**Colour Detection**

**Introduction:**

A RGB(red, green, blue) image is a colorful image consisting of fixed values of color contents for each pixel. These color contents have different values ranging from 0 to 255. This project is for RGB color detection in an image. The project will detect the color according to the need of the user and then will make outline around that color. Additional feature of the project is that it will also predict the center .

**Methodology:**

The main steps are explained below:

* **Capturing an Image**

Foremost step of this algorithm is to capture the good quality image for processing which should be done using high eminence camera, good illumination conditions. Good image means which is having high resolution and contrast, free from noise and proper depth of field. Then this image is imported to the MATLAB using Command:

IMG=imread ('TEP.jpg');

* **Preprocessing:**

In this step noise will be removed using median filter the holes are filled and edges are smoothened to make sure a perfect boundary line. Then the connected components are labelled in 2D image and region is measured.

* **Extraction of RGB color**

A RGB image is a blend of three different color layer (red, blue and green). In this step, extract these color contents differently. These color contents are extracted by using some inbuilt functions or some commands in MATLAB.

red=IMG(:,:,1);

green=IMG(:,:,2);

blue=IMG(:,:,3);

* **Center prediction:**

After extracting color or detecting color it will calculate the size of the object and then will predict the center and x and y co-ordinates of the center of object.



**Code:**

data = imread('TEP.jpg');

    diff\_im = imsubtract(data(:,:,2), rgb2gray(data));

    %Use a median filter to filter out noise

    diff\_im = medfilt2(diff\_im, [3 3]);

    diff\_im = im2bw(diff\_im,0.18);

    diff\_im = bwareaopen(diff\_im,300);

    bw = bwlabel(diff\_im, 8);

    stats = regionprops(bw, 'BoundingBox', 'Centroid');

    % Display the image

    imshow(data)

    hold on

    for object = 1:length(stats)

        bb = stats(object).BoundingBox;

        bc = stats(object).Centroid;

        rectangle('Position',bb,'EdgeColor','r','LineWidth',2)

        plot(bc(1),bc(2), '-m+')

        a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2)))));

        set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'black');

    end

    hold of

    data = imread('TEP.jpg');

    diff\_im = imsubtract(data(:,:,1), rgb2gray(data));

    %Use a median filter to filter out noise

    diff\_im = medfilt2(diff\_im, [3 3]);

    diff\_im = im2bw(diff\_im,0.18);

    diff\_im = bwareaopen(diff\_im,300);

    bw = bwlabel(diff\_im, 8);

    stats = regionprops(bw, 'BoundingBox', 'Centroid');

    % Display the image

    imshow(data)

    hold on

    for object = 1:length(stats)

        bb = stats(object).BoundingBox;

        bc = stats(object).Centroid;

        rectangle('Position',bb,'EdgeColor','r','LineWidth',2)

        plot(bc(1),bc(2), '-m+')

        a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2)))));

        set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'black');

    end

    hold of

    data = imread('TEP.jpg');

    diff\_im = imsubtract(data(:,:,3), rgb2gray(data));

    %Use a median filter to filter out noise

    diff\_im = medfilt2(diff\_im, [3 3]);

    diff\_im = im2bw(diff\_im,0.18);

    diff\_im = bwareaopen(diff\_im,300);

    bw = bwlabel(diff\_im, 8);

    stats = regionprops(bw, 'BoundingBox', 'Centroid');

    % Display the image

    imshow(data)

    hold on

    for object = 1:length(stats)

        bb = stats(object).BoundingBox;

        bc = stats(object).Centroid;

        rectangle('Position',bb,'EdgeColor','r','LineWidth',2)

        plot(bc(1),bc(2), '-m+')

        a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2)))));

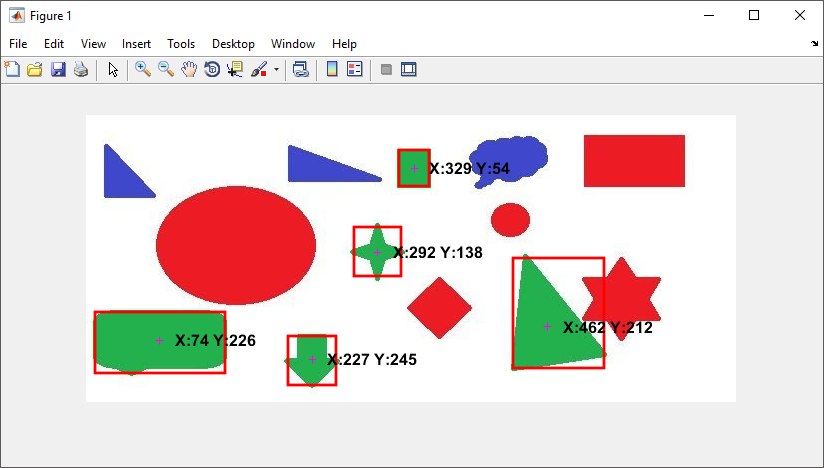
        set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'black');

