School of Electronics and Communication Engineering

Third Year B. Tech.

DSP PROJECT REPORT

(PBL ACTIVITY)

Academic Year 2021-2022 Trimester-VII

Name of Students: Tanishq Todkar (PA8) , Vedika Shinde(PA14) , Tenzing Monkyi (PA22)

PRN Number : 1032190197,1032191001,1032191135 respectively

Batch: A1

Project guide : Prof : Rupali Kute

S. No. 124, Paud Road, Kothrud, Pune-411038 Maharashtra, India.
Website: www.mitwpu.edu.in

B. Tech. Trimester VIII MINI PROJECT REPORT

On

CIRCLE RADII DETECTION

Submitted by,

Tanishq Todkar (PA08)

Vedika Shinde (PA14)

Tenzing Monkyi (PA22)

Project Guide:

Prof. Rupali Kute

Year: 2021 - 2022

School of Electronics and Communication Engineering

MIT World Peace University, Pune

MIT World Peace University, Pune

School of Electronics and Communication Engineering

Project Report Contents

Title

Circle Radii Detection

Objectives

The main objective is to detect a circle and its radius from any given image using MATLAB.

Introduction

Circle detection plays an important role in the area of configuration recognition. According to the image of the circle or arc, we calculate the characteristic parameter such as the circle center's coordinate and radius. It is a very important task in the area of computer vision and possesses extensive application foreground especially in the area of automation detection and robot assembly . Hough transform is the common method of detecting the circle objects, because it has the robustness to deal with the noises in the image. The essential of Hough transform is clustering the correlativity image elements in the image space. Then it searches the accumulative corresponding points which can contact the image elements in some analytical ways in the parameters. This transform has a perfect effect in the parameter space of one or two dimensions. But once the space of the parameter increases, the capacity of calculation will increase sharply, it will also take a tremendous amount of memory space and the time is increasing. This method makes use of the classical method, which will take the contour of the circle by clustering the parameters in the three dimensional space . Its basic idea is according to the circle equation establishing mapping from image plan to three dimensions parameter space . Get the circle center and radius on the basis of the histogram statistic result of the parameter space. The great peculiarity of this method is high reliability and it can also get the ideal result in the state of noises, distortion and missing parts of the image. Complicated calculation and great resource requirements are the shortcomings of this method. When we deal with complicated background images with interferential information, it will cause a mass of useless accumulation and reduces the capability of the algorithm greatly. At the same time when we calculate the objects with a simple circle, such as this paper listed the cylinder, gasket or bearing. Compared with the new method, it is more complicated and the resources requirement increased. In order to overcome the above shortcoming, this paper put forward an algorithm with simple calculation, high speed and not based on Hough transform, which can detect the center coordinate and radius of a circle.

Methodology

Hough Transform Algorithm:

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes, in our case it's a circle.

The HT implementation defines a mapping from the image points into an accumulator space (Hough space). The mapping is achieved in a computationally efficient manner, based on the function that describes the target shape.The Hough transform is a popular feature extraction technique that converts an image from Cartesian to polar coordinates.

We will be referring to this algorithm to find the centers and radii of the detected circle.

There are three essential steps common to all CHTs. First, a CHT contains an accumulator array computation of high gradient foreground pixels, which is chosen as candidate pixels. Then they are collected "votes" in the accumulator array. Center estimation is the second step. It predicts the circles centers by detecting the peaks in the accumulator array that produced through voting on candidate pixels. The votes are accumulated in the accumulator array box according to a circle's center. Figure 1 shows an example of the candidate pixels (solid dots) falling on an actual circle (solid circle), as well as their voting patterns (dashed circles), which coincide with the center of the substantial circle. The third step in a CHT is radius estimation; if the same accumulator array has been used for more than one radius value, as is commonly done in CHT algorithms, the radii of detected circles must be estimated as a separate step.

