ACTIVITY 5

Image Segmentation

MANALANG, JOHNENN R.
APPLIED PHYSICS 157

TABLE OF CONTENTS

Contents

OBJECTIVES

Objectives

Objective 1

Familiarize with image segmentation techniques: thresholding, parametric, and non-parametric.

Objective 2

Compare the results of the parametric and non-parametric segmentation in different images.

Objective 3

Provide several applications of image segmentation in different fields.

INTRODUCTION

Automating feature extraction requires three steps: image segmentation, morphological operations, and blob analysis. In this activity, we focus on image segmentation which consists of dividing an image into segments to be able to work with only the desired segments. We look at thresholding, parametric, and nonparametric segmentation and see which of these are better to use for different types of images.

01

THRESHOLDING

Thresholding

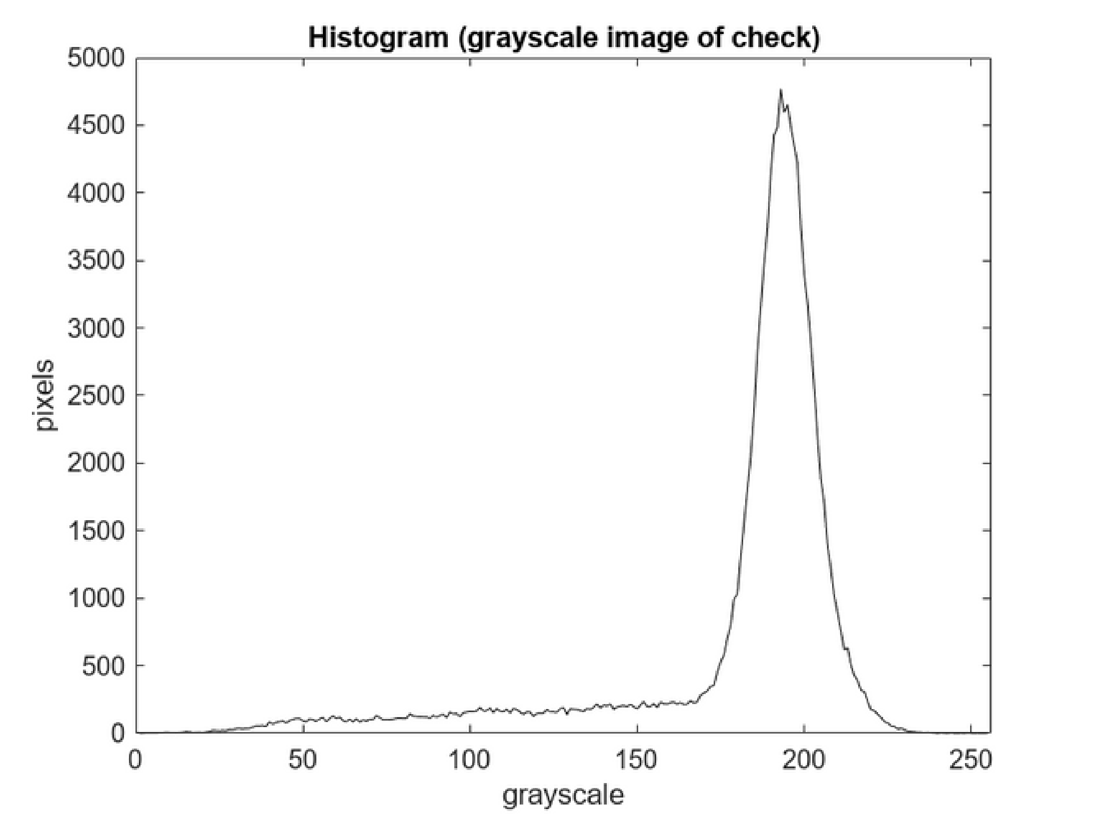
Segmentation can be done through thresholding for grayscale images if the region of interest (ROI) has a distinct grayscale range from the background.

EXERCISE: CHECK IMAGE

Here is a grayscale image of a check. Suppose we only want to get the handwritten text. We can do this by thresholding. But before that, we need to set the threshold values. To find out which values to use, let's take a look at the histogram of this image.

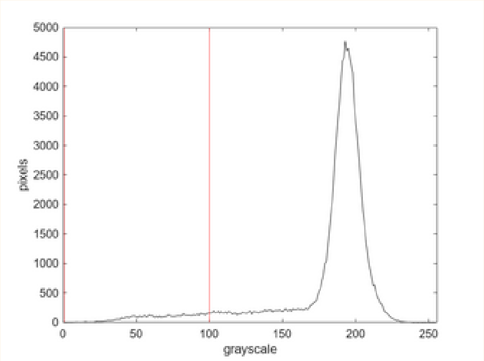
HISTOGRAM

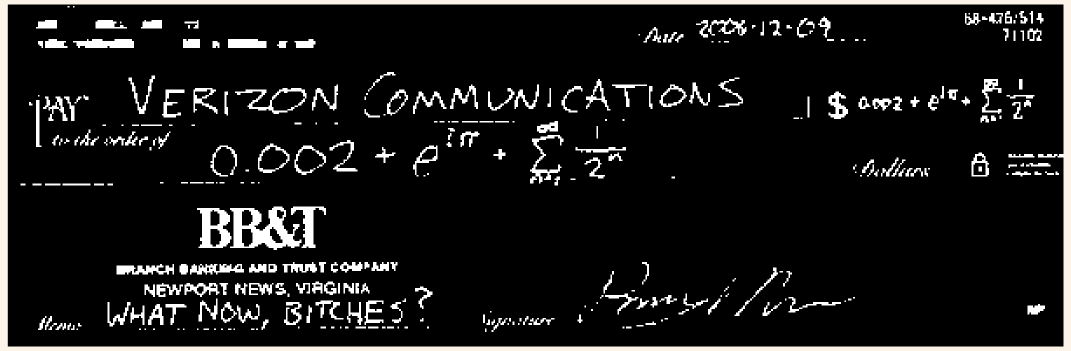
The histogram of the image tells us the number of pixels corresponding to the background or the text. Since the background of the image is larger compared to the text, then the largest peak corresponds to the background pixels. We can adjust the threshold to see which values best picks out the text from the background.



