Feature Extraction: Image Segmentation

Tolledo, Charmaine S.

OVERVIEW OF THE ACTIVITY

Image feature extraction entails the identification and extraction of significant visual attributes, also known as features, from images. This is done with the aim of minimizing the quantity of data required to describe the image, while retaining its crucial details. [1]

In this activity, we had the opportunity to go beyond basic visual information and explore more advanced image processing techniques. Last time, we used ImageJ, but this time, we focused on image segmentation, which involves coding and implementation. By employing these techniques, let's aim to extract important features and patterns from the image and uncover valuable insights from our visual data!

OBJECTIVES

- Performing image segmentation on grayscale and color images.
- Applying Parametric and Non-parametric probability distribution estimation technique for color image segmentation.
- Comparing the effectiveness of the two different segmentation techniques.

GRAYSCALE IMAGE SEGMENTATION

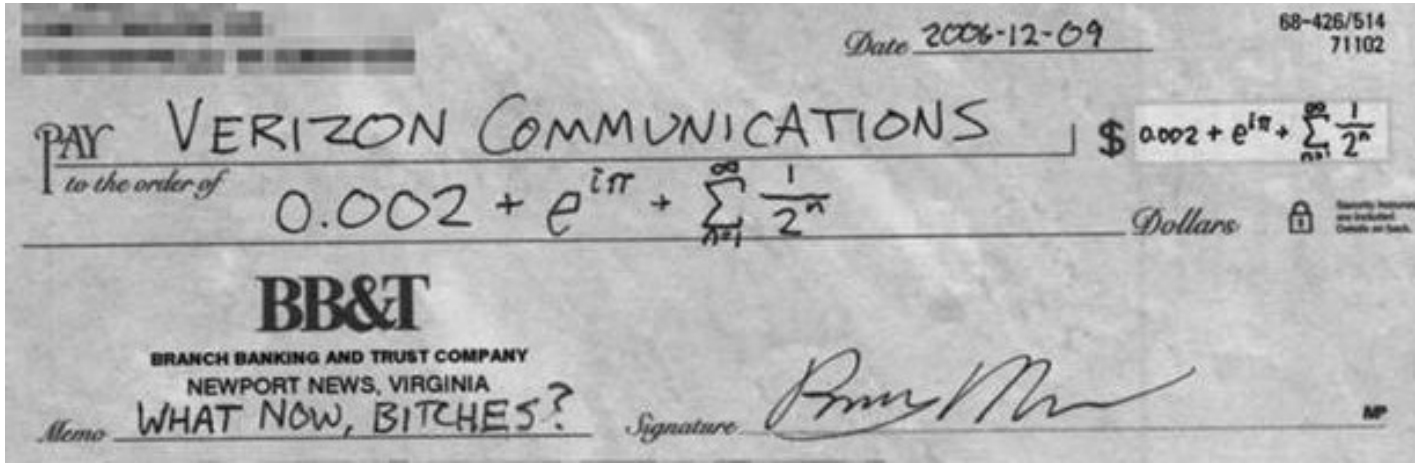


Figure 1
Cropped grayscale check

Image segmentation is a fundamental image processing operation that is essential for analyzing and comprehending digital images. It involves dividing an image into subdivisions known as image regions or segments, which enables the identification and extraction of particular objects or regions of interest. [2] This is easily accomplished with grayscale images. Figure 1 above shows the grayscale image we wish to segment.

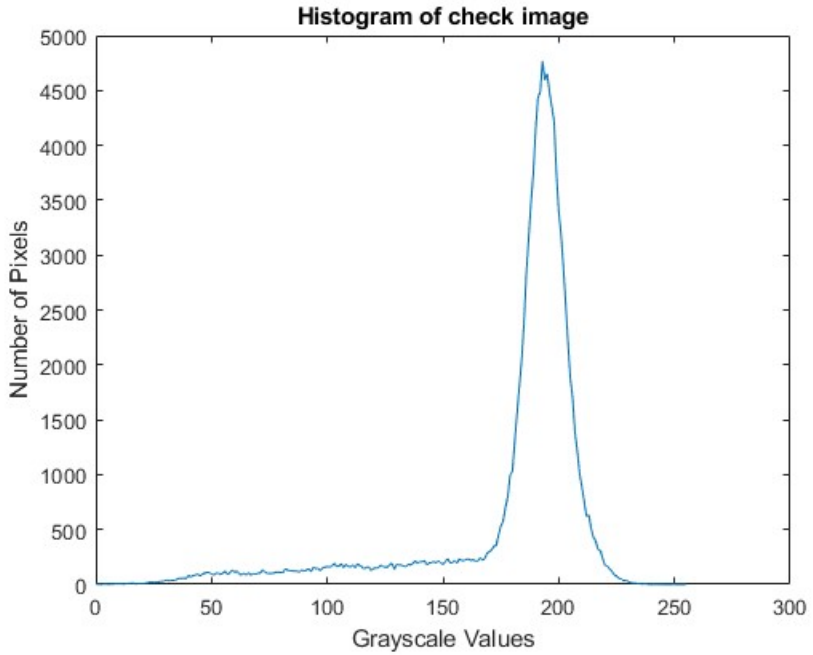


Figure 2
Histogram of the cropped grayscale check

The figure above shows the histogram of the cropped grayscale check from the previous slide. This demonstrates the relationship between grayscale values and the number of pixels containing those values in the provided image. Analyzing the graph, it is evident that a sharp peak is observed, indicating that a substantial number of pixels have grayscale values in the range of 180 to 200.

