International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 04, April-2015

Image Processing Based Rose Harvesting System using Raspberry Pi

Charu Bhartiya
M.Tech,
Embedded System
VIT University
Vellore, Tamil Nadu, India

Prof. Ashish P.
Assistant Professor,
School of Electronics Engineering
VIT University
Vellore, Tamil Nadu, India

***Abstract*—This project investigates the accuracy and effectiveness of harvesting the rose flower using OpenCV through C++ and Raspberry Pi. In order to avoid the difficulties of handling the flowers that arise during rose harvest and also to reduce the monetary cost of the entire system, a rose flower harvesting system using Raspberry Pi is introduced. It is used to create an environment which will automatically detect and identify the rose flower in the flower garden using the image processing algorithms running on OpenCV. We use the Raspberry pi module which is an Embedded SoC architecture (with ARM + Video core GPU) based Single Board Computer. The ARM architecture is for holding the entire operating system which is a variant of Linux made especially for ARM processor and a Video Core GPU to work with OpenCV, a library for Video & Image Processing in real time. This library also connects the device drivers for camera and GPIO for triggering. The open computer vision (OpenCV) is a machine vision technology that will give the feasibility to process images and videos using real time cameras. A glowing LED and rotating servo motor connected to the Raspberry Pi indicate the detection and identification of a rose flower based on specified parameters.**

Keywords*—Raspberry Pi, Rose Detection, Image Processing, OpenCV, LBP Classifier.

I. INTRODUCTION

Rose harvesting may be simply carried out using manual labourers. But when the farm land extends to tens of acres, huge man power is required. Significant amount of time, money and effort is needed. Since it involves human intervention, the process of rose harvesting becomes subjective and uncertain as to how much time will be required to complete the job. The efficiency and perception levels of different people may be different. Some may harvest the properly bloomed roses and some may not. Natural interruptions like heavy rain, fog and mist, etc. may also occur. Due to human limitations, limited working hours, interruptions during the work hours it is better to employ robots to do the task since they can do the work repeatedly without getting tired for longer hours. Also the perception of whether a rose has blossomed or not can be programmed correctly through computer programs and the robot can detect the correct colour, size and texture of the rose and identified the fully grown roses for harvesting.

These robotic processes are easily built and are controlled through single board computer, say; Raspberry Pi that captures the live video through a standard USB webcam and will identify the roses in the field. After identifying the rose through the image processing algorithms which are programmed, it will circle the rose which means the rose has been detected and an LED will glow and servo motors start rotating.

The OpenCV is installed in the Raspberry Pi which is one of the primary tools of Image Processing. We have created an XML file using cascade trainer in MATLAB. This XML file is used for detecting and identifying the rose flower. The cascade classifier can recognize many kinds of rigid objects, once the appropriate classifier is trained. Here, we are using LBP (local binary pattern) featured Cascade Classifier for Rose Flower Detection.

When the webcam captures the live video, the frame is parsed at the rate of 0.5/sec. Each frame is compared with this XML file which we have created. If it matches then Raspberry Pi will circle the rose and the rose has been detected and identified.

II. METHODOLOGY

This Rose Harvesting System is fully automatic system using raspberry Pi which reduces human work and offers speed of operation.

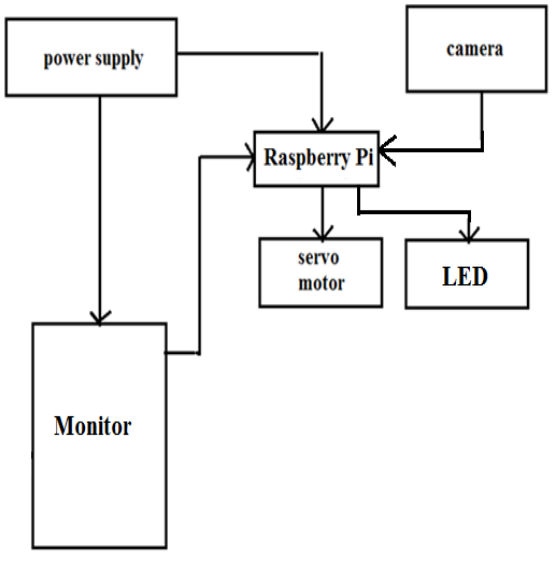

Fig.1. Block Diagram of Rose Harvesting System

Fig.1. shows the block diagram of Rose Harvesting System. When the power supply is ON, Raspberry Pi becomes active. Then the terminal on the PC monitor has to be opened and we have to run the OpenCV code. A window will be opened automatically and shows the scenery captured by the camera. We may place any object or rose flowers of other colours also. The system will identify only the rose flower that too of red colour. Based on the LBP classifier algorithm, the image processing will be carried out and once the count value becomes 20 it will glow the LED and start the servo. The servo will stop after five rotations. The servo will move in clockwise and anti-clockwise direction with predefined angles mentioned in the code. Thus the system successfully detects and identifies a red rose flower. Fig. 2 shows the flow chart of Rose Harvesting System and working principle of the system.

