International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 03 | Mar-2016 www.irjet.net p-ISSN: 2395-0072

ARM Controller Based Object Recognition

K.S.Sandhya[1], Pokuru Divya[2], Prajakta Saitwal[3]

[1]M.TechEmbedded Systems, School of Electronics Engineering, VIT University, Vellore, TamilNadu ,India
[2] M.Tech Embedded Systems, School of Electronics Engineering, VIT University, Vellore, TamilNadu, India
[3] M.Tech Embedded Systems, School of Electronics Engineering, VIT University, Vellore, TamilNadu, India

---***---

Abstract - *The proposed object recognition system is aimed at recognising the shapes of the objects in an image, identifying their colour and counting the objects based on the distinction. This paper focuses on recognition of standard shapes and primary colours. The system is implemented using ARM (Advanced RISC machine) based Raspberry Pi 2 model B controller and the algorithm is mainly based on the techniques of image segmentation and blob analysis. The shape analysis is done by estimating the object aspect ratio and extent by building bounding boxes and the colour estimation is done using the mean colour value of the object.*

Key Words: Object recognition, machine vision, shape, colour, oval, circle, contours, aspect ratio, extent

1. INTRODUCTION

Machine vision is one of the applications of computer vision or image processing to industry and manufacturing scenario aiming at automation of processes like object classification, sorting, object counting and monitoring. One of the most common applications of machine vision is the inspection of packages, cargos, manufactured goods such as semiconductors chips, foods and pharmaceuticals. This inspection and classification is based on the fact that each of the object that is monitored has distinguishing features from the others, in terms of shape descriptors like extent, aspect ratio or colour. This uniqueness is what makes them discernible from each other. Thus the proposed ARM controller based object recognition is based on fundamentals of machine vision wherein by image processing, counting and identification of all the shapes and their colours in an image is possible. The system is implemented using Raspberry Pi 2 model B and the algorithm is mainly based on the techniques of image segmentation and blob analysis. This identifies objects in the image that are clearly discernible from the background.

2. ARM BASED OBJECT RECOGNITION

Given an image which consists of a set of objects which are of standard shapes and colors, the system produces the outputs namely count of objects, shape of object, colour of object.

2.1 Features

The features of the object recognition system designed using the ARM processor, are:

- The algorithm is found to clearly identify the various objects in the image
- It can recognize and distinguish between the following shapes: square, rectangle, circle, oval and triangle; with others classified as “Other shapes”.
- The color of the objects will be detected and classified based on colour as Red, Green, Blue, Magenta, Cyan, Yellow, Black and White.
- The system produces the count of the total number of the objects in the image, as well as the shape-wise count.
- The algorithm takes into consideration the orientation of each object in the image. Thus even if an object in the image is rotated by an angle, the system can identify its shape with good accuracy.
- The algorithm can produce reasonable results for images consisting of objects with connected edges, wherein the object boundaries could be touching each other; as shown in the simulation results.
- The system is found to work reliably with real time images instead of just computer generated images.

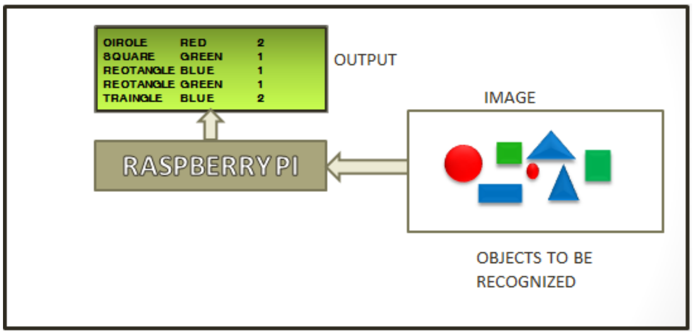

Fig -1: Block diagram

2.2 Hardware

The object recognition system is designed using the ARM processor based microcontroller. The Raspberry Pi is a controller which has ARM cortex A7 CPU, 1GB RAM , 900 Mhz quad core, 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port, Camera interface , Display interface, Micro SD card slot.The Picamera can be interfaced to the Pi for the imaging via the camera interface. The Raspberry Pi does not host any internal ROM for the OS and hence the SD card is used as a bootable device for the Raspbian OS. Additional USB devices such as mouse, keyboards can be connected along with a monitor for display via the HDMI port for the Input – Output peripherals.