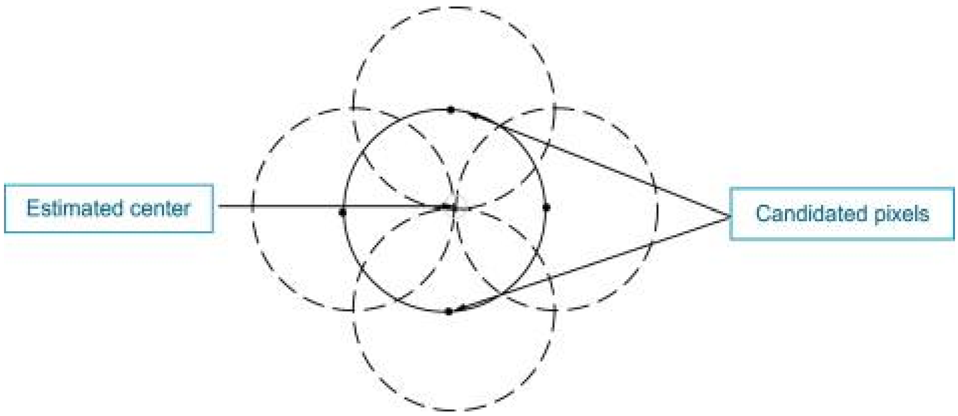


Figure 1

Finding the center and the radius (Theoretically)

Once a circle is detected, the following calculation is done to center and radius.

Assuming the detected circle in figure 2, here we make the chords P1P2 and P1P3 and then bisect both the chords and the point they both meet is the center of the circle O.

We have the coordinates of P1 as (x1,y1),P2 as (x2,y2),P3 as (x3,y3). Similarly we will

consider the coordinates of P4 as (x4,y4) and P5 as (x5,y5).

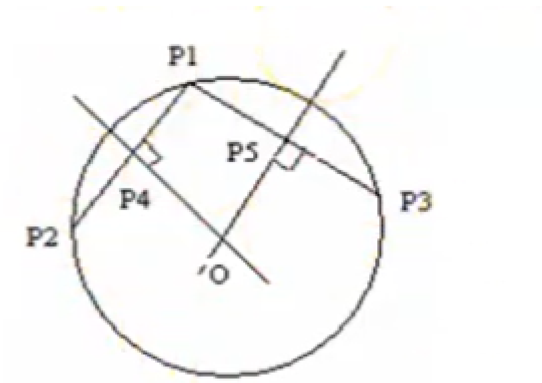


Figure 2

We then find the coordinates of point P4 and P5 from the mentioned formula in figure 3.

The line equation of the both chords and their respective chords are mentioned.As the two perpendicular lines that bisects the chords at point P4 and P5, the slope of these lines is given by the formula m1*m2 = -1 (m1 = here, slope of chord and m2= here,slope of perpendicular bisector).

From all the formulas mentioned we can find the center of the circle and radius.

$$P_4 \begin{cases} x_4 = x_1 + \dfrac{x_2 - x_1}{2} \\ y_4 = y_1 + \dfrac{y_2 - y_1}{2} \end{cases}$$

$$P_5 \begin{cases} x_5 = x_1 + \dfrac{x_3 - x_1}{2} \\ y_5 = y_1 + \dfrac{y_3 - y_1}{2} \end{cases}$$

The line of the chord P_1P_2 is: $y = K_1x + b_1$

Here into: $K_1 = \dfrac{y_2 - y_1}{x_2 - x_1}$, $b_1 = y_1 - K_1x_1$

The line of the chord P_1P_3 is: $y = K_2x + b_2$

Here into: $K_2 = \dfrac{y_3 - y_1}{x_3 - x_1}$, $b_2 = y_1 - K_2x_1$

The perpendicular bisectors of the two chords are:

$$y_{11} = K_{11}x + b_{11}$$

$$y_{22} = K_{22}x + b_{22}$$

Here into: $K_{11} = \dfrac{x_1 - x_2}{y_2 - y_1}$, $b_{11} = y_4 - K_{11}x_4$

$K_{22} = \dfrac{x_1 - x_3}{y_3 - y_1}$, $b_{22} = y_5 - K_{22}x_5$

So, according to the above formulas we can calculate the center's coordinate of $C(x_0, y_0)$