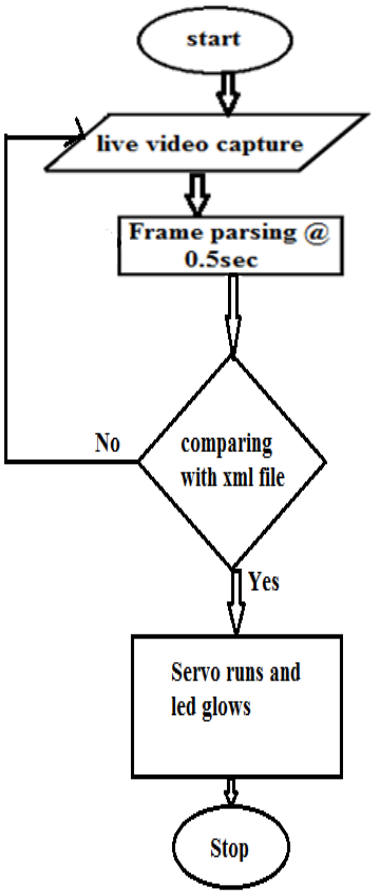

Fig.2. Flow chart of the system.

A. XML file

XML stands for Extensible Markup Language. This language is used to add additional information. This information cannot be seen by the end-user, but these are very important to machine to read and to process the text accurately and correctly.

Xml is a very similar language to HTML, but it is more flexible compared to HTML. XML file is the universal format for all structured data and documents present on the web. Xml files are very easy to understand and can be created easily. It is self-describing language.

B. Creating xml file

Here MATLAB tool called cascade trainer is used to create .xml file. For creating an xml file, we need to collect thousands of positive and negative images to train the cascade classifier.

- Positive images are those images which consist of object and here object is red rose.
- Negative images are that image which doesn't consists of object, only background is present in these images.

When we collect these images, now next work is to create xml file by using cascade trainer. In all the images, we have to define the appropriate ROI (Region of Interest). After defining ROI, we just have to train it so that it creates an xml file which we will compare with the frames which is coming from the live video.

C. Need of xml file

- XML file is used to identify and describe the information correctly, accurately and precisely, in such a way that machine can be programmed to understand all the information.
- XML file also allows the set of same type of data to be created and to be handled consistently and with no errors.
- The information which is stored in xml file can be changed programmatically under the control of the machine.
- They can be easily converted to any other format without any loss of data and information.

D. Cascade Trainer

It is used for managing the selection and positioning of rectangular ROIs (Region of Interests) in the list of images. It is also used for creating the new cascade classifiers. Cascade classifier is used to load an .xml file. LBP (linear binary pattern) classifier is used.

- It is a type of feature which is used for classification in the computer vision.
- Local Binary Pattern (LBP) feature is used very well in many applications like segmentation, texture classification, surface inspection and image retrieval.
- The LBP operator works by labelling the image pixels by thresholding the 3-by-3 neighbourhood of each pixel with the centre pixel value.
- The result is considered as the binary number.

Figure 3 shows an example of the LBP calculation.

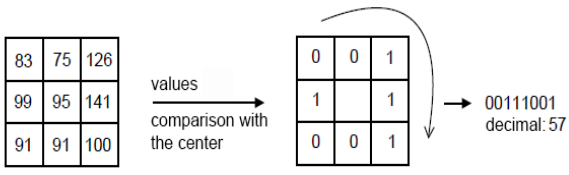

Fig.3. Example of LPB classifier

III. ANALYSIS

Since the "Raspberry pi - 1 Model - A" has less RAM of 256MB and because of the Single core CPU, the expected speed of the CPU is less achieved in the Model-A. There is a little delay in the period of detection and processing time nearly around 150 milliseconds in model-A.

Also the Cascade training is done with 500 positive and 500 negative images as inputs, so the accuracy of the Rose Flower identification will be 70% efficient approximately. Nearly more than 4000 positive and negative images are required to train the Cascade Classifier to meet any rose flower identification irrespective of Colour and petal shapes.