2.3 Software

The system software is the Debian Raspbian OS that is stored on the Micro SD card. The algorithm is programmed with Python, after importing the OpenCV library for image processing. OpenCV outdoes the use of MATLAB in our application as OpenCV is an open source real-time image processing unlike MATLAB that is licensed and has constraints for use in Raspberry Pi. Also in terms of speed, OpenCV is found to process faster as it is a library of functions that is written in C, close to the machine language unlike MATLAB that is built on Java. While OpenCV supports various programming languages like Python or C++, Python is viewed as a relatively better option. This is mainly because of its simplicity and code readability. Python can also be easily extended with C/C++. With this it is possible to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This is advantageous because the code is as fast as the original C/C++ code and it is easier to code in Python than C/C++. In this way, the OpenCV-Python which is a Python wrapper for the original OpenCV C++ implementation is more convenient to use.

3. IMPLEMENTATION AND WORKING

The implementation of the system is based on the working of the algorithm and the hardware design.

3.1 Flowchart and Algorithm

The object recognition is mainly based on the image segmentation and blob analysis techniques. Thus it involves extracting each of the blobs in image separately and analysing their shape descriptors. While the extent and aspect ratio values are used to distinguish between the shapes, the RGB component of each blob is used in the colour estimation.

The flowchart for the shape detection and colour recognition are as follows:

3.1.1 Shape recognition

The input image as shown consists of the following objects (from left end): a yellow lid, a coin, a ring, a fluorescent cap and an orange coloured note. The color image of the objects is first read and subjected to denoising. This is then converted to grayscale with successive filtering. The conversion of the grayscale image to black and white is done using the Canny edge detection. Canny edge detection algorithm converts the grayscale image to binary involving the following steps.

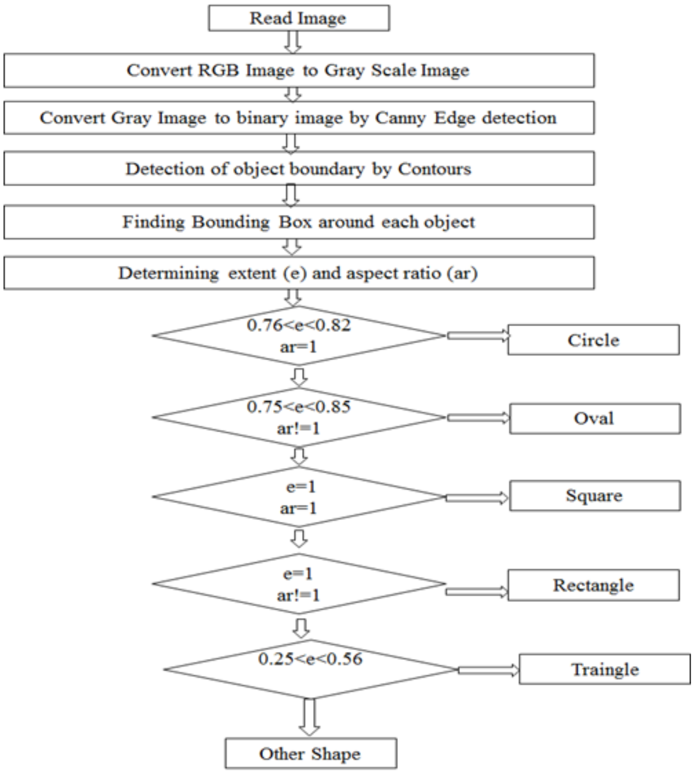

Fig -2: Flowchart for shape detection

The first is the noise reduction by a Gaussian filter. The next is finding the intensity gradient of the image and the smoothened image is filtered using Sobel kernel. Then suppression with upper threshold is done to remove the unwanted pixels. This is then subject to hysteresis with lower and upper threshold. Thus the gray image is effectively converted to black and white with edge detection.

