

# Real Time Object Counting using Raspberry pi

J.Manasa<sup>1</sup>, J.T.Pramod<sup>2</sup>, Dr.S.A.K.Jilani<sup>3</sup>, Mr.S.Javeed Hussain<sup>4</sup>

M. Tech student, Dept of Electronics and Communication Engg, Madanapalle Institute of Technology & Science,  
Madanapalle, A.P, India<sup>1</sup>

Assistant professor, Dept of Electronics and Communication Engg, Madanapalle Institute of Technology & Science,  
Madanapalle, A.P, India<sup>2</sup>

Professor, Dept of Electronics and Communication Engg, Madanapalle Institute of Technology & Science,  
Madanapalle, A.P, India<sup>3</sup>

Associate Professor, Dept of Electronics and Communication Engg, Madanapalle Institute of Technology & Science,  
Madanapalle, A.P, India<sup>4</sup>

**Abstract:** This paper proposes the design and implementation of object counting algorithm based on image processing by using Raspberry pi on real time basis. Features of an image containing objects to be detected are extracted uses a method called BLOB (Binary Large Object) analysis. This method is used so as to identify the objects/regions called BLOBs using pixel connectivity and further more parameters like area, centroid, and number of BLOBs available in the test image. Preprocessing of the input test image is done prior to the BLOB analysis like thresholding to get a binary image. The number of objects identified is displayed or read out using a speaker interfaced to the hardware. The hardware Raspberry Pi, running with UBUNTU operating system flashed into a memory card, works as a standalone device with the Raspberry Pi camera being used as an image acquisition device to constantly capture image frames of objects of interest.

**Key words:** BLOB analysis, object counting, Raspberry Pi.

## I.INTRODUCTION

The object counting is one of the important image processing techniques in industrial applications. This includes counting the number of objects that passing a conveyor belt or counting shapes. Some of the basic operations involved in such process normally thresholding, labeling. Initially thresholding is to convert a grayscale image into a binary image and labeling is to group those of pixels belonging to a same body as a one object and those of others as another object. Object counting is a simple task it is depends on detection of target objects. The conventional method [1] is regularly used for object counting but it is a manual and non-automatic form. Obviously counting of objects leads to eye fatigue and affects the accuracy of results but it is a time consuming. But the purpose of automatic counting is reliable and reproducible. Normally a user gets different results in each measurement when counting manually but in automatic counting consistently obtain a unique value.

But object counting is a challenging problem to count the target objects in high speed and it is used to acquire certain number of elements from images. Counting the objects in image is essential in numerous applications, such as counting the people on road, packaging and quality control [2] in industrial system etc. In present process work uses a powerful device like Raspberry pi for object counting. It is a small size, low cost and light weight credit-card sized single board computer. In generally objects on an image are labeled and indexed by finding algorithm but it is time consuming. In this project when utilize an object counting

algorithm that can easily count and display the number of objects and counted number will be speaks with the help of speakers in real time. So that there no time consuming.

Object counting

Here the object counting is done using the following three steps that is thresholding, Blob analysis, cell segmentation.

a. Thresholding:

The process of thresholding is a gray-scale image will be converted into a binary image (that is the image consisting only black and white pixels). To compare every pixel in the gray-scale image with a threshold value say  $T_d$ , considerable the value of the pixel exceeds that of  $T_d$  value than that pixel will be set to 1 (white pixel) otherwise it will be set to 0 (black pixel). This process can be expressed mathematically that is.

$$f(x, y) = \{ G_0(x, y) \text{ if } G(x, y) > T_d \quad (i)$$

$$f(x, y) = \{ G_1(x, y) \text{ if } G(x, y) \geq T_d$$

Here  $G(x, y)$  – gray level for input pixel

$f(x, y)$  – gray level for output pixel

$T_d$ – threshold value

By utilizing equation (i) the values of each pixel from the input image are replaced by the similar pixel in the target image using  $G_0(x, y) = 0$  and  $G_1(x, y) = 1$ .

b. BLOB analysis:

BLOB analysis [3] is one of the fundamental techniques and it consist a set of related (connected) pixels in a binary image. In image processing a BLOB is described as area (region) of connected pixels. Generally the recognition of