    end

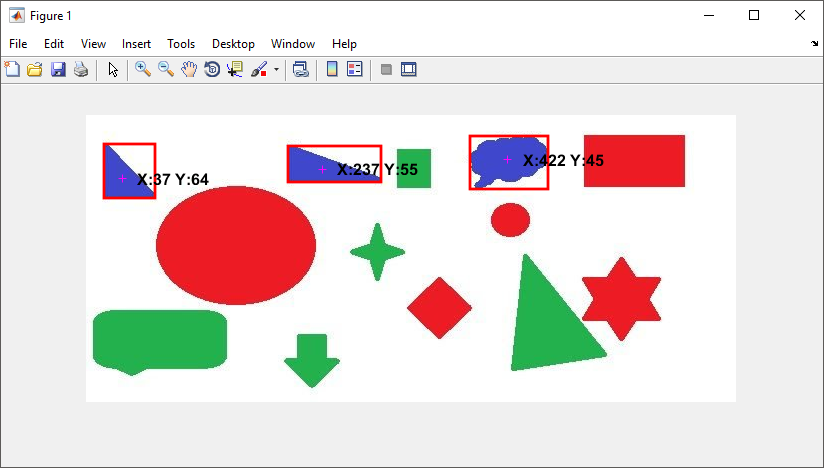
    hold of

**Output:**

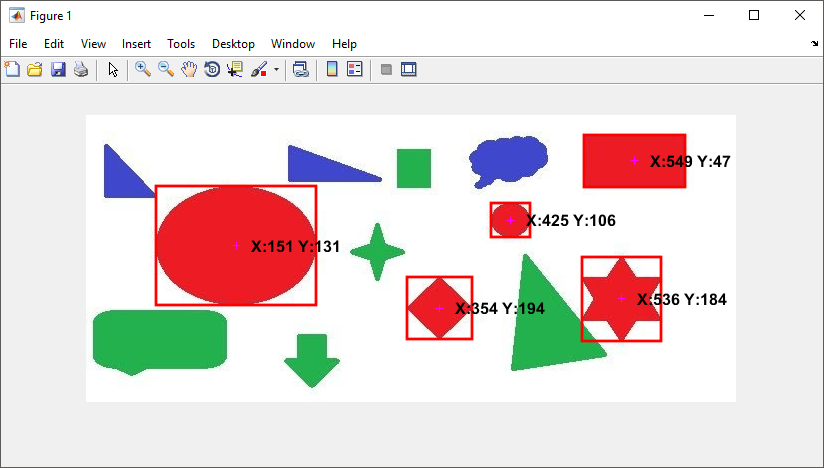
**Green :**



**Blue:**



**Red:**



You can see in the above figure, The shapes with colours are detected.

The + sign indicates the centre of each detected shape. X and Y are the x,y coordinates of the centre point of each shape which are shown in black for each detected shape.

**Conclusion:**

A methodology is defined to get the required color filed from an RGB image. In this various steps are implemented using MATLAB platform. Maim positive point of this method is its speed, boundary and the center prediction of the object.

**References:**

**[1]** Rafael C. Gonzalez (University of Tennessee), Richard E. Woods (MedData Interactive) and Steven L. Eddins (The MathWorks, Inc.), in „Digital Image Processing Using MATLAB‟ Second Edition,2009 by Gatesmark, LLC.

**[2]** Alasdair McAndrew, in „An Introduction to Digital Image Processing with Matlab, Notes for SCM2511 Image Processing1‟, School of Computer Science and Mathematics, Victoria University of Technology.

**[3]** Digital image processing using Matlab -Gonzalez woods & Eddins

**[4]** R. S. Berns, “Principles of Color Technology” (3rd edition New York: Wiley, 2000)

**[5]** G. Wyszecki and W. S. Styles, “Color Science: Concepts and Methods, Quantitative Data and Formulae” (2nd edition New York: Wiley, 1982)

**[6]** J. L. Vincent, “Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms”, IEEE Transactions on Image Processing, vol. 2, pp. 176–201, 1993.

**[7]** Devi, H.K.A., (2006). Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script. Ancient Asia. 1, pp.161–165.

**[8]** The Multi-stage Approach to Grey-Scale Image Thresholding for Specific Applications, Van Solihin and C. G. Leedham

**[9]** Document Image Analysis by Rangachar Kasturi, Louis Lam, Seong - Whan Lee & Ching Y. Suen.

**[10]** M. Sezgin and B. Sankur, “Survey over Image Thresholding Techniques and Quantitative Performance Evaluation”, Journal of Electronic Imaging, vol. 13, no. 1, pp. 146–168, Jan. 2004.