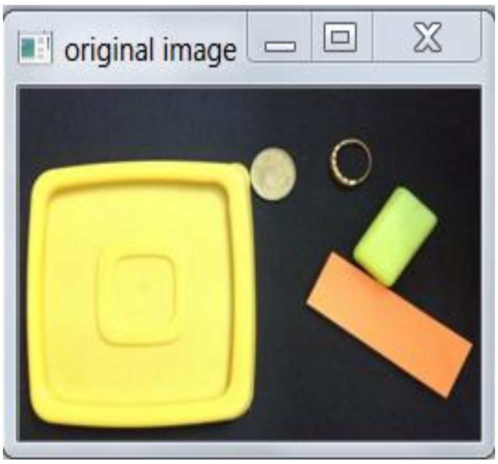

Fig -3: Input image

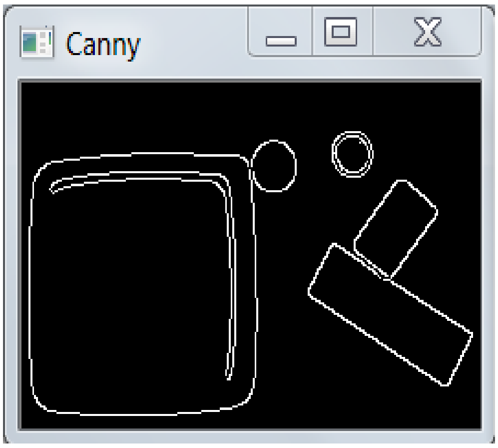

Fig -4: Canny edges

Using this binary image, contours are found for the objects and are then approximated. Contours are a curve joining all the continuous points (along the boundary), having same colour or intensity. For two or more object boundaries touching each other, there is a hierarchy of contours of which the innermost contour is to be identified to rightly extract the connected components. [3]

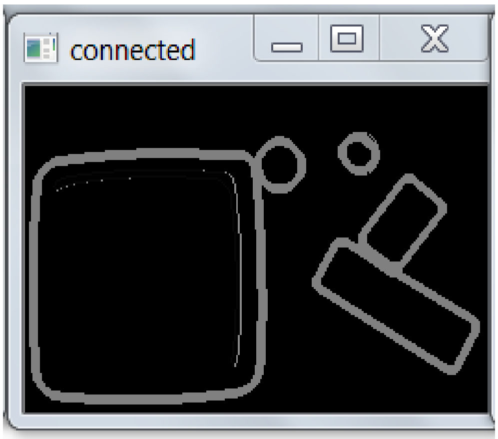

Fig -5: Connected components

In order to display the regions of interest, a mask of zeroes is created and mapped onto each of the extracted blob. To find out the extent and aspect ratio of each object, it is necessary to build bounding boxes around each object, also considering the orientation of each object. For this a rotated bounding box around the object is to be built , in which the angle of rotation of the bounding box is equal to the orientation of the object, but in the opposite direction.