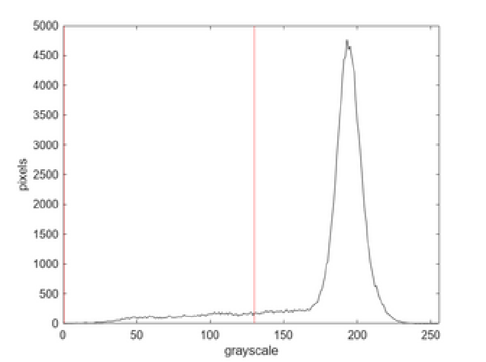
THRESHOLDING

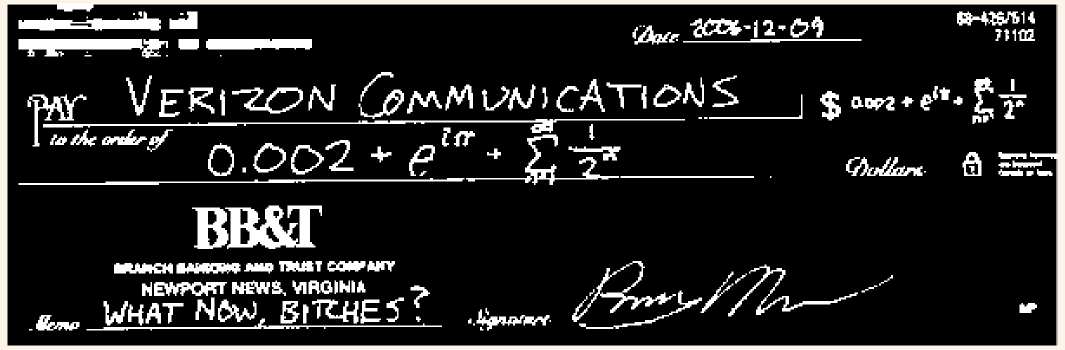
0–100



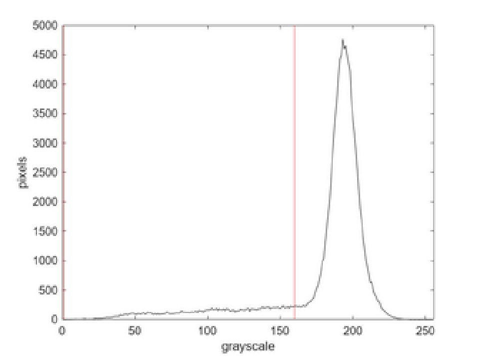


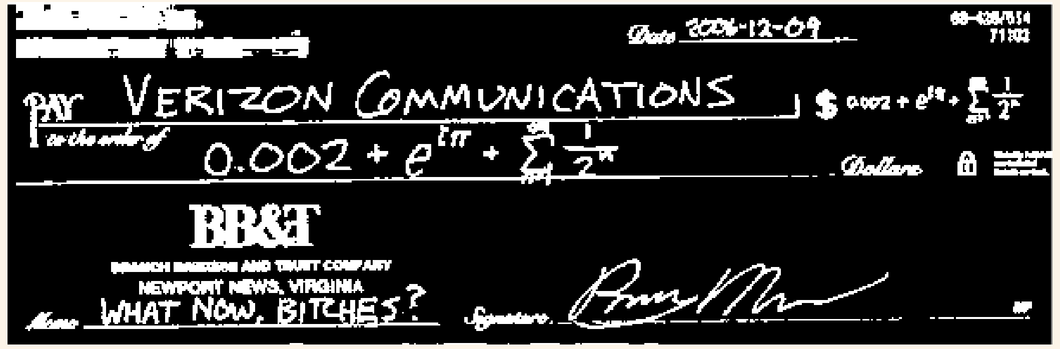
0–130





0–160

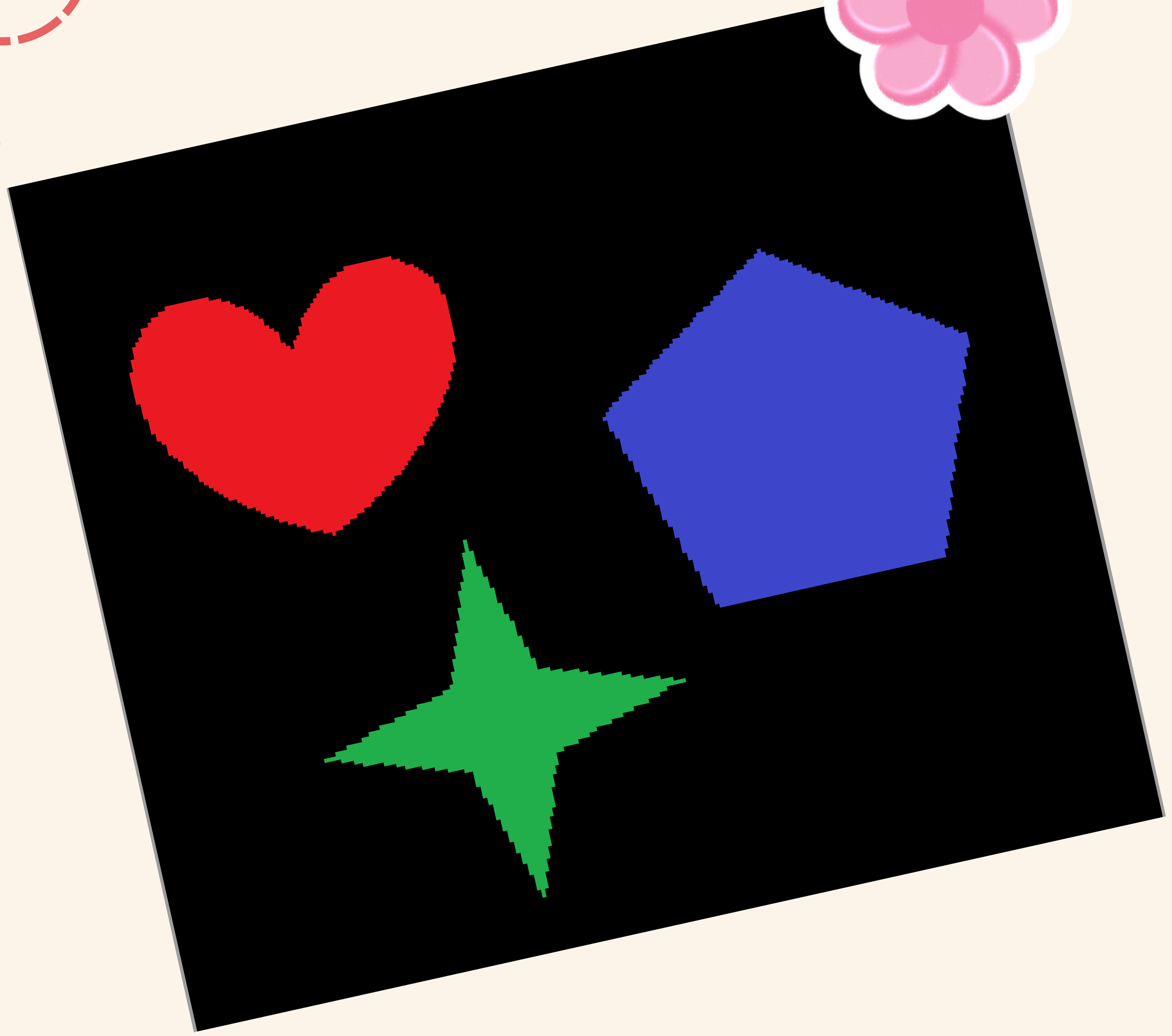




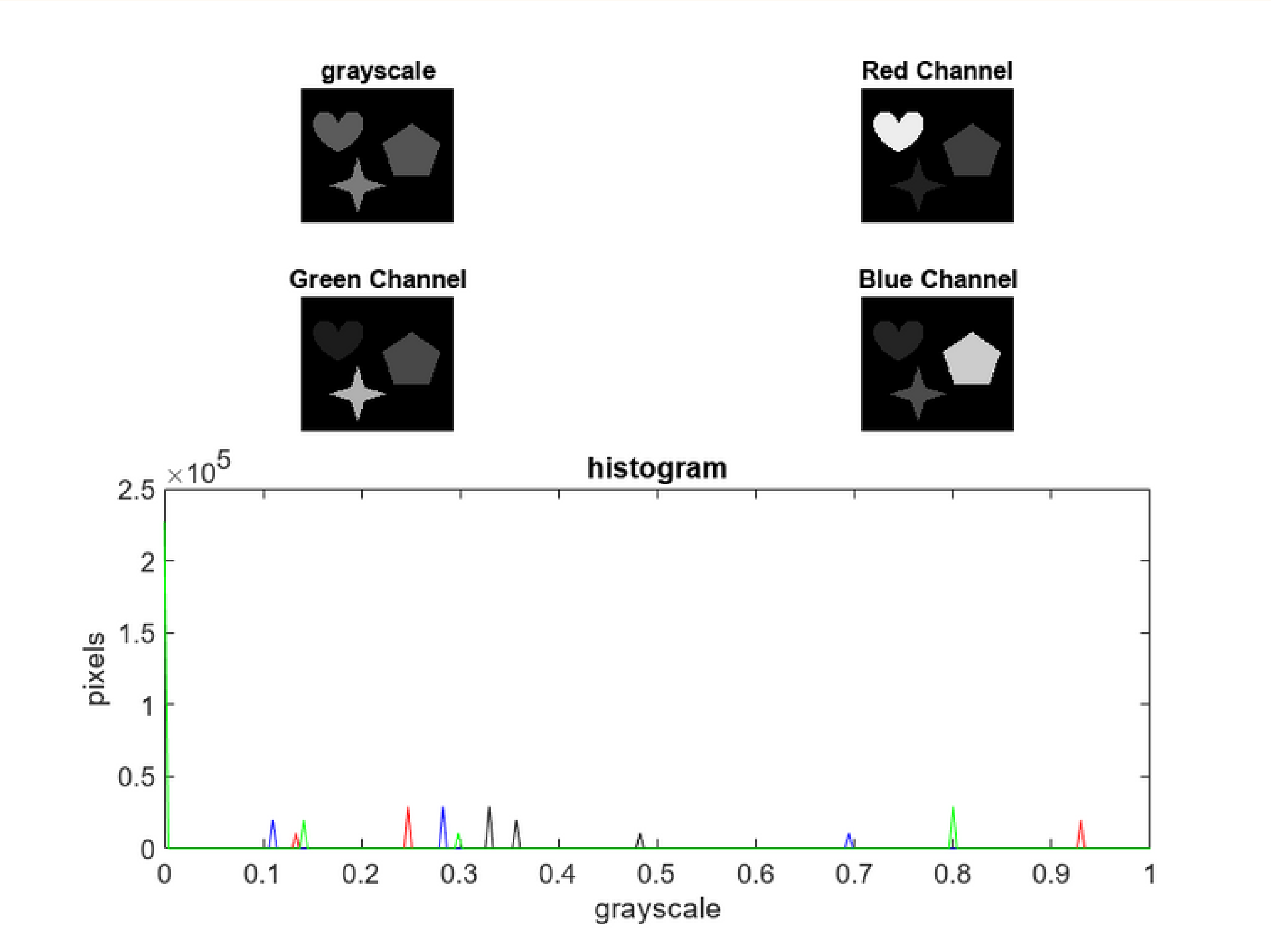
I used different threshold values here. Notice that the text is white and the background is black. This is because the field of interest for feature extraction in MATLAB should be in white. Since the largest peak of the histogram corresponds to the background, therefore, we need to set the threshold values from 0 to a value before the rising peak to get the text pixels. The best value to use is 0–160 because it contains all the text information.

