IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Rugged Autonomous Unmanned Ground Vehicle for Precision Agriculture Working Based on Color Navigation Algorithm using Raspberry Pi2

G. Pradeep Kumar[1], S. Shobana[2], P. Uma sankari[3], D. Devamalini[4], J. Dhanalakshmi[5]

Assistant Professor, ECE Department, Velammal College of Engg & Technology, Madurai[1]

Final Year ECE, Velammal College of Engg & Technology, Madurai[2, 3, 4, 5]

Abstract: A Rugged unmanned ground vehicle (UGV) is a robot that operates while in contact with the ground and without an onboard human presence. The autonomous vehicle can be used for several applications where it may be inconvenient, dangerous, or impossible to have a human operator present particularly in mine monitoring, mineral exploration, high radiation places. Generally, the vehicle will have a installed sensors to observe the environment, and will either autonomously make decisions about its behavior or transfer the information to a end user operator at a different location who will control the vehicle through teleportation. To navigate the ground vehicle, lot of researchers are working with RF controlled and tethered controlled system. Few are working in the autonomous vehicle by programming the path. All those control system can be perfectly used for flat and normal surface. Since the programmed path is based on timing that can be used for flat and normal surface. While considering the rugged path particularly agriculture filed or farm, these systems might not work. Because of rugged path the vehicle can get delay while moving, then the entire path navigation will get changed. This problem can be overcome in our proposed work having color based navigation system can be opted for any rugged path like agricultural field and so on.

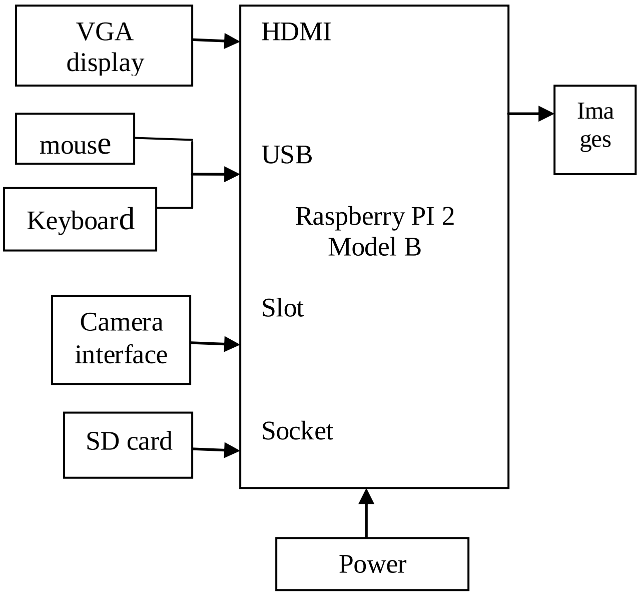
Keywords: UGV, OS, AGV, RF.

I. INTRODUCTION

The objective of this project is to implement automation of unmanned ground vehicle with the help of raspberry pi 2 B model. The reason behind the project is to surveillance the large area where human beings could not complete the work efficiently. Especially in Agriculture having larger size of farm is difficult to monitor and maintain. For example seeding, weeding, spot spraying, identifying the infected plants and so on. Not only in agriculture can this also used in Industries for monitoring the areas. The vehicle can be designed according to the applications. Our aim is to control the vehicle's motion in the rugged path of farm by tracking the color which is displayed for direction in the farm land using Raspberry pi 2 and the driver module. This new proposed method can be implemented in any kind of rugged farm.

II. HARDWARE AND SOFTWARE:

The raspberry pi 2 is core kit which replaces the personal computer and does the process very efficiently. The method includes vital hardware components and software.



Pic 1: Block diagram of Raspberry pi

Raspberry pi 2:
It is a small sized single board computer. There are various models available in Raspberry such as Model A, Model B, Model A+, Model B+, Model 2 b, Model 3. It has 1GB RAM model with 4 USB ports, an Ethernet connection, Speed is 900Mhz, the requirement of power supply for this device is 3V, an slot for SD card, it also consists of 40 pins and HDMI port for projection.

