997), "Automatic Pesticide Application in Greenhouses", Phytoparasitica, Vol 25, pp. 37-42.