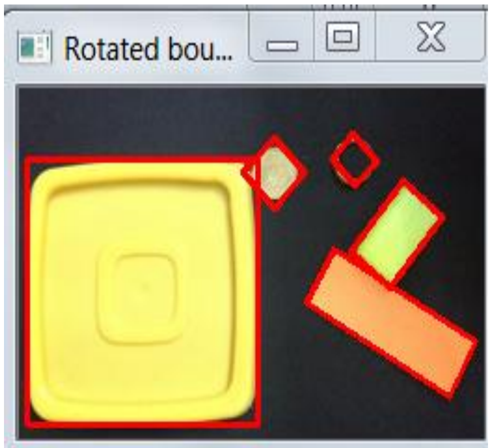

Fig -6: Rotated bounding boxes around each object

From the dimensions of each bounding box, the extent and aspect ratio of each object can be computed as below. [2]

Extent = (Contour area) / (Bounding box area)
Aspect ratio = (Width of bounding box) /
(Height of the bounding box)

These values thus evaluated can be used as shape descriptors to recognize the shape of the objects. In this way distinct recognition of squares, rectangles, circles, ovals and triangles in the image is possible.

3.1.2 Colour detection

Once the object contours are extracted, the mean colour of each object in terms of the R, G, B components must be extracted. Out of the three, the component identified with the highest value shows the dominant colour of the object. Thus every object gets classified as one of these: red, green, blue, black or white coloured object. [2] However ,in our work, by converting the RGB image to a HSV scale, with the hue values, the object colour can also be identified as yellow, cyan and magenta, with further precision. Thus converting to the HSV scale gives more precise classification. The object colour can be now broadly classified into black, white, red, yellow, green, cyan, blue and magenta using the Hue values alone.

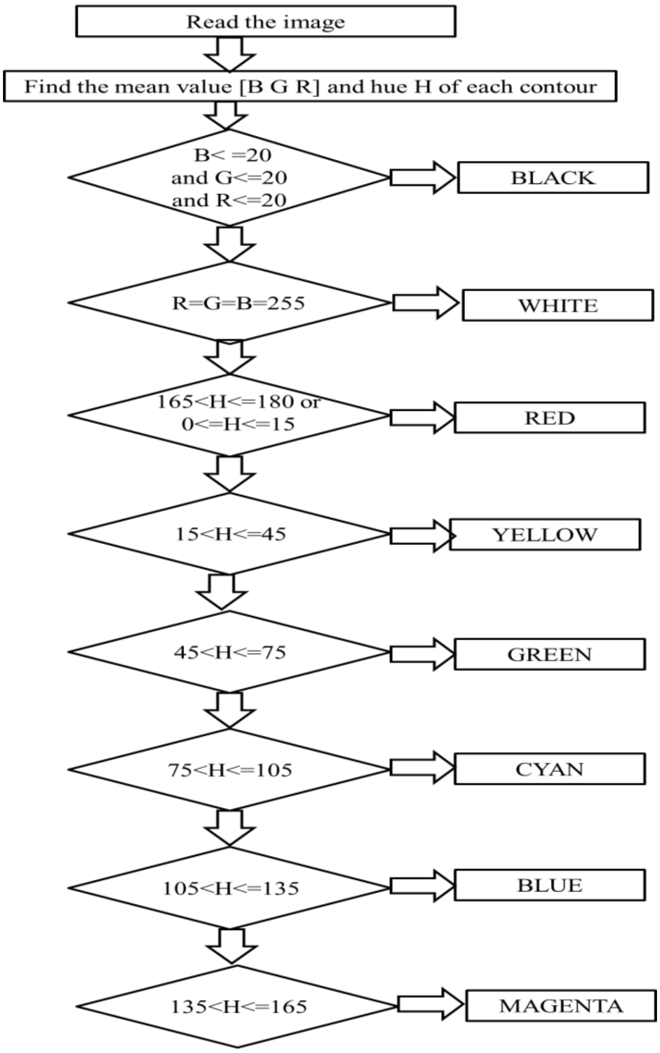

Fig -7: Flowchart for colour detection

```
if(b<=35) and (g<=35) and (r<=35):
      color='black'
elif (g==r==b==255):
      color='white'
elif (h>=165) and (h<=180):
        color= 'red'
elif (h>=0) and (h<=15):
        color= 'red'
elif (h>15) and (h<=45):
        color='yellow'
elif (h>45) and (h<=75):
        color='green'
elif (h>75) and (h<=105):
        color='cyan'
elif (h>105) and (h<=135):
        color='blue'
elif(h>135) and (h<165):
        color='magenta'
```

