MCT-453 MACHINE VISION

SEMESTER PROJECT-I

BLOB DETECTION

SUBMITTED BY : 2019-MC-26

SUBMITTED TO: Sir Rzi Abbas

DEPARTMENT OF MECHATRONICS AND CONTROL ENGINEERING

University of Engineering and Technology, Lahore

Contents

MACHINE VISION

INTRODUCTION

Machine vision allows you to obtain useful information about physical objects by automating analysis of digital images of those objects. This is one of the most challenging applications of computer technology. There are two general reasons for this: almost all vision tasks require at least some judgment on the part of the machine, and the amount of time allotted for completing the task usually is severely limited. While computers are astonishingly good at elaborate, high-speed calculation, they still are very primitive when it comes to judgment.

The key to successful machine-vision performance is the software that runs on the computer and analyzes the images. Software is the only component that cannot be considered a commodity and often is a vendor's most important intellectual property. In recent years, it has become more common to deliver these five components in the form of a single, integrated package. These systems often are referred to as machine-vision sensors to distinguish them from more traditional systems where each of the components is a discrete module. While the traditional systems are somewhat more versatile, the sensors generally are less expensive and easier to use.

SOFTWARE

- Matlab 2021a

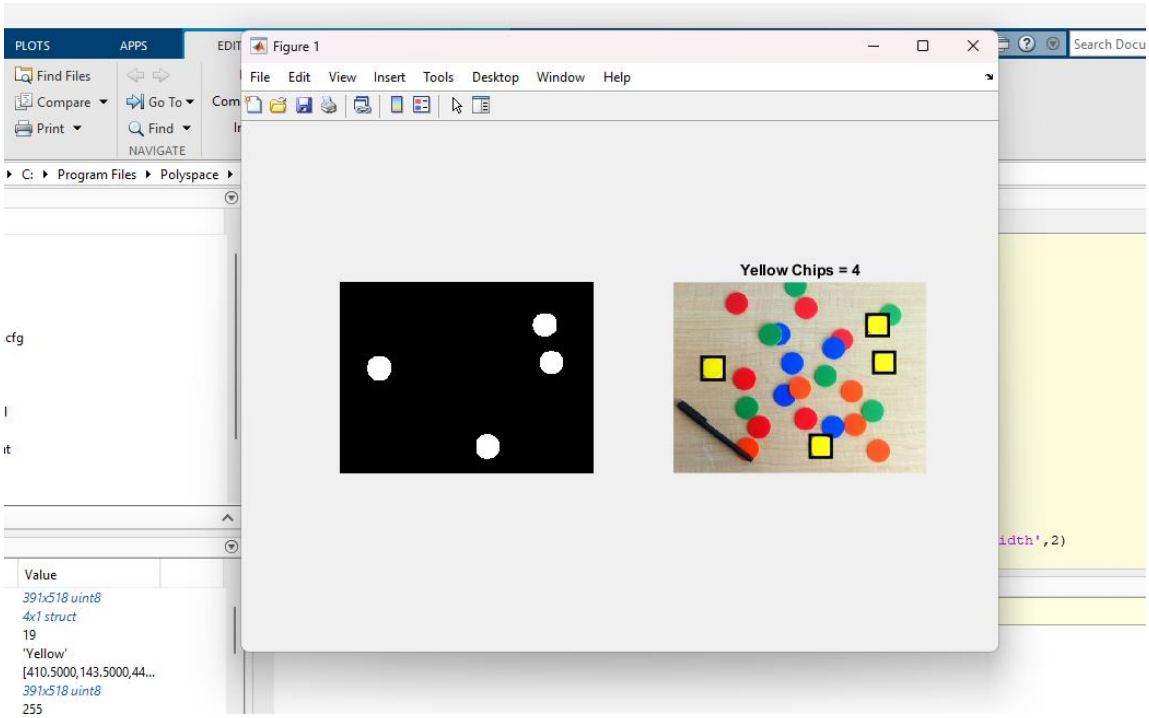
METHODOLOGY

First, I Read image 'coloredChips.png' using the command imread. Then get the coordinates using ginput (c) command, where 'c' argument defines the number of clicks on the image as in our case we only needed one click so, c is 1 in the code. Converted the floating value into integer using floor command then get the rgb value from the pixel coordinated and stored it in redV etc. Then separated all the RGB layers. Then generated three lookup tables for all Red, Green and Blue of zeros so we can convert the image that we demand.