BLOB and learning of regions in an image is the important phenomenon of BLOB analysis. The standardization of BLOB refers “Binary Large Object”. The large specifies that only objects of a certain size are of concern object. The main advantages of the blob analysis technique consist of high flexibility and excellent performance. In an image all pixels that belong to a BLOB are in a foreground state (pixels with non zero value) and all other pixels are in a background state (pixels with zero value). In numerous application where calculations are time-consuming, when use blob analysis to remove blobs based on their spatial characteristics and also find the statistical information as the size of blob or the numbers, location, centroid and presence of blob region.

i) Area:

The number of pixels in an image had been considered as an area of BLOB. The area of touching pixels with the same logic state is known as a BLOB. For this characteristic is often used to remove BLOBs that are too small or too big from the image.

ii) Bounding box:

The Bound Box of a BLOB is minimum rectangle which having the BLOB, it defined by going through all pixels for a BLOB and finding the four pixels with minimum x, y values and maximum x, y values correspondingly. The bounding box ratio of a BLOB is defined as the height of the bounding box divided by the width and the compactness of a BLOB is defined as the ratio of the BLOBs area to the bounding box.

$$\text{compactness} = \frac{\text{Area of BLOB}}{\text{Width} \times \text{height}} \quad (\text{ii})$$

iii) Centroid:

The centroid (or centre of mass) for a binary image is alike it is the average x and y positions of the binary object. The centroid is described as a point whose x value is computed by summing the x coordinates of all pixels in the BLOB and then dividing by the total number of pixels similarly for y-value. Mathematically center of mass (xc, yc) is computed as

$$X_c = \frac{1}{N} \sum_{i=1}^N x_i, \quad Y_c = \frac{1}{N} \sum_{i=1}^N y_i \quad (\text{iii})$$

Where N = number of pixels in the BLOB xi, yi = x and y coordinates of n pixels

c. Cell segmentation:

Segmentation of cells is the fundamental approach method in digital image processing. In that cells were divided from the image back ground by using the foreground-back ground intensity is based on pre-processing step. In simple thresholding, the touching cells are not separated, instead the cells can be divided using [4] watershed segmentation. This segmentation separates touching cells where the edge between them is the darkest. Applying the watershed segmentation to these images the cell edges are not correctly positioned. So that changes the shapes and positions were compensated by thresholding.

In generally to detect the cell using edge detection and basic morphology. The segmented objects are greatly changes in contrast from back ground image. The operators can be detected these changes in contrast and also calculated the gradient of an image and threshold can be applied to a binary mask having the segmented cell. Dilate the image by binary gradient mask that shows lines of high contrast in the image. But these lines are not quite it describe the outline of the object of interest. The line gapes will be disappearing if the image is dilated using linear structuring elements.

## II. HARDWARE AND SOFTWARE REQUIREMENTS

### A. Hardware Requirements:

This project requires some hardware components such as Raspberry pi, RPI camera, power supply, Audio speakers.

#### a) Raspberry pi:

The design of this project uses Raspberry pi (model B). Raspberry Pi is a single board computer with Linux or other small operating systems. It was developed by Raspberry Pi foundation in UK for the use of computer science education. The second version of the Raspberry Pi is used in this project.

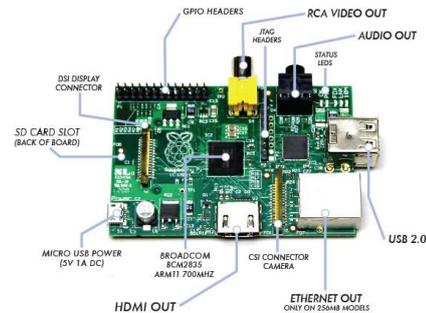


Fig.2.1.Raspberry pi

It consists of an ARM 1176JZF-S processor, which runs at 700MHz clock speed, 512MB SDRAM shared with GPU, a Video Core IV GPU, 2 USB port, 1 100 M bit/s Ethernet port, one video and audio output, one HDMI output. It also has 26 pins including 8 General purpose Input/output (GPIO), one SPI bus, one I2C bus, one UART bus and 3.3V, GND and 5V. The Raspberry Pi needs an external Secure Digital (SD) card to store its operating system and also all the user data.

Hence the Raspberry pi can be used as a really powerful microcontroller which can accomplish almost any functions, and also it can act as a normal use computer with keyboard, mouse and monitor connected.