THRESHOLDING FOR RGB

Thresholding can also be done in RGB images. This is when the image has distinct colors and high contrast with the background. Here, I created my own image of different shapes with different colors: red, green, and blue. Let's start by splitting the original image to its RGB channels and analyze them.

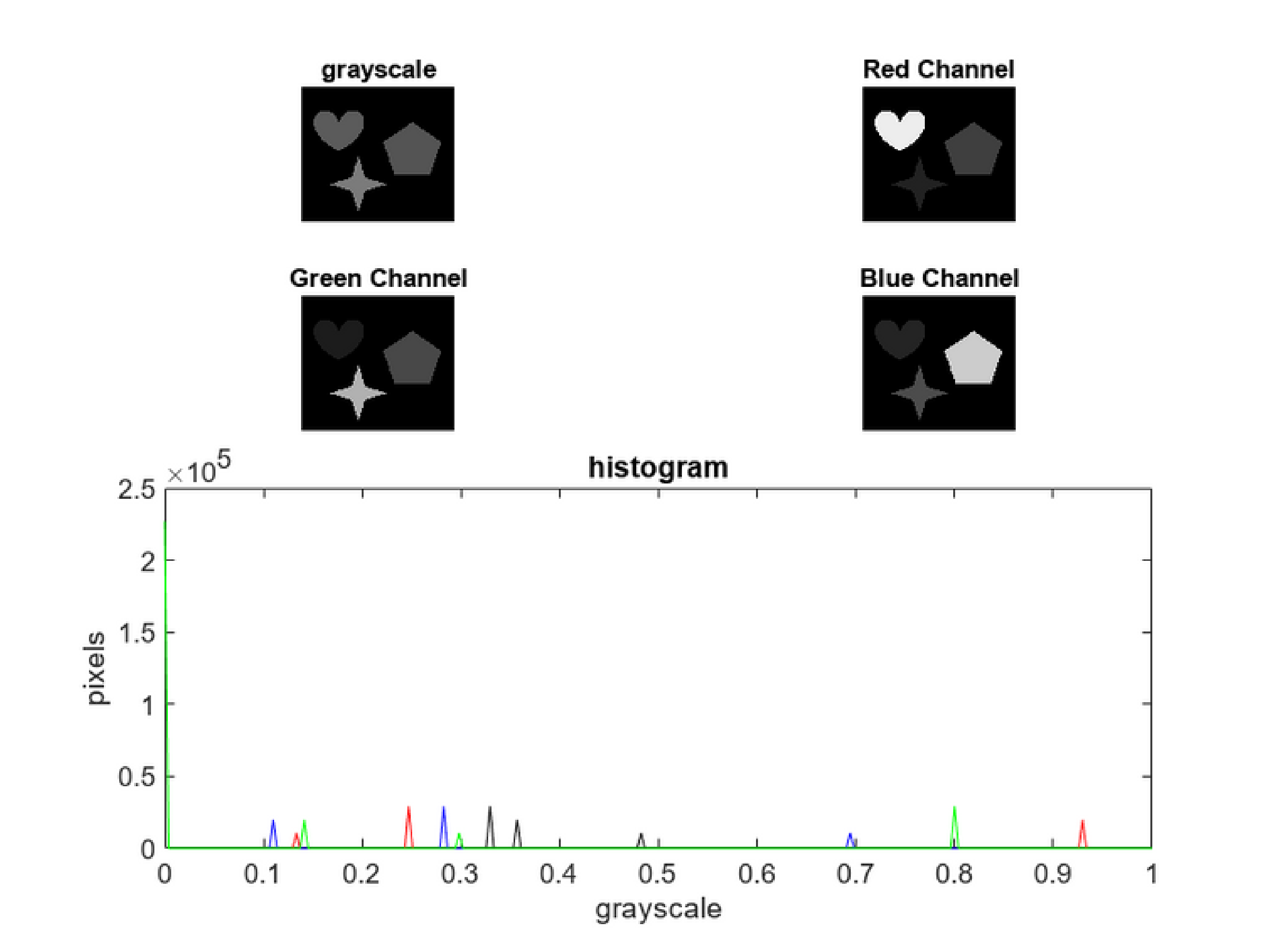


THRESHOLDING IN RGB



Here, we can see the RGB channels of the image and the corresponding histograms for the grayscale and RGB channels. The RGB channels appear to be gray. The greater the RGB value, the lighter the color. For instance, in the red channel, the lightest color is the heart since it is red. Notice that the RGB channels have more contrast than the grayscale of the image. Suppose we only want to pick out the red regions, we can use the red channel and apply thresholding. This can also be done for the other colors.

THRESHOLDING IN RGB

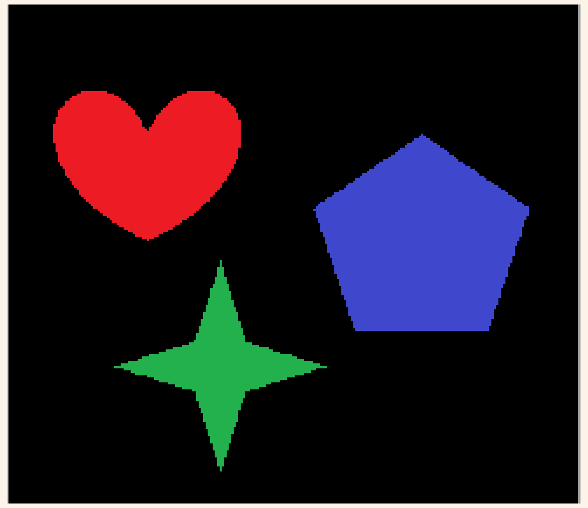


Let's analyze the histogram. Why is the histogram like this? The x axis corresponds to the intensity of the pixels and the y-axis shows the number of pixels. Recall that 0 is black and white is 1. Since the grayscale and the image only has extreme values and the background is larger than the region of interest, the number of pixels with intensity 0 is highest. So if we want to only get the region of interest (i.e. heart in R, star in G, and pentagon in B), we will set the threshold such that it filters out the background.

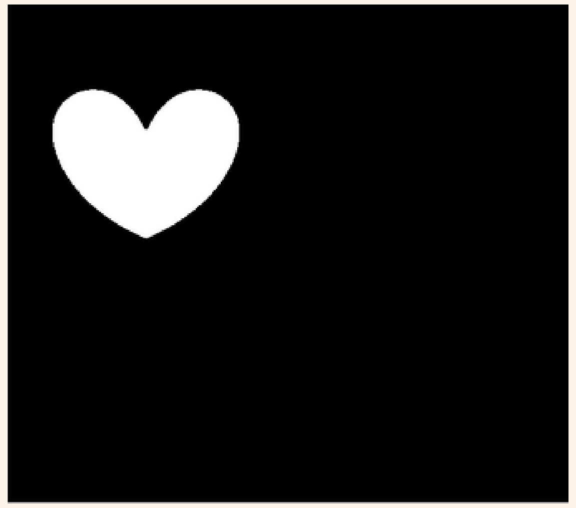
THRESHOLDING IN RGB

I created three separate functions that will only threshold the R, G, and B channels. Then, I set the intensity to 0.5 – 5 . The result is shown below.