Fig -8: Colour detection

3.2 Hardware implementation

The algorithm is programmed and implemented with OpenCv Python on the ARM processor platform. The ARM based microcontroller which involves the ARM Cortex A7 can be configured with a dedicated monitor for display via the HDMI port or as a headless system, suing the secure shell (SSH) connection from another computer. [1]

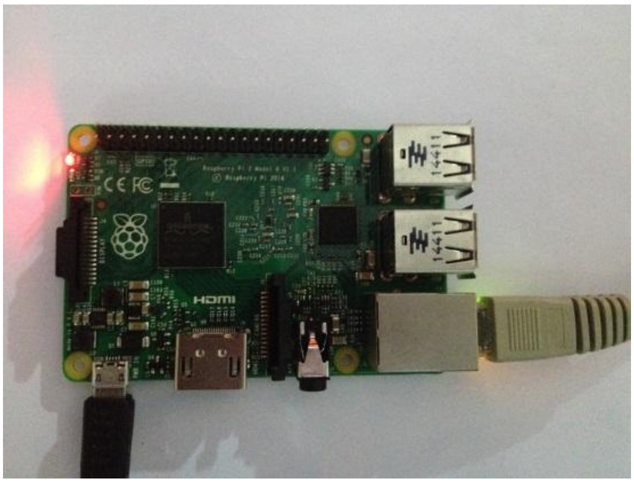

Fig -9: Hardware implementation on Raspberry Pi

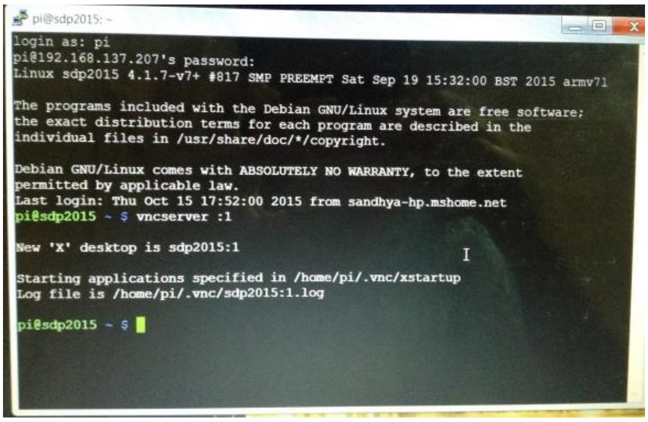

Fig -10: SSH connection to raspberry pi

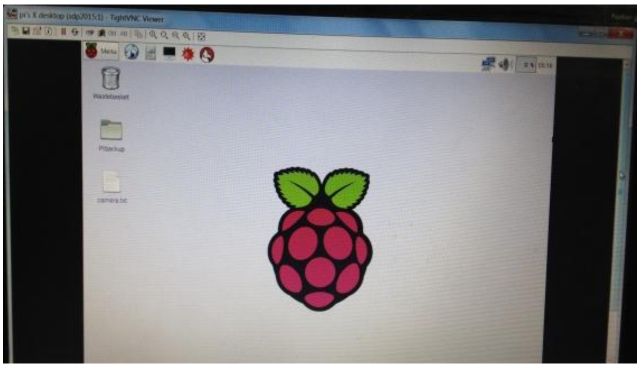

Fig -11: GUI of the Raspberry PI

4. SIMULATION RESULTS

On executing the Python code with the Python IDLE on the ARM processor, and on testing it with possible test cases, the algorithm is found to have an accuracy of about 90%. The following are the simulation results obtained for the input image as shown.