Raspbian OS - Noobs:
Among all operating system such as Risc OS, Chromium OS, Angstorm Linux, Pidora, Gentoo, Raspbian is the best one for Raspberry. It is feasible for working and also free operating system available in the website. Raspbian also has types as wheezy, Jessie, noobs. We prefer noobs because it improves the performance and flexibility, particularly as regards the control of the system processes,

DOI 10.17148/IJARCCE.2016.55145

fast, pre-programmed, high capacity and as with any update, there are also numerous bug fixes and tweaks. At the same time as upgrade to noobs we can able to add bunch of changes and improvements to desktop user interface.

Pi-Camera:
There is a slot to attach camera in Raspberry which enables us to take high definition pictures and videos. It has resolution as 5 megapixels and video mode rates as 1080p.

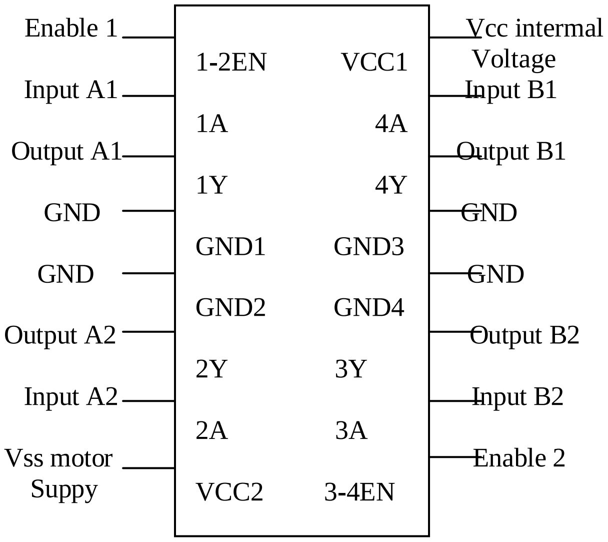
Python:
It is high level programming language which allows coders to express their ideas in fewer lines when compared other programming languages like C,C++,Java. It also had a large number of packages and also used to combine software with hardware. Python can be executed directly by the system without compiling it. So the execution time is very less when compare with other programming languages.

Open CV:
Open Source Computer Vision Library focus at real-time computer vision. The library source contains lot of modules which can be used for image processing like section, tracking, enhancement of image and so on. And the Open CV also supports 3D model extract, augmented reality and etc. This machine language is an efficient programming which does not require the compiling process. So the machine gets the code directly and executes which improves the system performance much better. To run on the raspbian OS each code require lot of modules according to the nature of code. Example to run the image processing python code, the kit requires NUMPY, CV2 modules basically.

Driver Module:
Driver module is used for the external power supply for motor support. It consists of 16 pins, phoenix connectors which is used for motor connection, PWM (Pulse Width Modulation) selection switch. The module used here is L293D, it is quad push-pull drivers which has the ability to deliver output current up to 1A for each channels. TTL compatible logic is used to control the channels and so each pair of drivers (a full bridge) is enhanced with an constrain input available at pin 1 and pin 9. The motor can run only when chip curb is at high logic i.e. chip inhibit is enabled. The power supply is 12-40V.

Unmanned Ground Vehicle:
The vehicle can be operated autonomously i.e(able to work without human). It is widely used where inconvenient, hazardous or difficult to handle the situation by human operators. It has the ability to track the path, gather user about the environment, period of working can be extended without the help of operators. They can also repair themselves without outside assistance. They can be widely used in Defense applications, Agriculture, Industries, etc,. It can able to monitor and self-restrained machine.