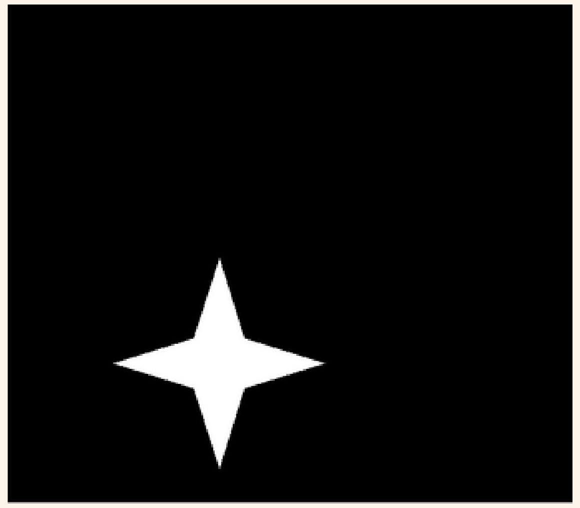
original



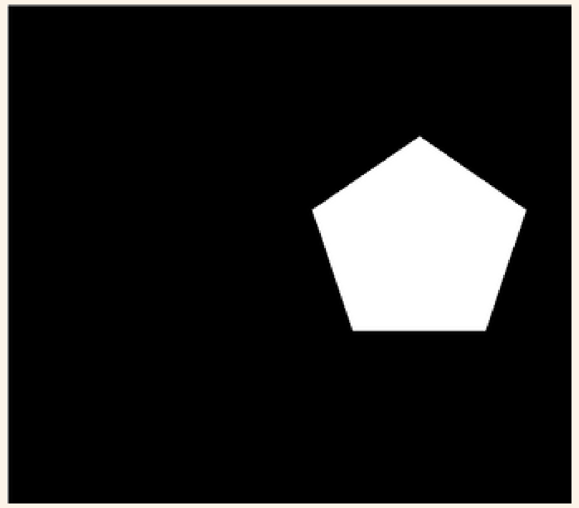
red



green



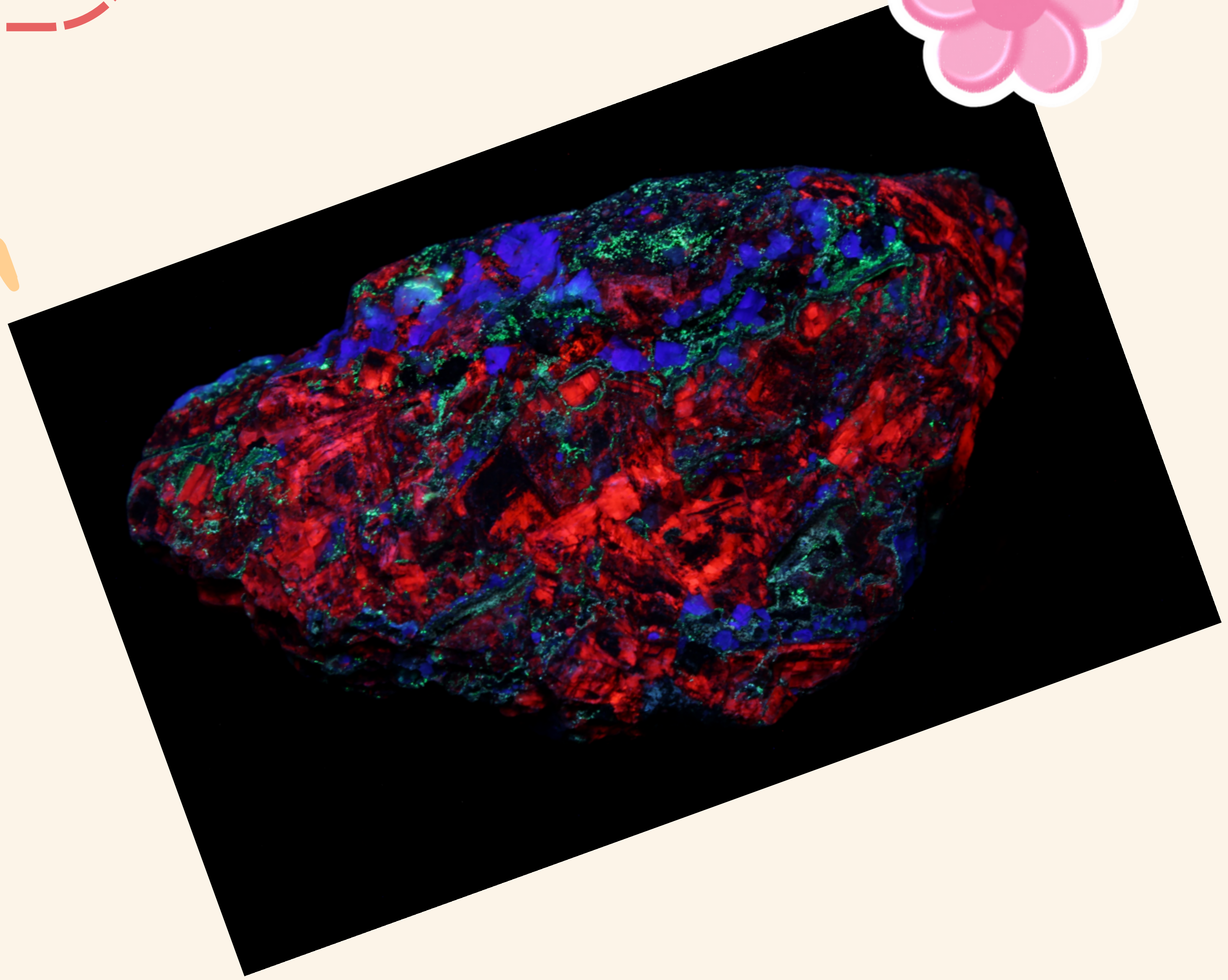
blue



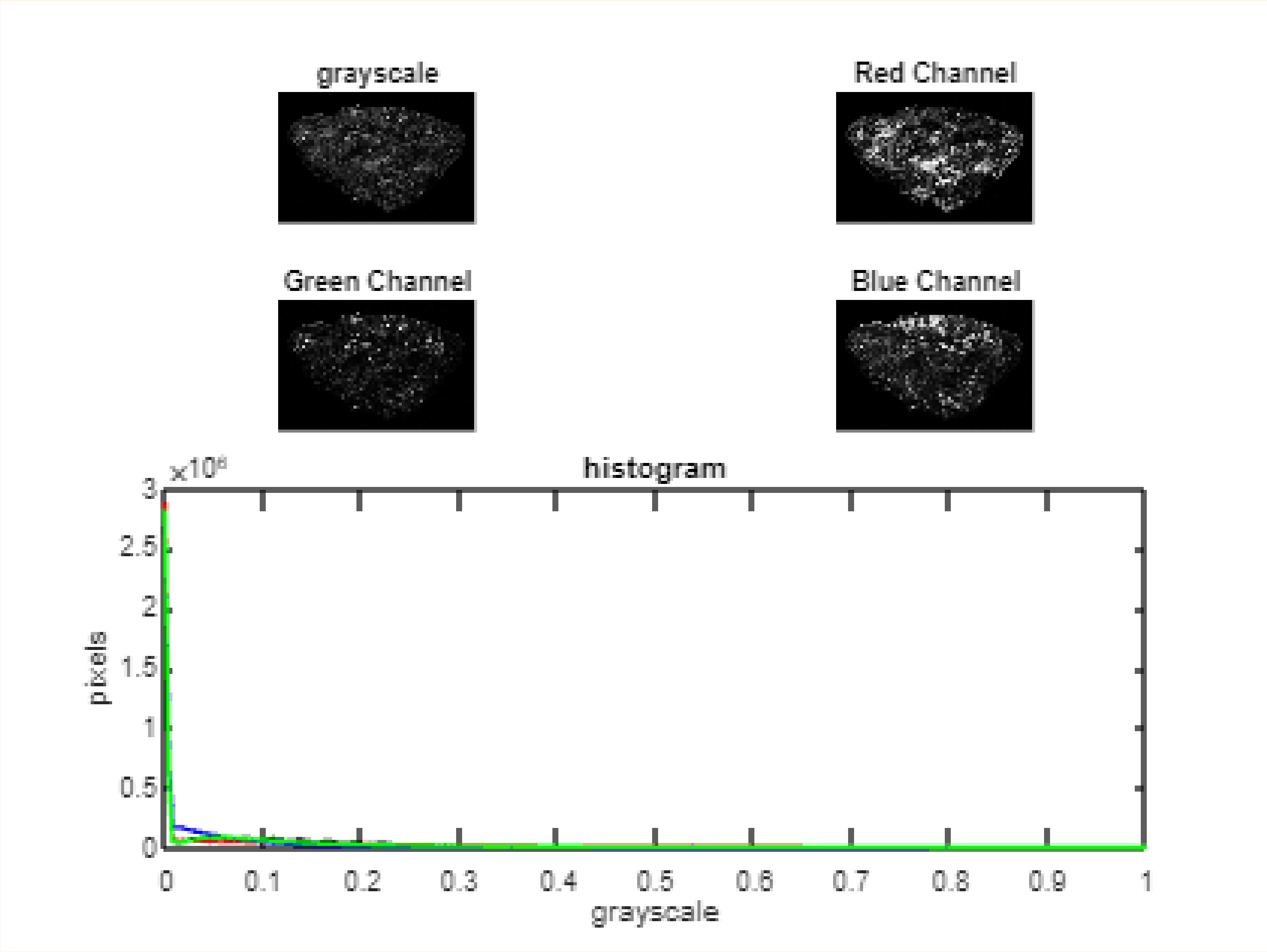
We successfully picked out only the red, the green, and the blue parts of the image by using the RGB channels. Let's try to apply this into another image.