#### b) Raspberry pi camera:

The Raspberry pi camera board plugs directly into the CSI connector on the Raspberry pi. The Raspberry pi camera module attaches to Raspberry pi by way of a 15 pin Ribbon cable to the dedicated 15-pin MIPI camera serial

Interface (CSI) which was designed especially for interfacing to cameras. It's able to deliver a clear 5 mega pixel resolution image or 1080p HD video recording at 30frames/sec.



Fig.2.2 Raspberry pi camera (NOIR)

c) Power supply:

The power supply on Raspberry pi is quite simple. This uses a Micro USB connection to power itself and the micro USB connection ability of supplying at least 700mA at 5v. Probably normal mobile phone charges are appropriate and do not efforts to power of Raspberry pi from a USB port of another computer or hub because they are frequently not capable of supplying the required current.

B. Software implementation:

Software implementation of this work uses Raspberry pi and MATLAB (R2014a) software. The present model is developed in SIMULINK which supports Raspberry pi support packages. SIMULINK provides an interactive graphical environment and customizable set of block libraries not only signal, image, video processing systems but also communication and control systems. SIMULINK let users, design, simulate, implement and test their projects.

### III. PROPOSED MODEL

a) Proposed work:

The above show the block diagram of proposed model. At first the RPI camera is connected to Raspberry pi and object counting program is performed in SIMULINK. In order to know how many objects are counted within particular area we will connect the speaker to Raspberry pi it speaks the counted object number.

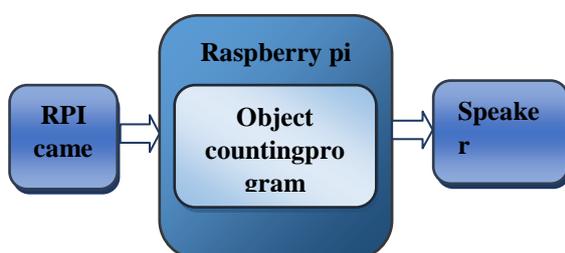


Fig 3.1. Block diagram of proposed model

b) Process steps:

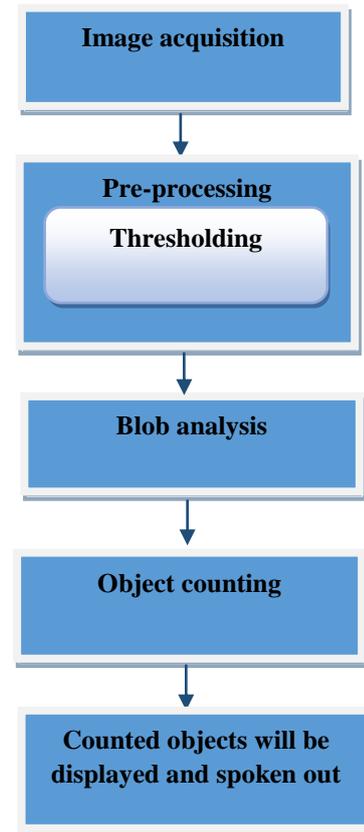


Fig 3.2 Process steps

The above steps are described the process steps of proposed work.

i) Image acquisition:

This step intends to capture the image through camera. Acquire the scene from the Raspberry pi camera. And then convert the raw video frame to RGB color space.

ii) Pre-processing:

It is required to pre-process the captured video frame in order to extract the features. The following step is performed in pre-processing. The first step in pre-processing is Automatic Thresholding where the captured color video frame is converted to gray scale and then to binary. In binary representation of the image, the pixels are valued as '0' and '1' which helps in feature extraction. The next is Edge Detection using which the edges in the captured frame will be extracted in order to get the object contours.

iii) BLOB analysis:

BLOB analysis is a fundamental technique utilizes for computational purpose by using these analysis to calculate the features of objects in a given image. Obviously the performance of Blob analysis operation depends on a successful segmentation of the image that is good BLOBs are separated from the back ground and other elements (noise) was eliminated in the image. Usually segmentation involves a Binarization operation.

iv) Object counting:

This step assigns a label to the detected object based on its descriptors and also number of objects in an image will be counted. The number of counted objects will be displayed with the help of video display.

c) SIMULINK model of proposed work:

The whole programming is developed in SIMULINK which supports the Raspberry pi packages. The raw video frames converted into gray scale image and after performing pre-processing then in BLOB analysis it computes the perimeters like area, centroid, etc. After completion of process the output images will be displayed in video viewer. The following figure 3.3 shows the SIMULINK model of proposed work.