The pin configuration of driver module is shown below



Pic 2 : Pin diagram of L239D module

III. EXISTING METHODS

There are so many researchers working in the emerging field and proposing lot of ideas for navigating the unmanned vehicle particularly for agriculture, aquaculture in prawn cultivation, surveillance and security systems. The following methods have its own issues which cannot be applied for all kind of fields.

i. Autonomous vehicle based on line following:
The line follower vehicle is based on simple technique it works but the issue is, the line follower vehicle is good only in the indoor environment. Particularly in the outdoor it fails to track the line since the sun rays' reflection during day time. This causes wrong prediction of line in the outdoor environment.

ii. Programmed path for vehicle direction:
The path of the vehicle is programmed in this proposal. This method can be applied for industrial area which has flat surface as path. Since the programming depends the timing if any rugged path delays the motion the actual destination cannot be reached as per the program. So this proposed work cannot be applied for rugged path like farm and some other area.

iii. IR & Ultrasonic sensor based navigation:
The ground vehicle will have the IR sensor or the Ultrasonic sensor for finding the obstacle during the motion. If there are any obstacles the front motor will take an angle shift so the vehicle gets diversion from the obstacle. But while moving in a up and down unstable path the sensor keeps detecting as obstacle and stops the motion. This proposed work also cannot be applied for outdoor farms.

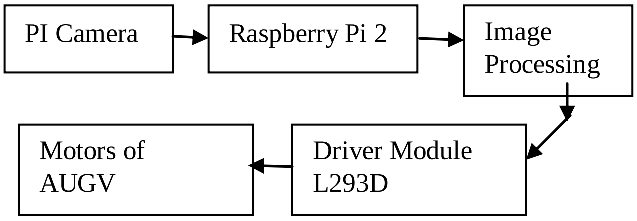
iv.RF controlled unmanned ground vehicle:
This proposed work, include the human as interface for controlling the vehicle and not an autonomous vehicle.

DOI 10.17148/IJARCCE.2016.55145

The vehicle will have a camera mounted in front of the head. Then the human can make error while controlling the vehicle it causes damage to the vehicle.

IV.PROPOSED METHOD

In the proposed method we use raspberry pi 2 for navigating the vehicle. The camera capture the colors from the fixed position, based on the color captured image the instruction is given to the ground vehicle and changes its direction according to the program as already written in the kit with the help of python language.
The Raspbian OS installed in the Raspberry pi 2 kit and the necessary library functions are installed using python libraries. There are three colors are used and for each one we use instructions to control the vehicles path. Our project consists of three vital phases.



Pic 3: Block diagram of project layout

Phase 1: Design of unmanned ground vehicle:
The design part is the vital part of the proposed work since the vehicle is going to be used in the agricultural field where the path will be uneven, muddy and unpredictable. So the vehicle wheels are specially made up with grip and rugged.

Pic 4: Rugged Unmanned ground vehicle

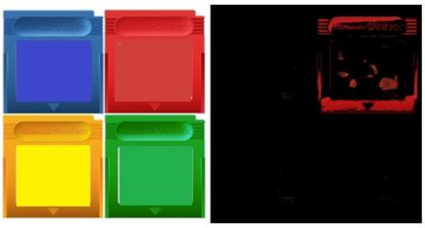
The springs attached with body makes the free movements in any terrain which can auto adjust and maintain its stability. And it can climb at any rugged terrain.
While designing the vehicle the bellowed equation should be considered for finding the required torque for vehicle.

$$F_t = F_r + F_g + F_d + F_{ie}$$

Here the total torque is calculated by adding
Fr- Tyre rolling resistance,
Fg- Forces due to gradient,
Fd- Aerodynamics drag,
Fie- Equivalent inertial force.

Phase 2: Algorithm development:
The algorithm is developed in the python language since the execution time is less done by the raspberry pi 2 kit. The aim of the algorithm is to detecting the colors which is posted in the farm for navigation support, and according to the color the vehicle has to take decision that whether it should move forward or taking right or left direction, so the Picamera will at the in front of vehicle for detecting the color and navigating autonomously.