With direct DC motors rotation based positioning is little difficult, so we are using servo motors for showing the result. In servo motors we can adjust the angle of rotation and it is more useful in robotic arm handling compared to dc motors.

The Single Camera video capturing system is limited to use in auto-localization.

IV. RESULT

The Rose flower has been successfully identified. After detecting rose, it will circle the rose and LED glows and servos start rotating.

Other objects were not detected even though they are red in colour. As shown in the figure, the red coloured object is not detected.

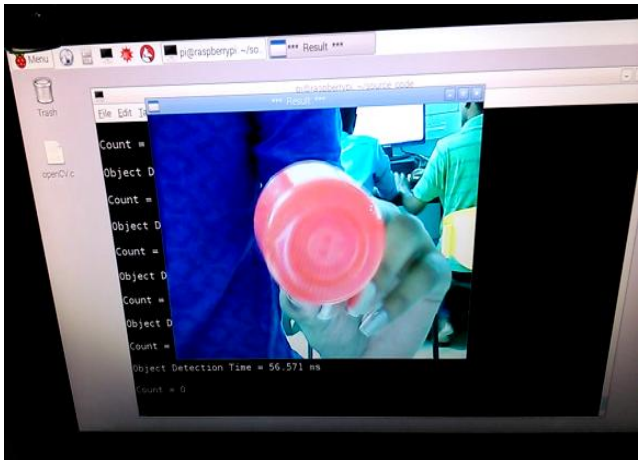

Fig.4. Other object is not detected.

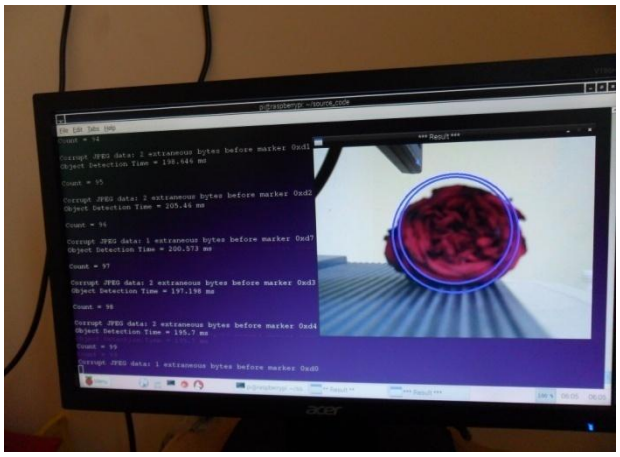

Fig. 5. Detection of rose

The raspberry Pi is connected to the monitor for showing the result. The camera is connected to the raspberry Pi. HDMI to VGA convertor cable is connected to the raspberry Pi so as to get the video displayed on the monitor of the computer.

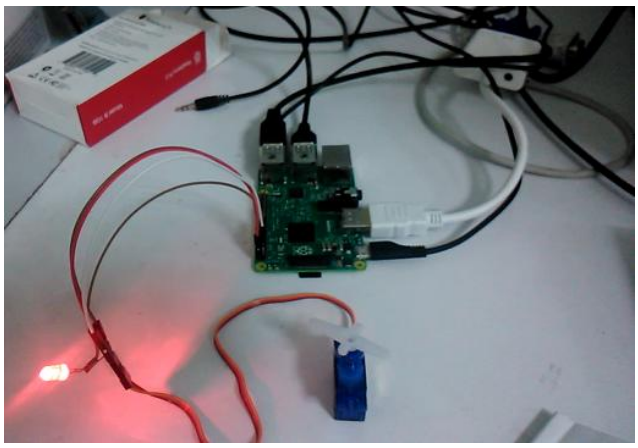

Fig. 5 Led glows and servo run

Fig.4 and Fig.5 shows the result of the Rose Harvesting System. When the rose is detected and when the count becomes 20, led glows and servo runs in clockwise and anti-clockwise direction with some angle less than 180 degrees mentioned in the code.

V. CONCLUSION

Thus an effective system for Rose Flower harvesting prototype has been done with a Single board computer say Raspberry Pi, which is of SoC based ARM architecture with GPU system (Graphics Processing Unit).

So instead of using bigger kind of giant robotic system, it is also possible with miniature type Embedded based Robotic Systems which is governed by Linux Operating Systems. By reducing the Kernel size and avoiding unwanted libraries or Package installations, a high Speed System can be developed easily with time and money.