THRESHOLDING FOR RGB

Here is an image of a cabinet-sized calcite (red), willemite(green), and fluorite (blue) from the Hull Mine, Arizona under the UV light. Let's try to pick out each of the minerals using thresholding technique.



THRESHOLDING IN RGB

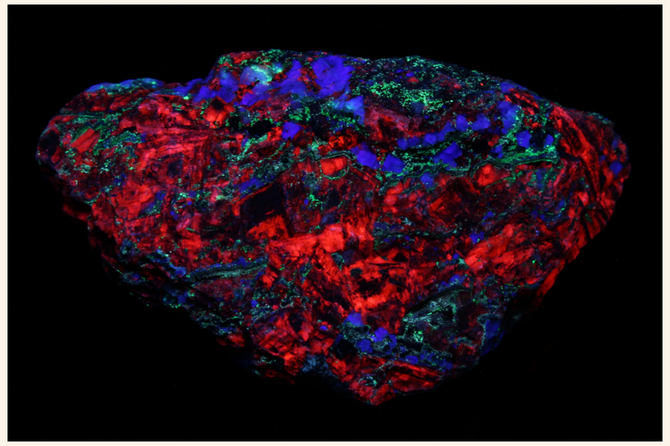


We split the image in RGB channels. Similar to the histogram of the shapes from before, we see that most of the pixels also has intensity 0 since the background is black.

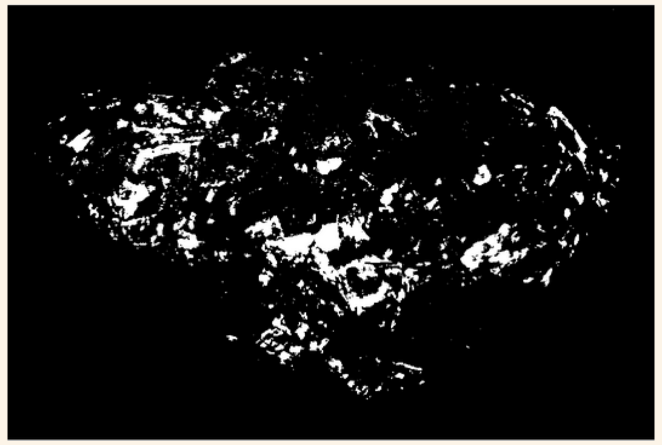
THRESHOLDING IN RGB

Using the three separate functions that I created and the thresholding values 0.5 – 7, the images below shows the result.

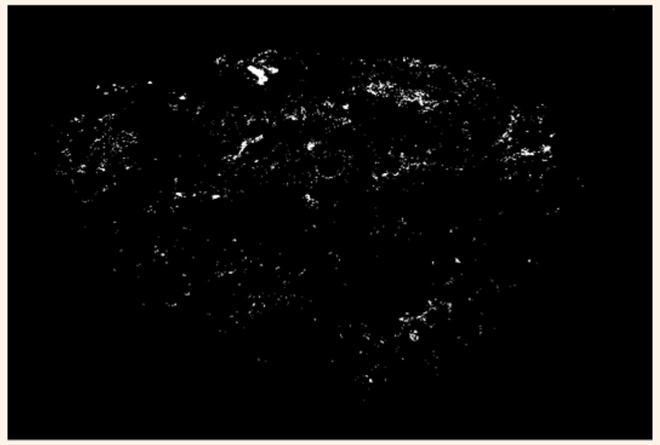
original



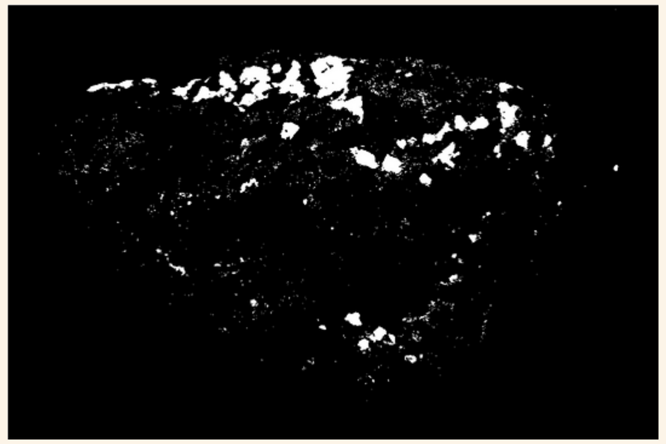
calcite (red)



willemite (green)



fluorite (blue)



Tada! We have separated the regions of the rock that contain only calcite, willemite, and fluorite. This is called mineral segmentation and is used by geologists to study mineral deposits in rocks.

02

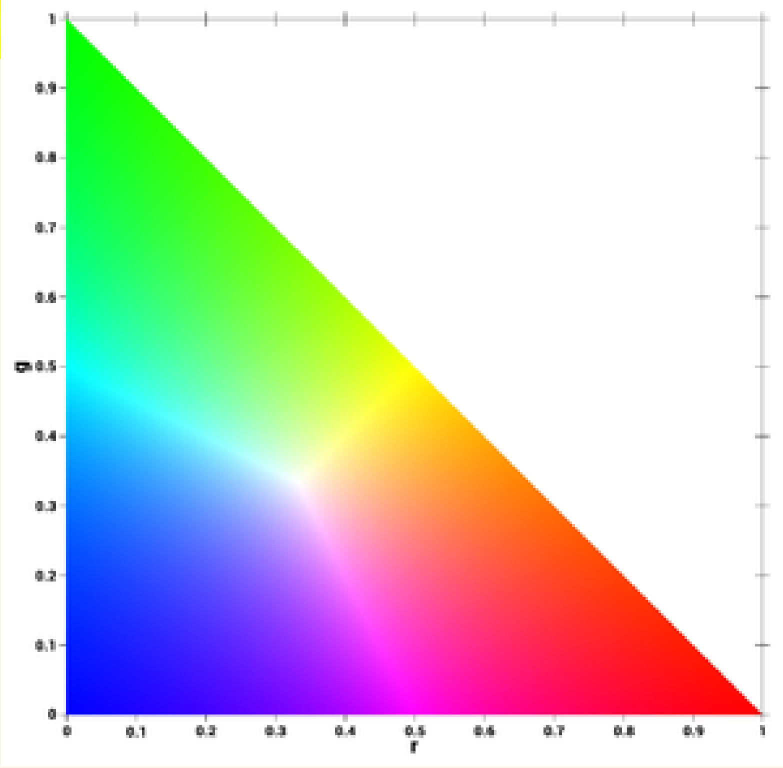
PARAMETRIC VS. NON-PARAMETRIC

Parametric and Non-parametric probability distribution estimation are two techniques also used in image segmentation. Let's compare these two techniques.