Pic 5: Raspberry pi 2 kit with Pi camera

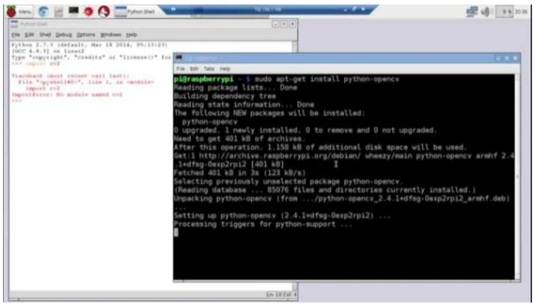


Pic 6: Configuration of python and OpenCV

Before going for the algorithm development configuration of python package and OpenCV installation is the primary work should be done by the following code.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
$ sudo apt-get install python2.7-dev
$ wget -O opencv-2.4.10.zip
$ unzip opencv-2.4.10.zip
$ cd opencv-2.4.10
```

Once installed all the library files and modules the algorithm can be developed. Initially from the loaded image the image



```
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for
>>> import cv2

Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    import cv2
ImportError: No module named cv2
>>>
```

```
pi@raspberrypi ~ $ sudo apt-get install python-opencv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  python-opencv
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 401 kB of archives.
After this operation, 1.158 kB of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian/ wheezy/main python-opencv armhf 2.4
.1+dfsg-0exp2rpi2 [401 kB]
Fetched 401 kB in 3s (123 kB/s)
Selecting previously unselected package python-opencv.
(Reading database ... 85076 files and directories currently installed.)
Unpacking python-opencv (from .../python-opencv_2.4.1+dfsg-0exp2rpi2_armhf.deb)
...
Setting up python-opencv (2.4.1+dfsg-0exp2rpi2) ...
Processing triggers for python-support ...
```

Pic 8: Red color extraction using python language

feature extraction algorithm is developed to trace the certain color. So the boundary of the red, yellow, blue and grey colors' pixel value are been set in the python coding.

```
([17, 15, 100], [50, 56, 200]),
([86, 31, and 4], [220, 88, 50]),
```

DOI 10.17148/IJARCCE.2016.55145

```
        ([25, 146, 190], [62, 174, 250]),
        ([103, 86, 65], [145, 133, 128])
```

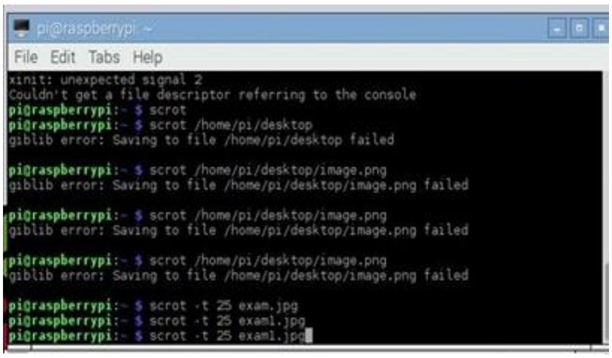
Then the loaded image pixel value is calculated by the coding

```
lower = np.array(lower, dtype = "uint8")
        upper = np.array(upper, dtype = "uint8")
```

Once it is calculated the certain pixel will be extracted and will be displayed as a subtracted image file.

```
mask = cv2.inRange(image, lower, upper)
   output = cv2.bitwise_and(image, image, mask = mask)
```

Similarly all the colors can be obtained by the processor by the predefined boundaries. The coding considered the main colors which can be observed by the PI camera easily and can be processed. This image processing work will be happen on the board while moving towards the path.

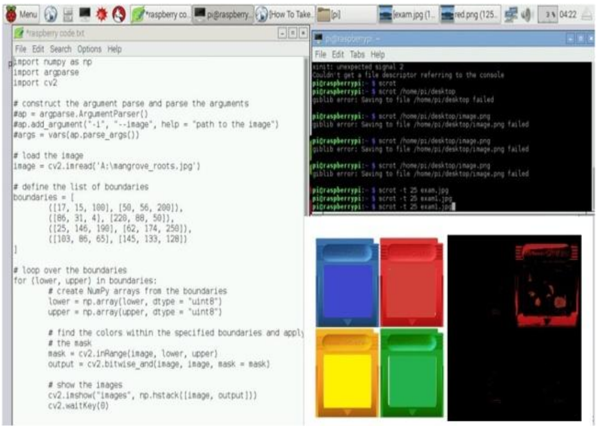