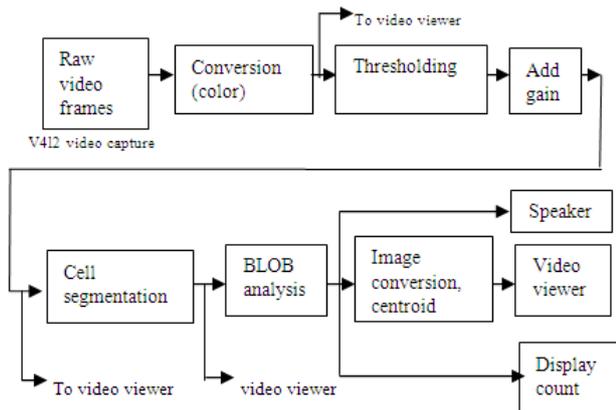


Fig 3.3. Simulink model of object counting and speak out count object number

**V. EXPERIMENTAL RESULTS**

(a) Original image:

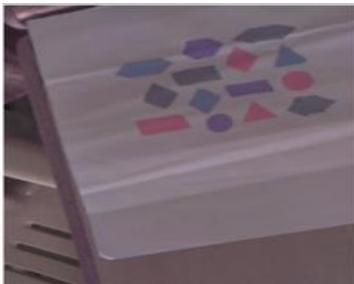


Fig.4.1. Original image

(b) Gray Scale image: The gray scale image contains only brightness information. In grayscale image each pixel is represented by a byte or word.



Fig.4.2. Gray scale image

(c) Thresholding image:

The process of thresholding is to convert the grayscale image into a binary image that is the image consists of only black and white pixels.

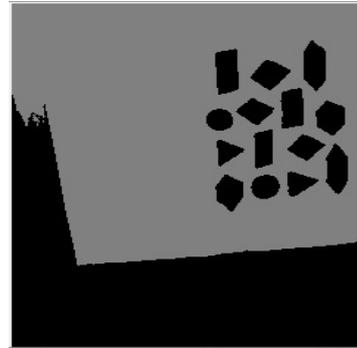


Fig .4.3. Threshold image

(d) Cell segmented image: In this subdividing the cells into its constituent areas to detect the cells by using edge detection. Edge detection using which the edges in the captured frame will be extracted in order to get the object counter.

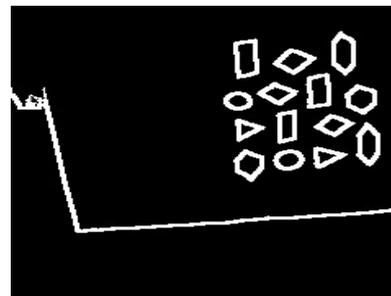


Fig . 4.4. Cell segmented image

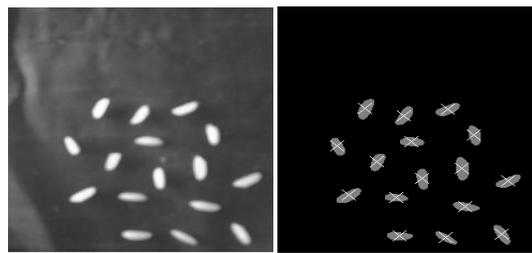
(e) Object counting: The objects will be counted based on cross marks. Several methods are used in object counting that is connected components analysis, statistical area measurements etc.



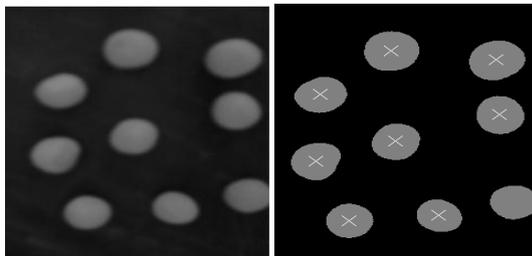
Fig .4.5. Object counting based on cross mark

Other examples:

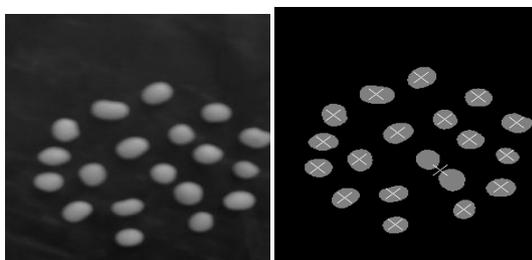
Here to take different objects for the purpose of testing.