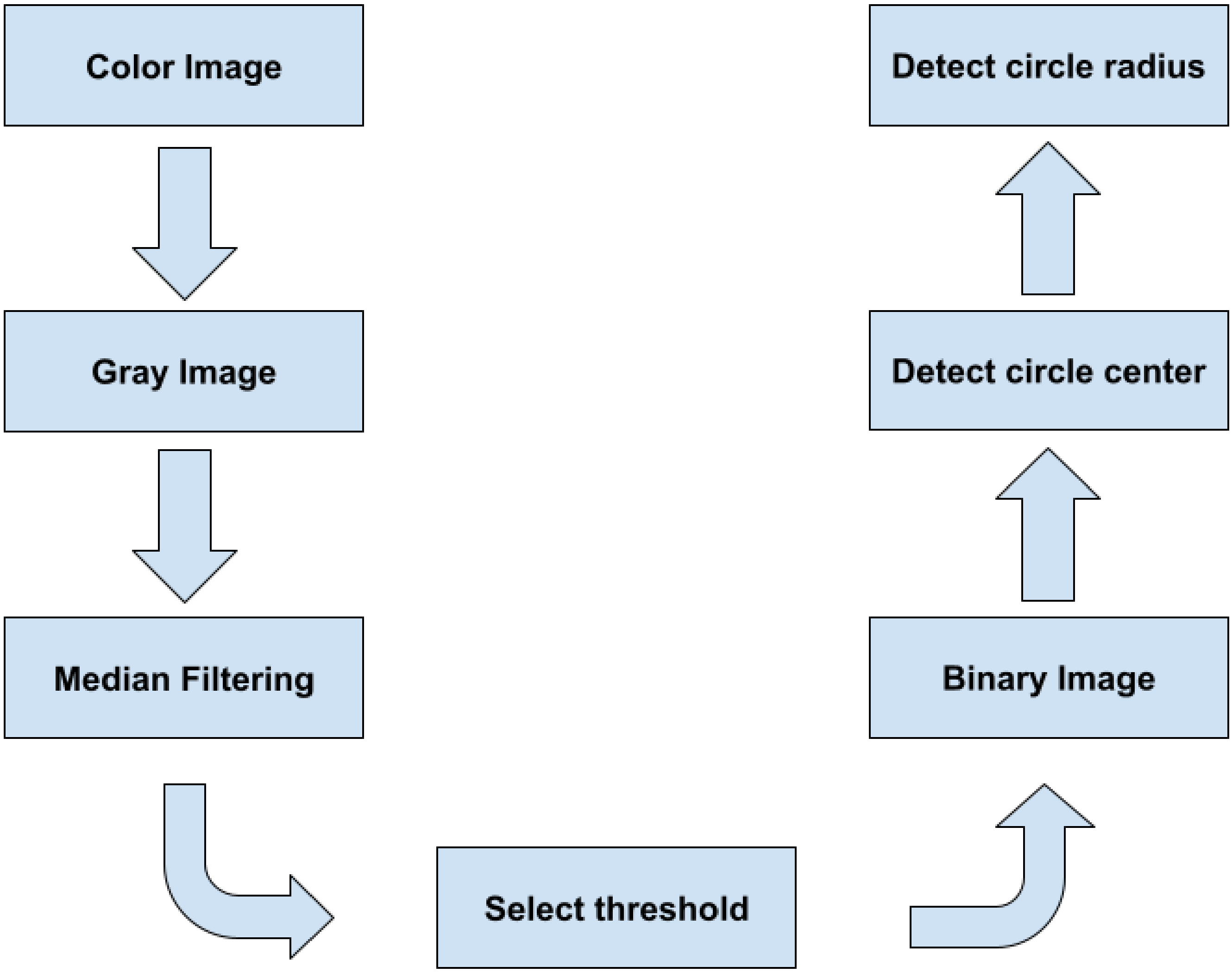
Here into: $x_0 = \dfrac{b_{22} - b_{11}}{K_{11} - K_{22}}$, $y_0 = K_{11}x + b_{11}$

And the radius is:

$$r = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

Figure 3

BLOCK DIAGRAM



Algorithm

1. Load the image and save the file in the variable
2. Display the image (colored) as figure1.
3. Use the *imdistline* tool to find the approximate estimate of the radii of various circles (which will be used later in the code).
4. Delete/ remove the *imdistline* tool since it has served its purpose.
5. Convert the colored image to grayscale image by using the *rgb2gray* function.
6. Display the grayscale image as figure2.
7. The background in the grayscale image is brighter than the chips (dark circles) in the image, but *imfindcircles* finds circular objects that are brighter than the background by default.
8. Hence we set '*objectpolarity*' to '*dark*' in *imfindcircles* to search for dark circles.