Fig -12: Input image

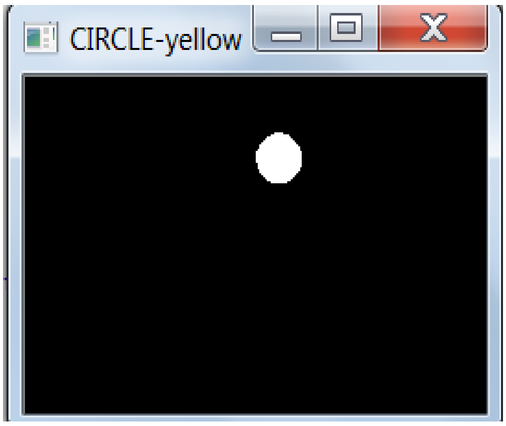

Fig -13: Circle – Yellow

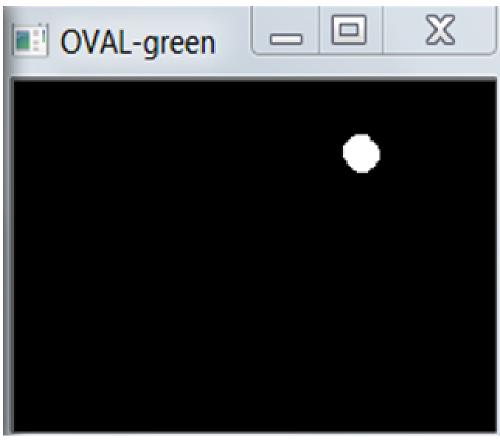

Fig -14: Oval- Green

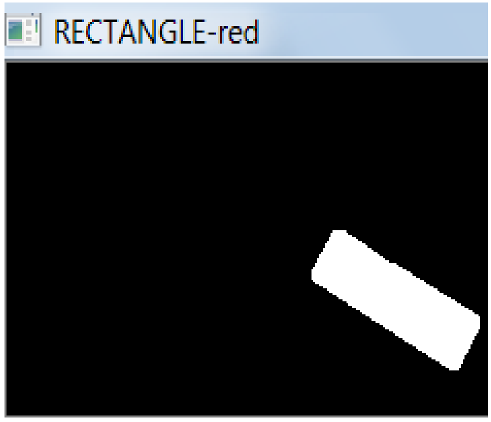

Fig -15: Rectangle - Red

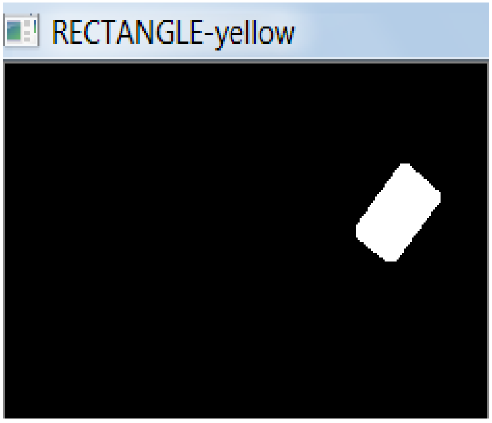

Fig -16: Rectangle – Yellow

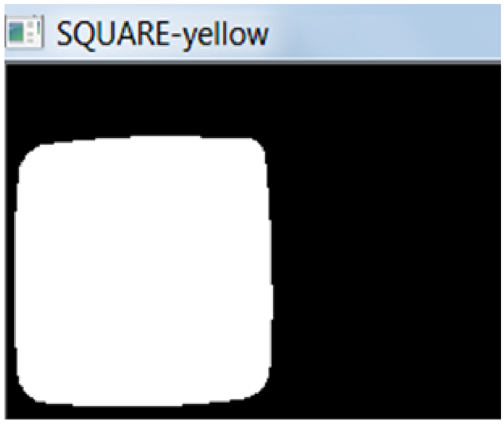

Fig -17: Square – Yellow

Fig:18 gives the textual output of the image processing code, displaying the number of objects of each shape based on the classification results and also the total number of objects (blobs) identified in the image.

```
Number of triangles is              :0
Number of circles is                :1
Number of rectangles is             :2
Number of ovals is                  :1
Number of squares is                :1
Number of other shape objects is    :0
Total number of objects is: 5
```

Fig -18: Results – Count of shapes

The simulation was also carried out with an image consisting of a single object, which is a pen drive as shown. The proposed image processing algorithm could also clearly distinguish between the inner and the outer contours in the object, thus distinctly recognizing the object's boundary.