Parametric vs. Non-parametric

NORMALIZED CHROMATICITY COORDINATES

For 3D objects, thresholding is not an easy task. This is due to the shading variations (e.g. shadows seen as different brightness levels of the same color). The solution to this is to convert the image color space from RGB to the Normalized Chromaticity Coordinates (NCC). This color space separates brightness and chromaticity information. This is what we use for parametric and nonparametric segmentation.



NORMALIZED CHROMATICITY COORDINATES

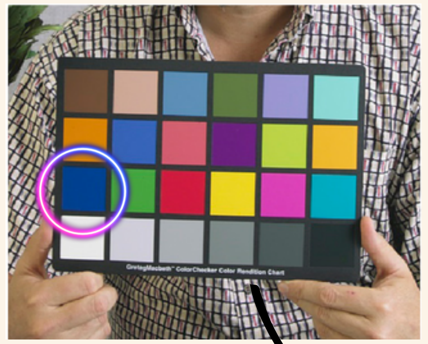
NCC coordinates

RGB channels

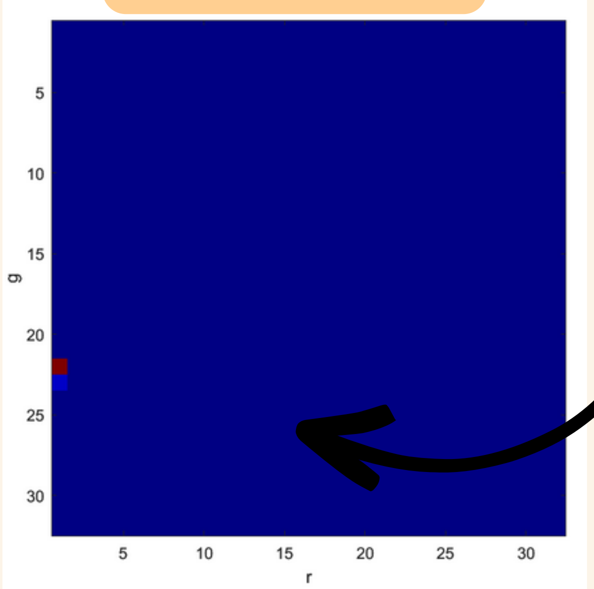
```
function [r,g] = NCC(R, G, B)
    I = R + G + B;

    I(I==0) = 1e5;

    r = R./I;
    g = G./I;
end
```

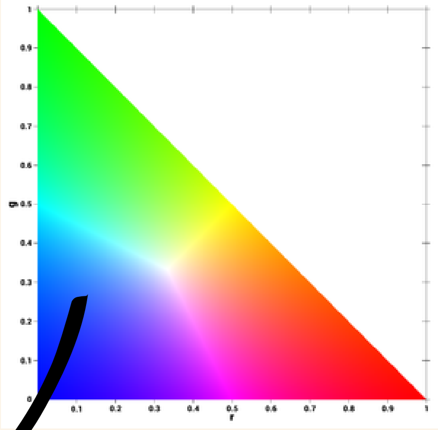
I created a function NCC with inputs of RGB channels of an image and the output are the r and g coordinates.

To validate this function, we can plot the 2D histogram of our region of interest (ROI) and see if the peaks match the location of the rg chromaticity diagram.



2D histogram





We chose the blue square as our ROI. As we can see, the 2D histogram peaked at the small blob in the blue region. Hence, the function and the 2D histogram are accurate.

COMPARISON

Parametric

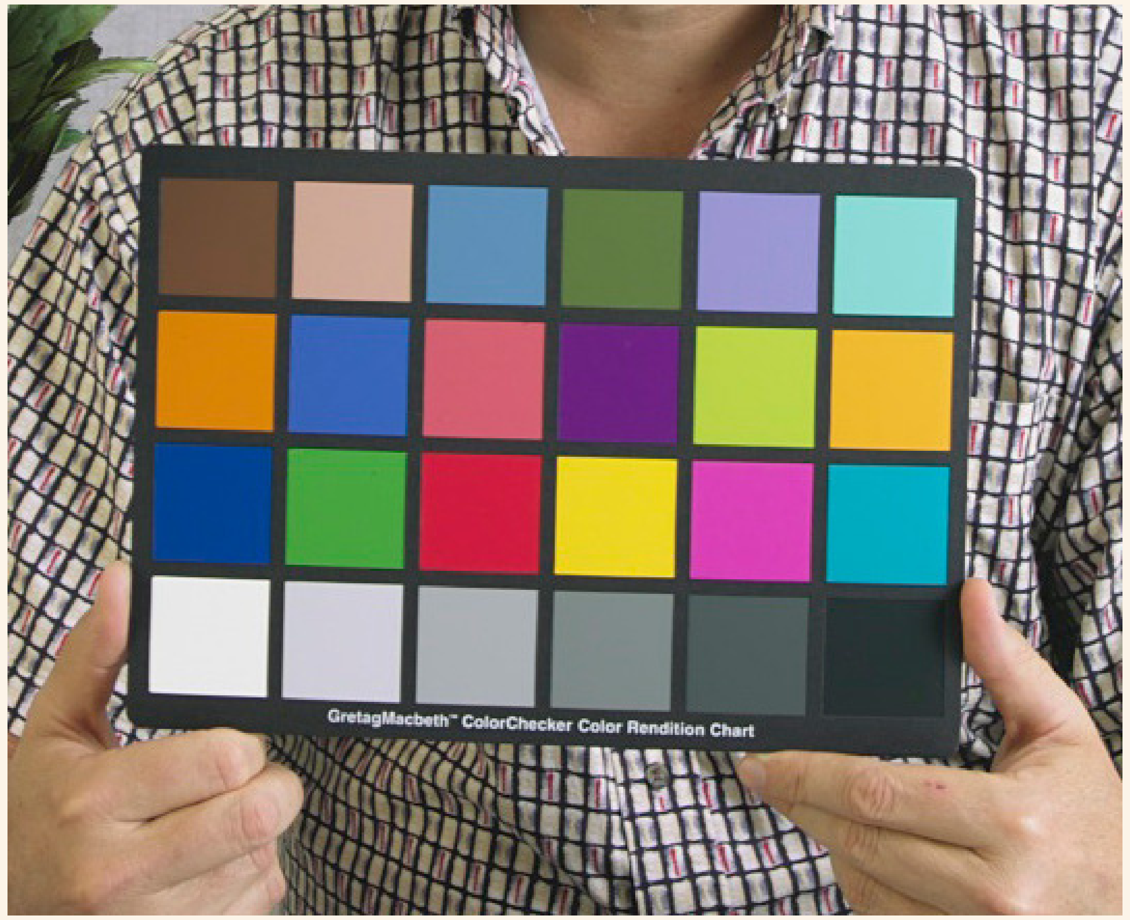
This technique consists of deriving the Gaussian probability density function (PDF) in the r and g values separately of the ROI and combining the two to get the joint PDF p(r)p(g).

Non-parametric

The 2D histogram of the ROI is obtained in this technique. It will be used to segment the image through the process of histogram backprojection.

PARAMETRIC VS. NON-PARAMETRIC

For the comparison, let's use this image of a Macbeth chart. Let's see if we will be able to use parametric and non-parametric segmentation by using the red square as our ROI. The results are shown in the next slides.

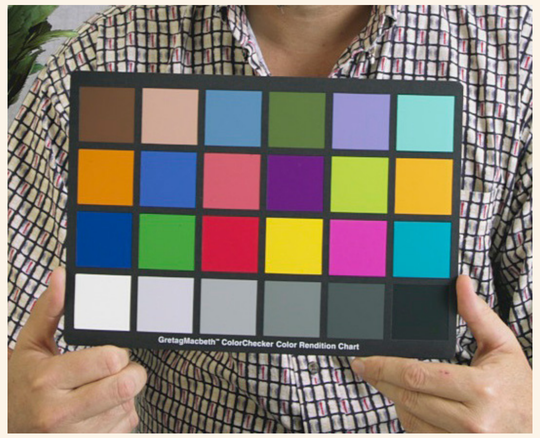


PARAMETRIC VS. NON-PARAMETRIC

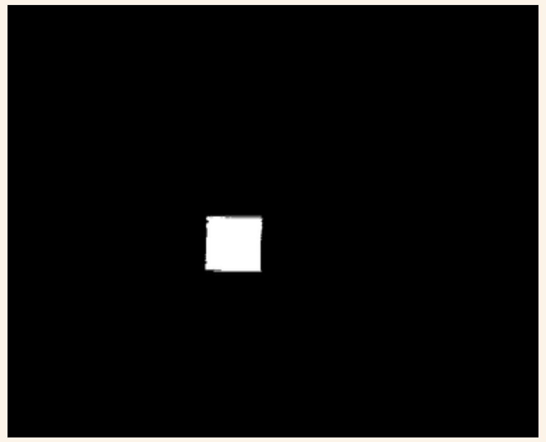
Tada! We have successfully used the parametric and non-parametric segmentation in segmenting the parts of the image that corresponds to our ROI. As we can see, not much difference can be seen in the resulting images. What if we try a different ROI in the same image?