Now here we know the values of pixel we get from our image, of course the click has been done, so we have the desired pixel rgb values, now form all the possible conditions for red,green,blue,yellow,orange and black. For example in case of red disk ,we clicked red, we got values I redV variable we defined earlier, now check from the if condition if the value obtained falls in that particular region update all the lookup tables particularly.

Now as we know the data type of lookup tables we created above was of double data type convert it into unsinged integer 8 ‘uint8’. Now get the image mask by applying all the lookup tables on each plane, then convolve the separated planes. Apply the mask to the original image to get the RGB image. Covert in into grayscale image using ‘rgb2gray() ’ command convert it into logical datatype.Then opened the image on the basis of area then dilate the image to get the bboxes fitted properly now by using the command the ‘regionprops’ count the number of objects and show it in the plot, then final make the bounding boxes and show in the plot.

SAMPLE OUTPUTS



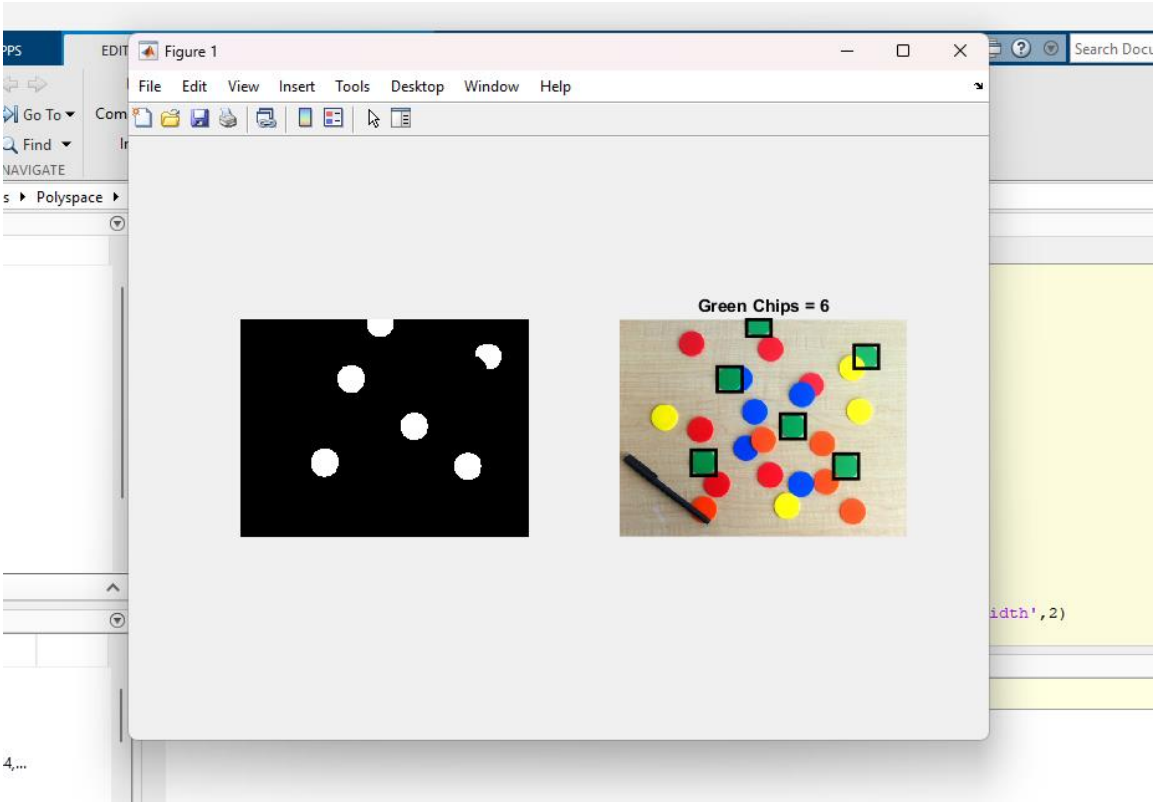


Figure 1
File Edit View Insert Tools Desktop Window Help
Green Chips = 6

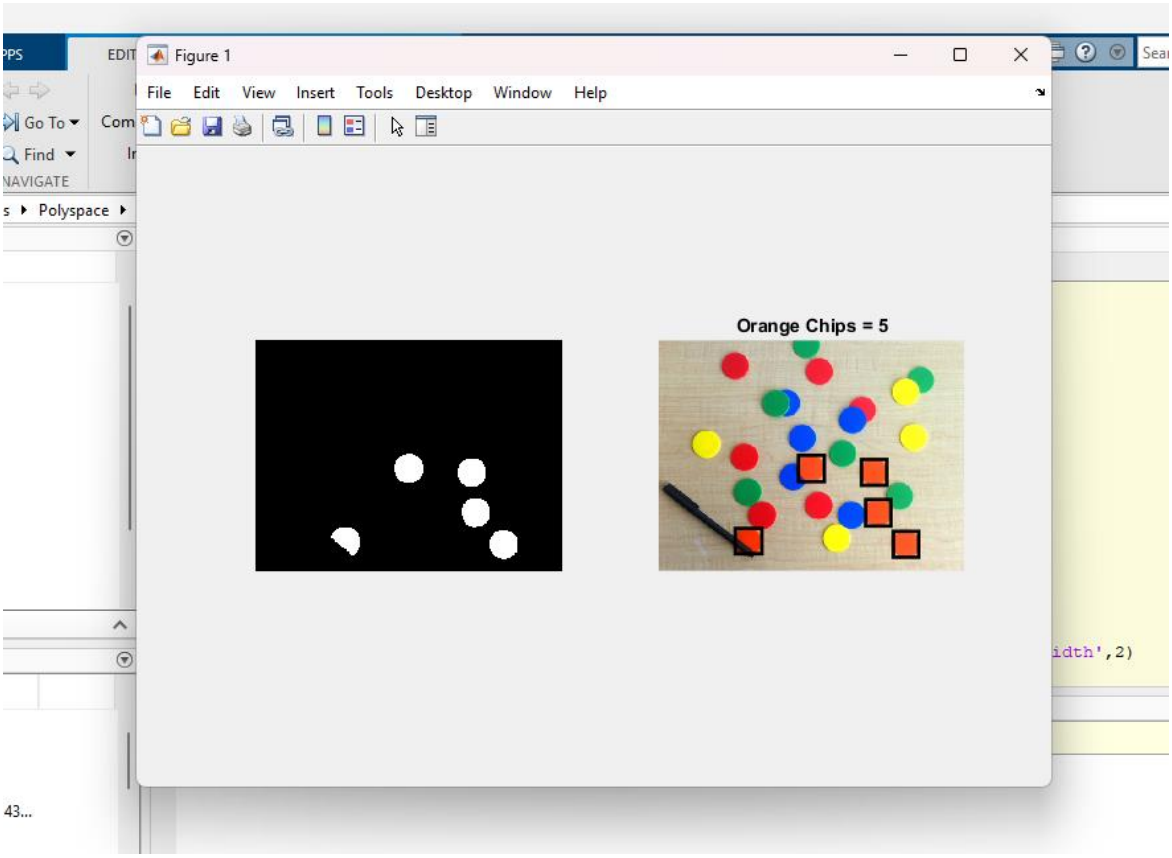
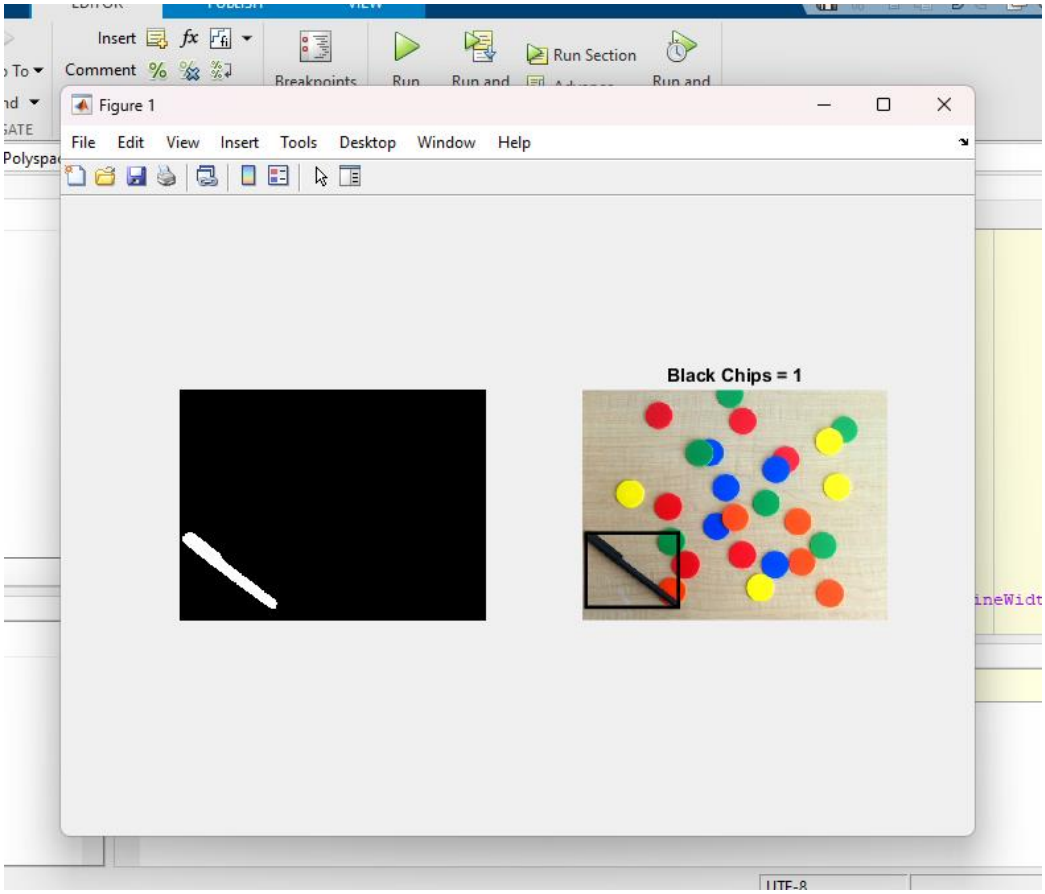


Figure 1
File Edit View Insert Tools Desktop Window Help
Orange Chips = 5



REASONS FOR ERRORS

- The orange disk under the pen cannot be obtained because the black shade under pen on orange disk makes it difficult to get and probably it considers it as a background.
- The problem of fitting the bounding boxes as it wasn't seems appropriate so I dilated the image to get the bounding boxes fitted well.

REFERENCES

- Identify axes coordinates - MATLAB ginput - MathWorks France
- How to get RGB values from x,y coordinates - MATLAB Answers - MATLAB Central (mathworks.com)
- Implement Lookup Tables in MATLAB - MATLAB & Simulink (mathworks.com)
- How to draw BoundingBoxes in Matlab (bounding boxes) - YouTube