```
xinit: unexpected signal 2
Couldn't get a file descriptor referring to the console
pi@raspberrypi:~ $ scrot
pi@raspberrypi:~ $ scrot /home/pi/desktop
giblib error: Saving to file /home/pi/desktop failed

pi@raspberrypi:~ $ scrot /home/pi/desktop/image.png
giblib error: Saving to file /home/pi/desktop/image.png failed

pi@raspberrypi:~ $ scrot /home/pi/desktop/image.png
giblib error: Saving to file /home/pi/desktop/image.png failed

pi@raspberrypi:~ $ scrot /home/pi/desktop/image.png
giblib error: Saving to file /home/pi/desktop/image.png failed

pi@raspberrypi:~ $ scrot -t 25 exam.jpg
pi@raspberrypi:~ $ scrot -t 25 exam1.jpg
pi@raspberrypi:~ $ scrot -t 25 exam1.jpg
```

Pic 9: Python code Execution in through virtual terminal

The vehicle motion is controlled by the colors which are posted in the agricultural field and the pi camera capture the image and feeds to the processor. The processor analyzes the image and detects the color and takes the decision finally.

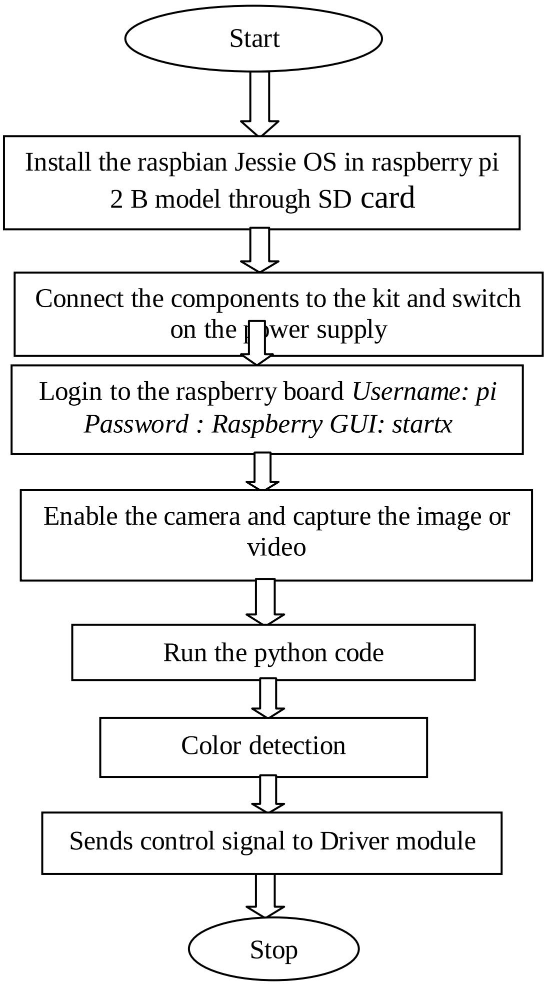
The reason going for image extraction is that, in real time while capturing the image in the field, apart from the posted color for the direction, remaining color of sand, sky, green plants can interrupt the process of taking decision. So from that by extracting the wanted color like Red, yellow and dark blue the processor can take decision confidently.