ROI

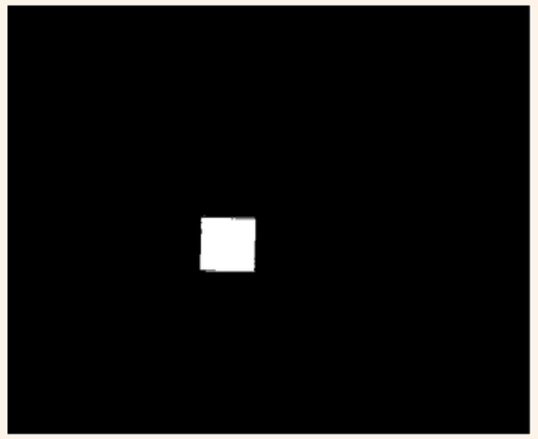
original



parametric



non-parametric

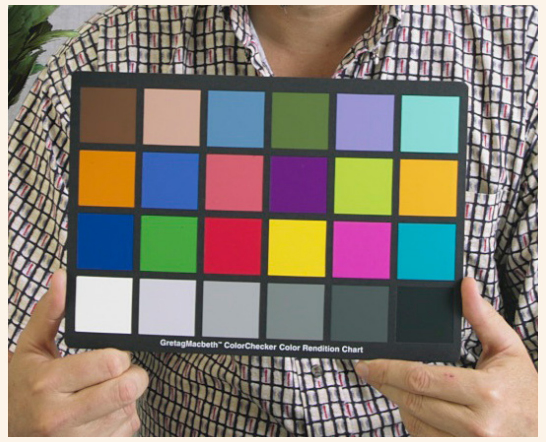


PARAMETRIC VS. NON-PARAMETRIC

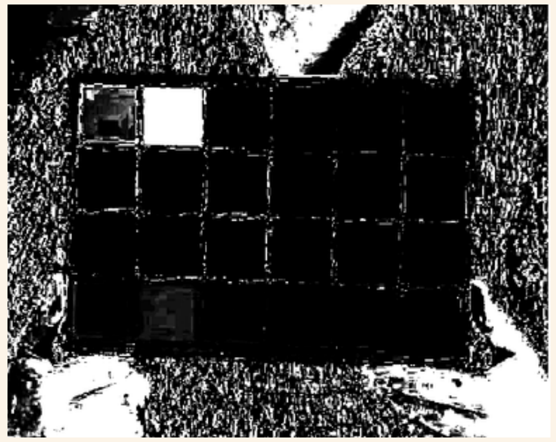
Here, I tried to use the beige color for the ROI. Since the person holding the chart has the same complexion as the color of the ROI (even the color pattern in the polo), both the techniques used included them in the segmentation. The parametric segmentation captured most of the detail, while the non-parametric lacks some of the details.

ROI

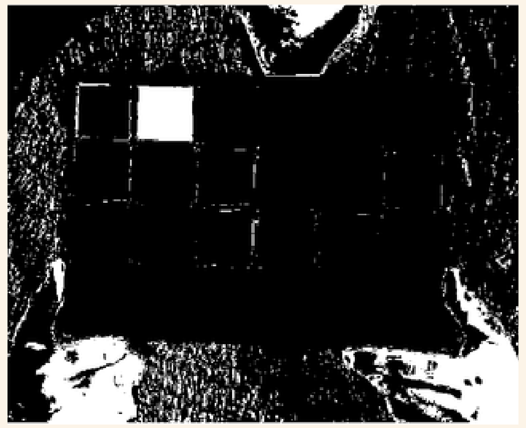
original



parametric

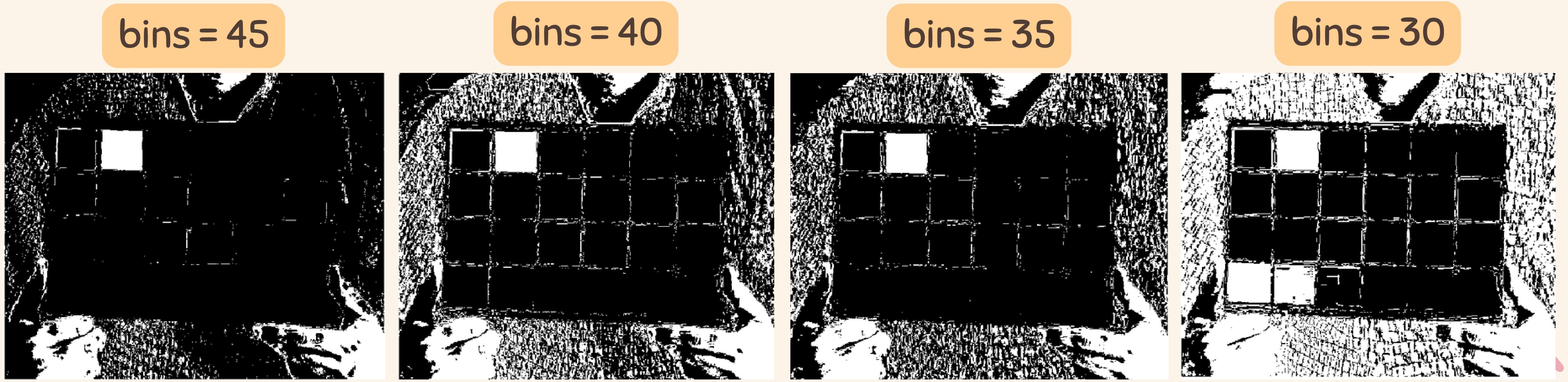


non-parametric



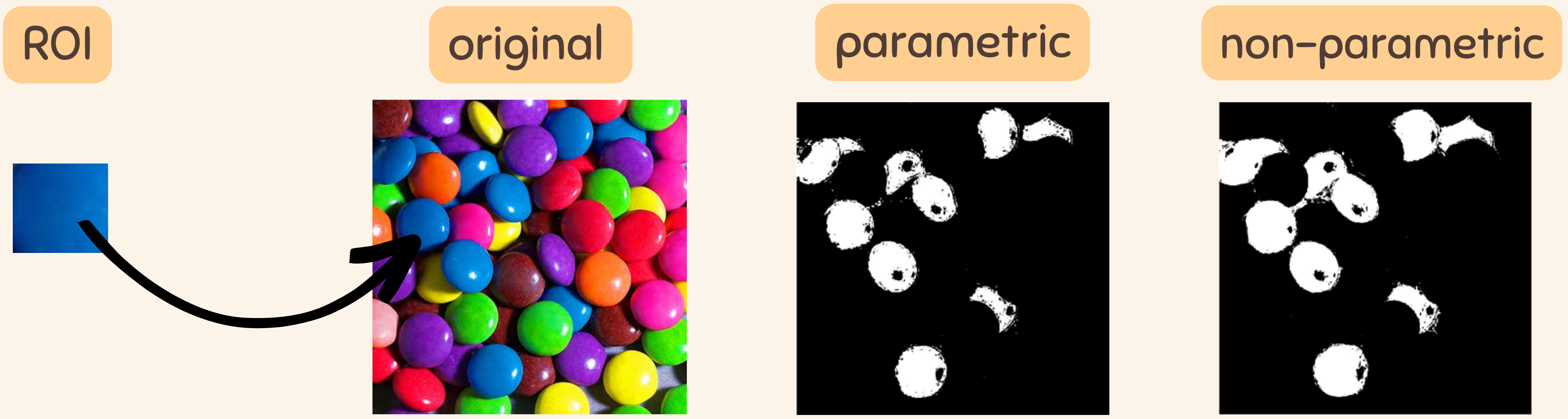
NON-PARAMETRIC: BIN SIZE EFFECT

What if we change the bin size of the histogram for the non-parametric segmentation? Using the same ROI (i.e. the beige-colored square), let's investigate what is the effect of decreasing the bin size. Decreasing the bin size, as shown below, increases the segmented regions. As we can see, with bins = 30, the image appears to have more white regions than the one with 45 bins.