9. We use the previously approx radii values found in step 3 and set the radius value in imfindcircles as shown below.

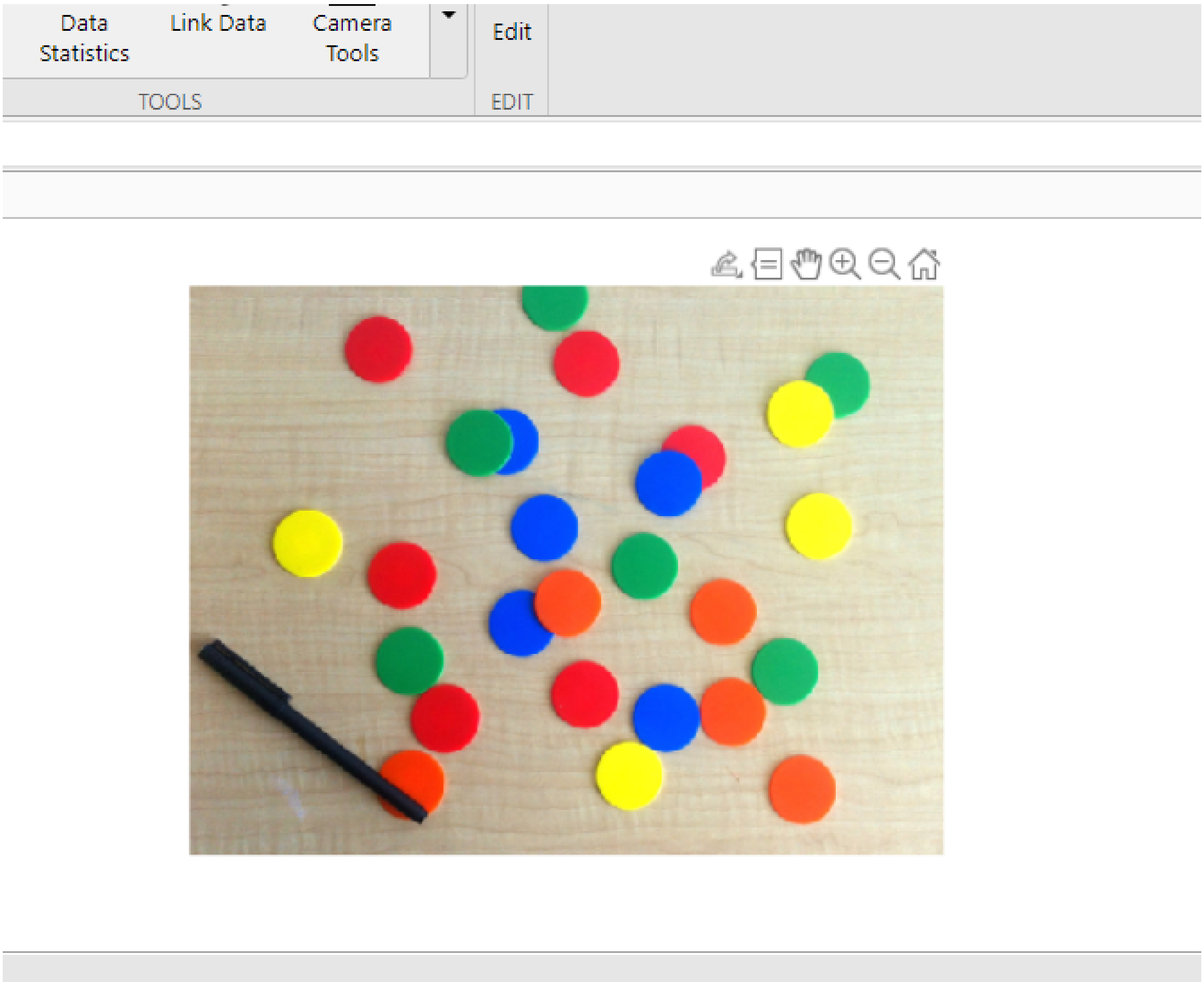
   ```
   [centers,radii] = imfindcircles(rgb,[20 25], 'ObjectPolarity','dark')
   ```