```
import numpy as np
import argparse
import cv2

# construct the argument parse and parse the arguments
#ap = argparse.ArgumentParser()
#ap.add_argument("-i", "--image", help = "path to the image")
#args = vars(ap.parse_args())

# load the image
image = cv2.imread('A:\mangrove_roots.jpg')

# define the list of boundaries
boundaries = [
        ([17, 15, 100], [50, 56, 200]),
        ([86, 31, 4], [220, 88, 50]),
        ([25, 146, 190], [62, 174, 250]),
        ([103, 86, 65], [145, 133, 128])
]

# loop over the boundaries
for (lower, upper) in boundaries:
        # create NumPy arrays from the boundaries
        lower = np.array(lower, dtype = "uint8")
        upper = np.array(upper, dtype = "uint8")

        # find the colors within the specified boundaries and apply
        # the mask
        mask = cv2.inRange(image, lower, upper)
        output = cv2.bitwise_and(image, image, mask = mask)

        # show the images
        cv2.imshow("images", np.hstack([image, output]))
        cv2.waitKey(0)
```

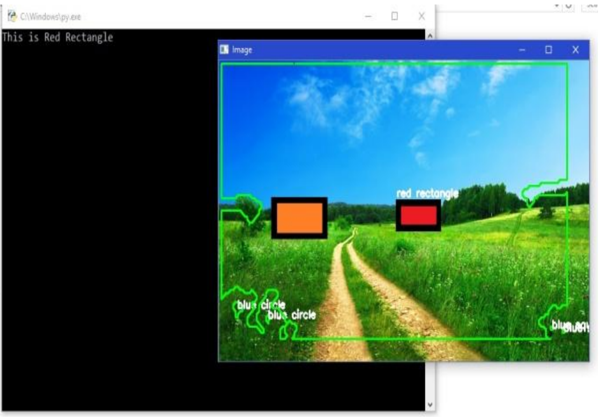
```
xinit: unexpected signal 2
Couldn't get a file descriptor referring to the console
pi@raspberrypi:~ $ scrot
pi@raspberrypi:~ $ scrot /home/pi/desktop
giblib error: Saving to file /home/pi/desktop failed

pi@raspberrypi:~ $ scrot /home/pi/desktop/image.png
giblib error: Saving to file /home/pi/desktop/image.png failed

pi@raspberrypi:~ $ scrot /home/pi/desktop/image.png
giblib error: Saving to file /home/pi/desktop/image.png failed

pi@raspberrypi:~ $ scrot /home/pi/desktop/image.png
giblib error: Saving to file /home/pi/desktop/image.png failed

pi@raspberrypi:~ $ scrot -t 25 exam.jpg
pi@raspberrypi:~ $ scrot -t 25 exam1.jpg
pi@raspberrypi:~ $ scrot -t 25 exam1.jpg
```

Pic 10: Executing in the Raspbian environment


Start
Install the raspbian Jessie OS in raspberry pi 2 B model through SD card
Connect the components to the kit and switch on the power supply
Login to the raspberry board *Username: pi Password : Raspberry GUI: startx*
Enable the camera and capture the image or video
Run the python code
Color detection
Sends control signal to Driver module
Stop

Pic 11: Flowchart of algorithm & Output

For each color the program was predefined and it will sent the control signal to the driver module. Literally by enabling the certain pin of driver module according to the color, the corresponding motor will function for certain period till it finds the path without obstacle. The ultrasonic sensor can be attached for detecting the obstacle in the pathway while executing in future.

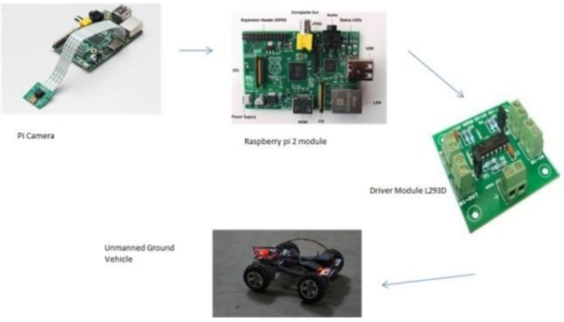