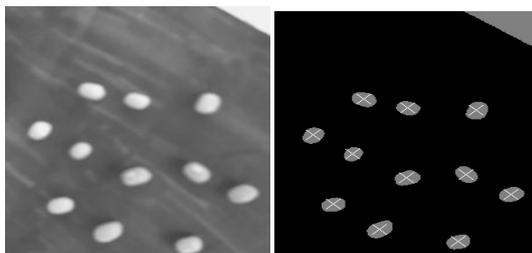
(a) Rice (b) Identified objects



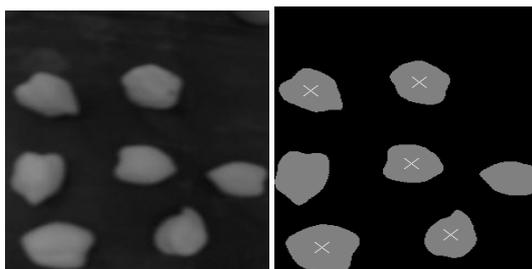
(c) Green peas (d) Identified objects



(e) Dal (f) Identified objects



(g) Green gram (h) Identified objects



(i) Chick peas (j) Identified objects

Fig : (a) (c) (e) (g) (i) illustrates that the gray scale images of different objects (Rice, Green peas, Dal, Chick peas, Green peas) and (b) (d) (f) (h) (j) is the centroid based counting.

The above figures illustrate the different shapes of object images like rice, green peas, Green gram, chick peas, dal etc. When cross mark is arrive at center of the object then the number of objects will be counted. The following table shows that the experimental values for different objects.

S.No	Image	Available objects	Counted objects	success rate%	Failure rate%
1	Rice	16	16	100	0
2	Green peas	9	8	99	1
3	Dal	20	18	98	2
4	Green gram	11	11	100	0
5	Chick peas	7	5	98	2

Table .1.Experimental values for different objects

The given table shows that the list of different objects and corresponding to the number of available objects, counted objects and their success rates% and failure rates%. For example the actual count of Rice and Green gram is 16, 11 after testing the displayed number is 16, 11 and their success rates is 100% and failure rates is 0%. But the actual count of Green peas is 9 after testing the displayed number is 8 so that the success rate of green peas is 99% and 1% is their failure rate. Similarly the real count of Dal and Chick peas is 20, after testing the displayed numbers is 18, 5 corresponding success and failure rates are 98%, 2%.

## VI. CONCLUSION AND FUTURE SCOPE

The system realizes real-time object counting with voice. The main intention of this project is developing an algorithm that can count and identify the objects in a given scene. The features of objects will be easily computed by BLOB analysis this method distinguishes itself from other computer vision based approaches and the system works will be done in standalone device. This makes the system inexpensive to use. As a future scope, the output of the proposed algorithm can be used in traffic surveillance for counting and detecting the vehicles.

## REFERENCES

- [1] Amruta Pandit, Jyothi Rangole "Literature Review on Object Counting Using Image Processing Techniques" International Journal of Advanced Research In Electrical, Electronics and Instrumentation Engineering, vol.3, Issue 4, April 2014.
- [2] Selcuk Sevgen, Fethullah karabiber "Implementation of a CNN based Object Counting Algorithm on Bi-Cellular Vision System, Isthambul University, Department of computer engineering, Turkey.
- [3] Sarah Sookman Matrox Imagining "Blob Analysis and Edge Detection in Real World" August 2006, <http://www4.evaluationengineering.com>.
- [4] E. Bengtsson, J. Lindblad, "Robust cell image segmentation methods", center of image analysis, Uppsala University.
- [5] Md. Maminul Islam , Md. Sharif uddin Azad, Md.Asfaqu, Alam, Nazmul, Hussan," Raspberry pi and image processing based EVM", International Journal of Scientific & Engineering Research ,volume 5, Issue 1, January-2014.
- [6] G. Senthil Kumar, Gopal Krishna, V.sathis kumar, "Embedded Image capturing System using Raspberry pi", volume 3. Issue 2, March-April 2014 ISSN 2278-6856.
- [7] <https://www.cl.cam.ac.uk/projects>, physical computing with raspberry pi.