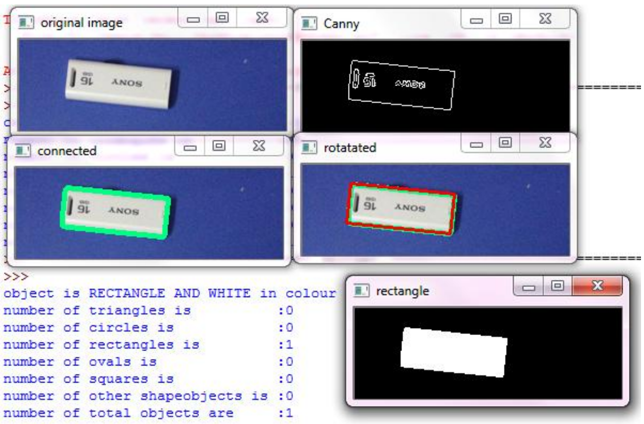

Fig -19: Results – For second input image

The objects above were captured as an image with a 5MP camera in a well illuminated and a balanced environment to eliminate shadows in the image.

5. CONCLUSIONS

From the above, we can conclude that the ARM based object recognition system is a low cost solution to automated object classification and inspection in industries. This project not only optimizes the cost of implementation but also produces considerably fast results on recognizing the objects. For an industrial scenario wherein a real time dedicated system is required for classification, counting and sorting of objects, the system implementation can be extended to camera interfacing on the input side and a conveyor belt system for sorting or a robotic arm for pick and place operations. Furthermore, this is a generic algorithm devised and implemented on functionally powerful hardware platform and can thus be suited for a wide range of applications like identification of traffic signs and signals, identification of cargo containers and packages based on colours, in medical imaging to estimate shape changes in cells or tissues related to illness.

REFERENCES

[1] Viren Pereira, Vandyk Amsdem Fernandes and Junieta Sequeira "Low Cost Object Sorting Robotic Arm using Raspberry Pi " IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS) , September 26-27, 2014

[2] Shambhavi Vijay Chhaya, Sachin Khera, Pradeep Kumar S, "Basic Geometric Shape And Primary Colour Detection Using Image Processing on MATLAB " , IJRET: International Journal of Research in Engineering and Technology ,vol. 4 eISSN: 2319-1163 | pISSN: 2321-7308 , May-2015

[3] Nidhal El Abbadi1 and Lamis Al Saadi "Automatic Detection and Recognition of Different Shapes in an Image " , IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, November 2013

BIOGRAPHIES

K.S.Sandhya is a graduate in Electronics and Communication Engineering from St.Joseph's College of Engineering, Chennai. After a work experience of 15 months, she is currently pursuing her first year of post-graduation studies in Embedded Systems at VIT University, Vellore.

Pokuru Divya is a first year post graduate student pursuing her Masters in Embedded Systems at VIT University Vellore, Tamil Nadu. She obtained her Bachelor's degree in Electronics and Communication Engineering in the year 2013 from Bangalore Institute of Technology, India and has also completed her Master's degree in Public Administration from University of Madras in 2015.

Prajakta Rajendra Saitwal is currently pursuing her Masters in Embedded Systems at VIT University Vellore, Tamil Nadu. She has completed her undergraduation in Electronics and Communication Engineering at WIT Solapur Her areas of interest are image processing and microcontrollers.