10. Increased the detection sensitivity parameter to 0.92 after trials and errors .
11. Then we draw circles on the grayscale image using the '*viscircles*' function which uses the `[centers,radii]` that we found above.
12. Delete the previously drawn circles on grayscale image.
13. Display the original image with the drawn circles as figure3 using the '*viscircles*' function again.

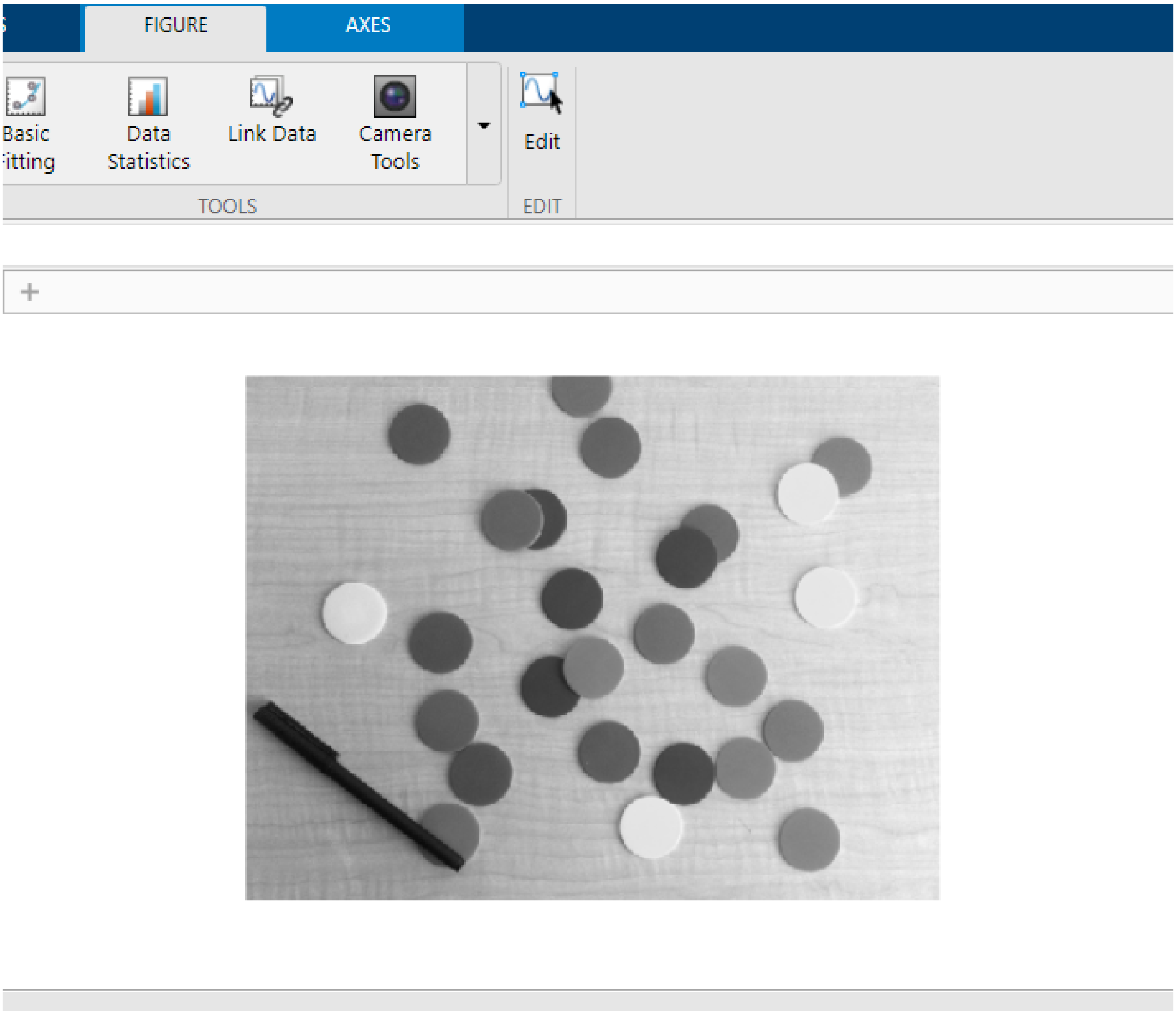
Program

```
clc;
clear all;
rgb = imread('coloredchips.png');
figure(1)
imshow(rgb)
d=imdistline;
delete(d)
gray_image = rgb2gray(rgb);
figure(2)
imshow(gray_image)
[centers,radii] = imfindcircles(rgb,[20 25], 'ObjectPolarity','dark')
[centers,radii]=imfindcircles(rgb,[20
25],'ObjectPolarity','dark','Sensitivity',0.92)
h=viscircles(centers,radii);
[centers,radii]=imfindcircles(rgb,[20
25],'ObjectPolarity','dark','Sensitivity',0.92);
delete(h)
figure(3)
imshow(rgb)
h=viscircles(centers,radii);
```