Servo runs with different angle given in code, we can change the setting of angle of rotation of servos. LED is connected to the Raspberry Pi to show the output that the rose has been detected and identified.

VI. FUTURE ENHANCEMENTS

The first enhancement can be on the Single board Computer Version, from Raspberry Pi - 1 Model - A to Raspberry Pi – 2 Model – B+. Raspberry Pi -2 has a Clock Speed of 900 MHz (Turbo approx. 2 GHz), DDR2-RAM of 1 GB (4 times more), SD card with 32 GB (4 times more), Dual Core Video Core IV based GPU Multimedia Co-processor with supports on 3D, 1080p30, OpenGL, Hardware accelerated OpenVG, H.264 high-profile decoder and mainly QUAD CORE processor with Cortex series, ARM-Cortex A7 unlike the Raspberry Pi -1 model which has a Clock Speed of 700 MHz (Turbo approx. 1GHz), 256 MB RAM, SD card's Primary Partition of 8 GB, 2D - Video Core IV based GPU and a Single Core of ARM-11 hardware and ARM-6 Architecture. Also a speed enhancement of up to 6 times is possible in the processing.

Cascade training should be made with 5,000 to 10,000 Snap shots of positive and negative images of Rose Flowers which should be trained on HAAR algorithm based Cascade Classifier training.

The robotic arms with position Sensors based methods for easier and accurate positioning, and then the robotic arm's holding pressure should be increased to 1.5 to 2 Kilo Pascals, which cuts the Stem smooth.

Two or more Cameras can be arranged for video capturing and localization systems.

A perfect rover can be made which rolls on the mud or any kind of rough surface in the garden point of view.

ACKNOWLEDGMENT

I express the deepest gratitude to our chancellor, Dr. G. Vishwanathan for providing excellent infrastructure and lab facilities. I express my sincere thanks to Dr. G. Ramachandra Reddy, Dean, SENSE for giving me an opportunity to carry out this project in VIT University. I am very grateful to thank our Program Manager, Prof. G. Mary, SENSE for her kind support during the entire course of this project. I am very thankful to my guide, Prof. Ashish P., Assistant Professor with his exemplary guidance, immense help, motivation and guidance for successfully finishing the project. Finally, I take this opportunity to extend my deep appreciation to my family members and friends, for all that they meant to us during the crucial times of the project. And most importantly, thanks to God for always being my motivation and guiding light towards the right path to achieve success.

REFERENCES

[1] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek, *A brief introduction to OpenCV*, Opatija, Croatia MIPRO 2012, May 21-25.
[2] Xianghua Fan, Fuyou Zhang, Haixia Wang, Xiao Lu, *The System of Face Detection Based on OpenCV*, IEEE-2012, 24th CCDC.
[3] Cahit Gürel, Prof. Dr. Abdulkadir Erden, *Conceptual Design of a Rose Harvesting Robot for Greenhouses*, The 20th Int. Conf. on Mechatronics and Machine Vision in Practice- M2ViP 2013, September 18-20, 2013, Ankara, Turkey
[4] J.C., J. Hemming, C. Van Heerde, F. Goldbach, R. van Soest and E. Wekking, *Automated rose cutting in greenhouses with 3d vision and robotics: analysis of 3d vision techniques for stem detection*, Noordam, , 2005, Acta. Horticulturae, 691: 885-892.
[5] Forsyth, A.F. and J. Ponce, *Computer Vision A Modern Approach*, Prentice Hall of India, 2006.
[6] R N Daschoudhary, Rajashree Tripathy, *Real Time Face Detection And Tracking Using Haar Classifier On* SOC, Proceedings of SARC-IRF International Conference, 12th April-2014, New Delhi, India, ISBN: 978-93-84209-03-2
[7] Petre Anghelescu, Ionut Serbanescu, Silviu Ionita *Surveillance System using IP Camera and Face-Detection Algorithm*, IEEE, 2013.
[8] D.Jeevanand, K.Keerthivasan,J.MohamedRilwan,P.Murugan *Real Time Embedded Network Video Capture And SMS Alerting system* International Journal of Communication and Computer Technologies,2014.
[9] Sneha Singh, PradnyaAnap, YogeshBhaigade, Prof.J.P.Chavan, *IP Camera Video Surveillance using Raspberry Pi*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015.
[10] Jo Chang-yeon, *Face Detection using LBP features*, CS 229 Final Project Report, December 12, 2008
[11] http://opencv.org/
[12] http://docs.opencv.org/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
[13] http://docs.opencv.org/doc/user_guide/ug_traincascade.html.