```
This is Red Rectangle
```

Pic 12 : Color detection in a farm field

The overall pictorial project flow diagram shown in the pic 13 followed by the Pi camera.

DOI 10.17148/IJARCCE.2016.55145



Pic 13: Project flow Diagram

Phase 3: Hardware interface with vehicle
Once the algorithm part is done the control signal will be generated and fed it to the driver module which is going to control the motors of vehicle. The control signal generation is nothing but, enabling the certain pin in the raspberry kit particular pin will be enabled in the driver module since it is connected with the patches.

Pic 13 : Raspberry pi 2 and LM293D

Table 1: Color along with vehicle direction

COLOR OF THE IMAGE	DIRECTION OF VEHICLE
RED	RIGHT
BLUE	LEFT
BLACK	STOP

generation is nothing but, enabling the certain pin in the raspberry kit particular pin will be enabled in the driver module since it is connected with the patches. So 5V will be generated in the driver module output pin. By connecting relay in the pin we can increase the voltage till 12 V where the motors will function effectively to check in the real time environment. The following colors have been fixed for certain decisions. Here the decision is nothing but running the certain motors. For tilting right side a DC motor will be connected in the front wheel of vehicle. Similarly for the left side another color will be chosen or fixed. This proposed work has greater advantage of can be worked efficiently in any rugged path and can be used in various fields like agriculture, mine monitoring, mineral exploration, bomb detector, defense field and so on.

Pic 13: Field test of UGV with Raspberry kit

V. CONCLUSION

The aspiration of our work completed with the built of self driving rugged unmanned ground vehicle working based on color detection of the image using raspberry pi single-board computer and controlling the vehicle navigation as per the coding in an agricultural field. This paper consists of two sections as hardware and software. The hardware part includes the design of vehicle, configuring kit, interfacing and field test. Then the software part includes coding in python language and running on Raspbian OS with the support of open CV library package. The python language makes it easier to detect the color using image enhancement techniques. The algorithm that we have developed on the raspberry has been successfully implemented and it worked under all the conditions.

VI. FUTURE SCOPE

The work could be enhanced by improving the color detection technique along with ultrasonic sensors for obstacle avoidance in future. Then the GPS can also be installed to locate the path for the vehicle. It could be further enhanced like quick response to obstacle, vehicle climbing over rugged path. Our work proposed for purely to enhance the agricultural work. The proposed work along with sensor integration can be implemented in various fields like Industries, Power plant monitoring using WSN, Mine monitoring, Bomb detection, Eaves dropping, Security surveillance and so on.

REFERENCES

[1] https://www.raspberrypi.org/products/raspberry-pi-2-model-b
[2] https://www.element14.com/community/community/raspberry-pi/raspberrypi2
[3] http://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie
[4] http://www.instructables.com/id/RasPi-OpenCV-Face-Tracking
[5] http://visualgdb.com/tutorials/raspberry/opencv/camera
[6] http://www.raspberrypi.org/
[7] http://en.wikipedia.org/wiki/Raspberry_Pi.
[8] http://www.python.org/
[9] http://www.codeproject.com/Articles/850842/Setting- up- and-Programming-Raspberry-Pi-in-Python
[10] Gurjashan Singh Pannu , Thapar University, Patiala International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 9, March 2015
[11] Narayan Pandharinath Pawar & Minakshee M. Patil, Driver Assistance System based on Raspberry Pi, International Journal of Computer Applications (0975 – 8887) Volume 95– No.16, June 2014, pp. 36-39.
[12] K.S.Shilpashree M.Tech, VLSI Design and embedded system, E & C Department, Kalpataru Institute of Technology, Tiptur, India. International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015.
[13] Sahani, M., Rout, S.K., Sharan, A.K., Dutta, S. real time color image enhancement with a high regard for restoration of skin color by using raspberry pi Communications and Signal Processing (ICCSP), 2014 International Conference IEEE.
[14] http://www.raspberrypi-spy.co.uk/2012/05/install-rpi- gpi-python-library
[15] http://elinux.org/RPi_Easy_SD_Card_Setup
[16] https://solarianprogrammer.com/2015/05/08/detect-red-circles-image-using-opencv

DOI 10.17148/IJARCCE.2016.55145