Results:

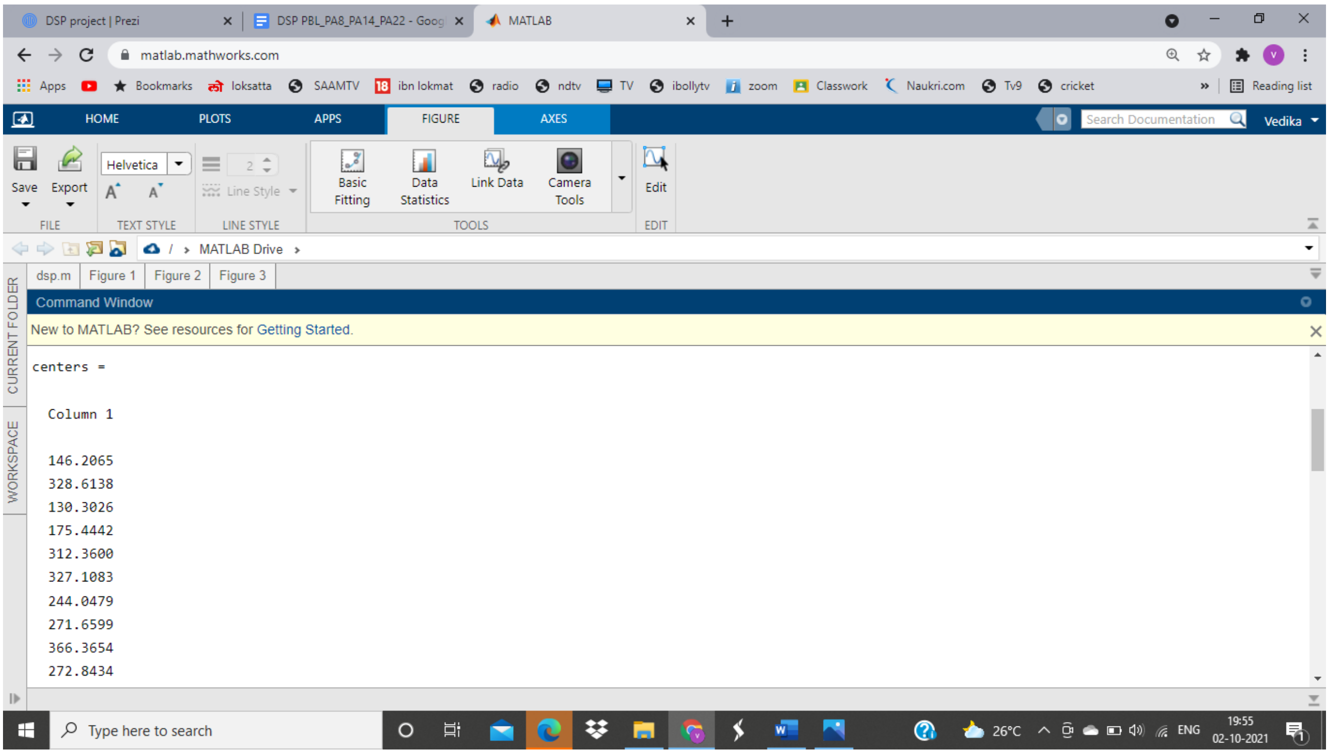


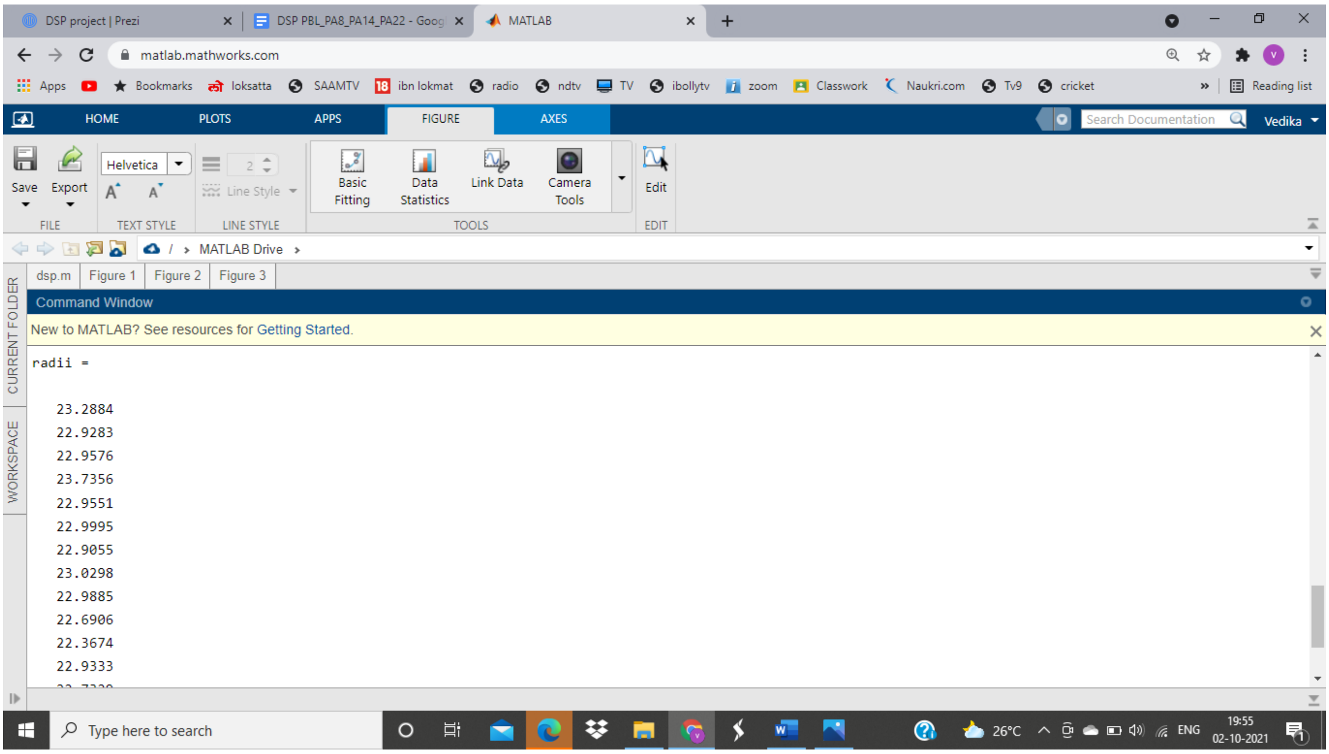
Original Image



Gray Scale Image

Detected Circles





Conclusion:

In this project, we were successfully able to detect the circles, center of the circle and radius of the detected circles from a given image.

References-

"A new method of circle's center and radius detection in image processing" Authors: Zhang Mingzhu, Cao Huanrong. Published by: IEEE

https://en.wikipedia.org

https://homepages.inf.ed.ac.uk