PARAMETRIC VS. NON-PARAMETRIC

Let's try the segmentation techniques in 3D objects. Here is an image of different-colored candies. Notice that there are shading variations for each of the candies. Some parts are dark and some are brighter even if they are the same color. We use the blue region as our ROI. After segmenting, we see that both techniques were able to capture all the blue regions in the picture. However, the non-parametric was able to capture more regions.



What's with the black holes inside the regions? These holes correspond to the glare caused by the scattering of light when the image was taken. It appears to be white patches in the image. There are techniques on how to fill these holes which I believe we'll tackle in the next activities.

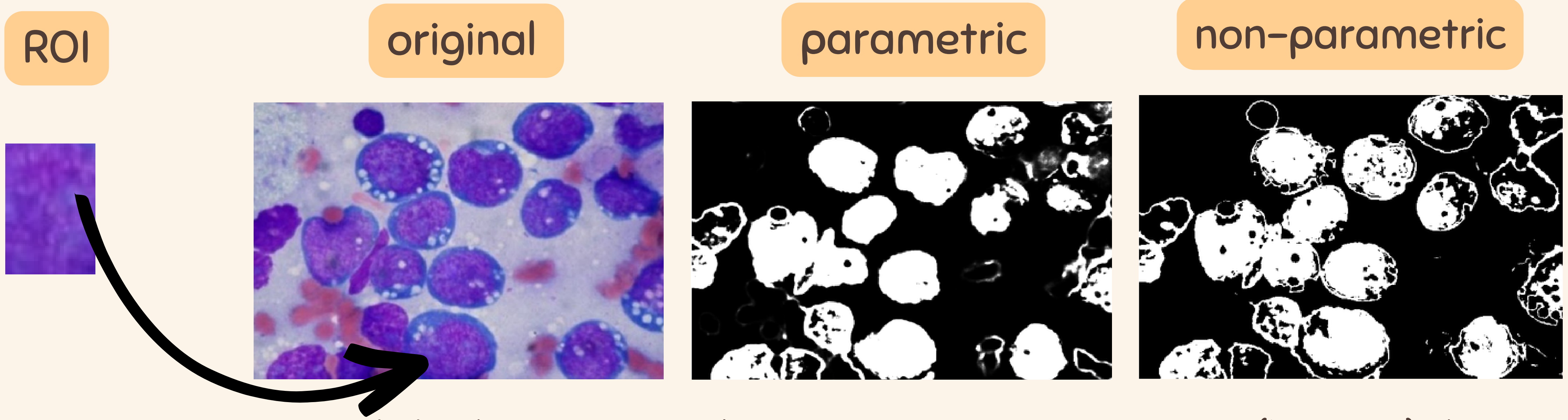
03

APPLICATIONS OF IMAGE SEGMENTATION

Image segmentation has numerous applications in fields such as medical imaging, object recognition, robotics, etc. In this section, we'll look at some of the applications of image segmentation.

APPLICATION 1: MICROSCOPY

Image segmentation is applied in microscopy. Microscopy is not just about observing, but also requires measuring or quantifying. Automated extraction of image data is a big step for the digitization of microscopy. The process of segmentation helps in analyzing certain regions in digital images obtained from microscopes. Here we have an image of stained bacteria. Let's try if we can segment only the desired region.

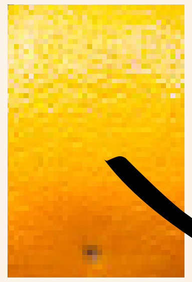


The parametric segmentation provided a cleaner output. The non-parametric segmentation (bins = 30) also segmented the image, however, there are still noise present. Maybe adjusting the bin size would improve the quality of the output.

APPLICATION 2: OBJECT RECOGNITION

Image segmentation is also used in object recognition algorithms. This helps identify and isolate the objects from the background. It is a crucial step in object recognition as the next step after it would be analysis of the segment to determine if it represents an object or just a part of it. Here, we use an image of oranges hanging on a tree. We'll try to segment only the oranges.

ROI



original

parametric

non-parametric

Both the techniques successfully segmented the image of the oranges. Notice, however, that the non-parametric segmentation has a cleaner output image.

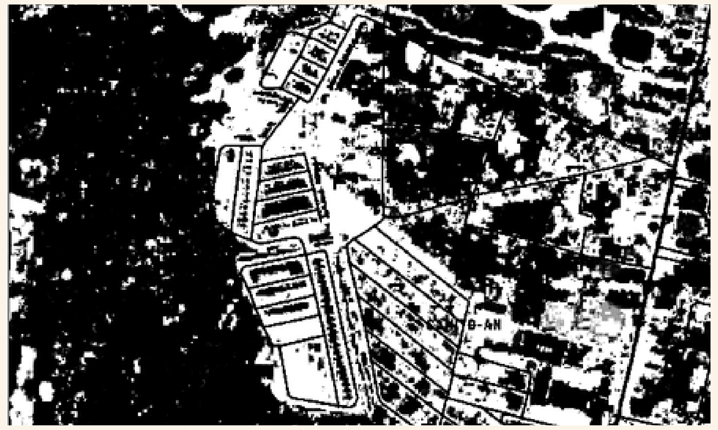
APPLICATION 3: REMOTE SENSING

Another application of image segmentation is remote sensing. Remote sensing is the collection of information about objects or phenomena on earth Earth's surface using sensors mounted on satellites or aircrafts. Image segmentation plays an important role in identifying and classifying different land cover types within satellite or aerial images. Here I used an image I borrowed from my labmate's thesis. Suppose we want to get the land parts which are black (i..e. no houses or vegetation), let's see if we can do it using the two techniques.

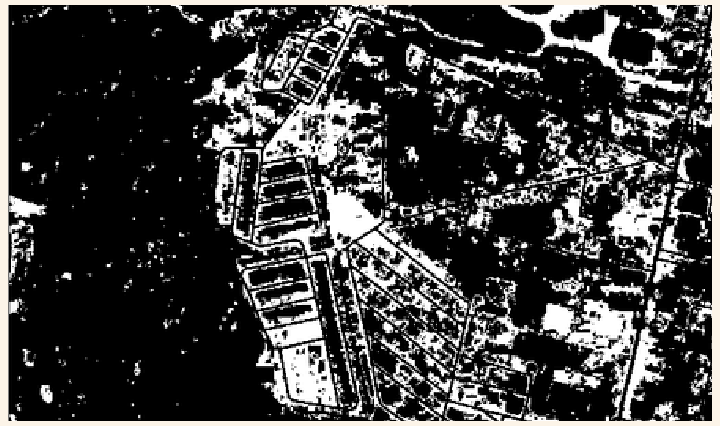
ROI

original

parametric



non-parametric



The output images seem to still have noise included. Nevertheless, both techniques managed to segment the desired regions. It would have been better if the images zoomed in for the segmentation to get better results.

KEY TAKEAWAYS

- Image segmentation is an essential process in different fields of research.
- Parametric and non-parametric segmentation both have advantages and disadvantages.
- Bin size affects the output of the non-parametric segmentation. Decreasing the bin size , increases the segmented region.

REFLECTION

This is another fun activity to do. It was fascinating to discover that image segmentation has several applications in different fields and to be able to do it, even its simple algorithm, is a fun experience. However, I must admit that I got stuck for a while in trying to debug my code for the parametric segmentation. Since the syntax in MATLAB is quite different from what I used in Python before, it took me a while to debug my code. After some trials, I figured out that the only thing wrong with my code is that I should've used the operator ".^" for the squared quantities of the PDF instead of only "^". I will forever keep this in mind. After debugging the code, it was actually a smooth ride from there.

The ideas for the applications are actually influenced by my labmates' thesis. It gave me insight on how image segmentation is used in different applications.

REPORT GRADE

Criteria	Score
Technical Correctness	35
Quality of Presentation	35
Self Reflection	30
Initiative	10
TOTAL	100 (+10)

EVALUATION

Overall, I know that I have accomplished all the tasks included in this activity. Moreover, I did some extra experiments and made some comparisons in the resulting images. I also put a lot of effort in creating this presentation and cited a lot of sources for additional information in the report.