SEGMENTED GRAYSCALE IMAGE

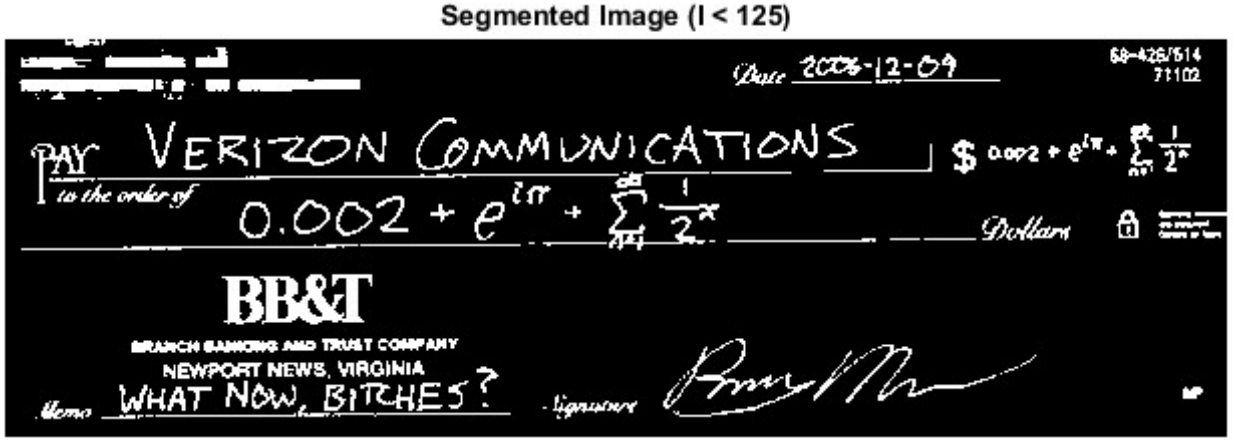
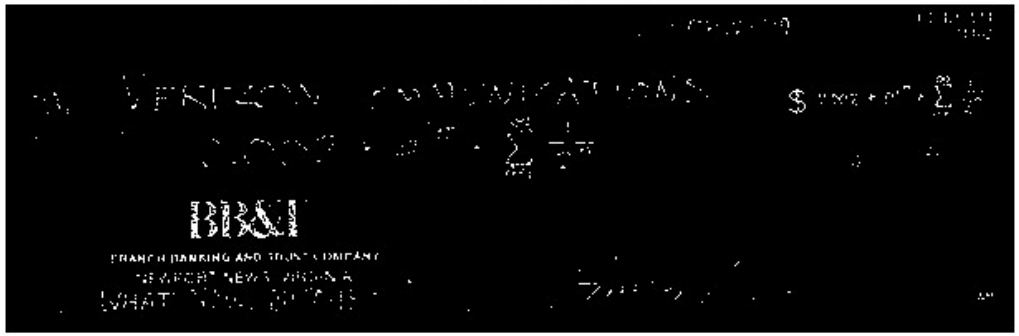


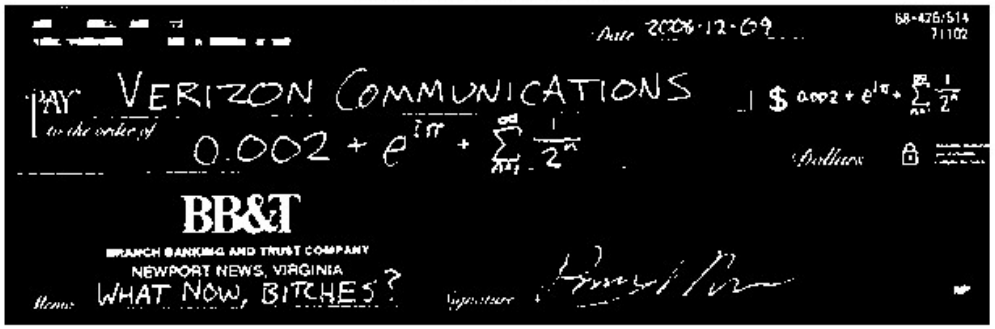
Figure 3

This is the segmented image of the grayscale check. To segment the handwritten and printed parts, the threshold value was set to 125, this means that pixels with values below 125 are considered foreground and set to true (1), while pixels with values greater than or equal to 125 are considered background and set to false (0). Now that we segment it, we can clearly see that the handwritten and printed parts were highlighted. By doing so, we separated and distinguished these two types of information which can be essential to subsequent analysis or processing tasks.

Segmented Image (I < 50)

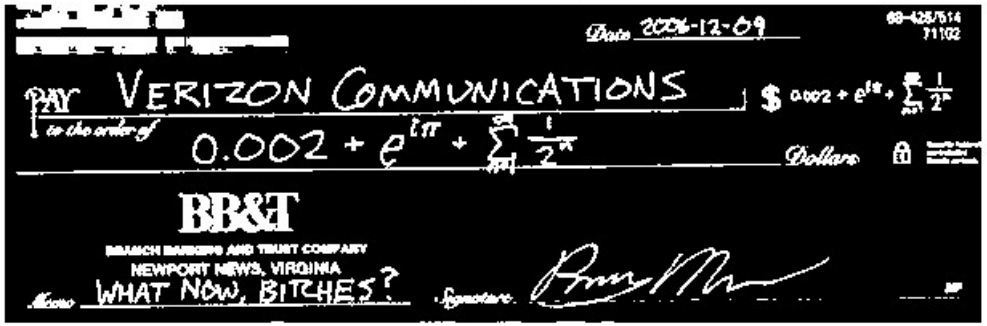


Segmented Image (I < 100)



Segmented Image (I < 75)

Segmented Image (I < 150)



Segmented Image (I < 175)

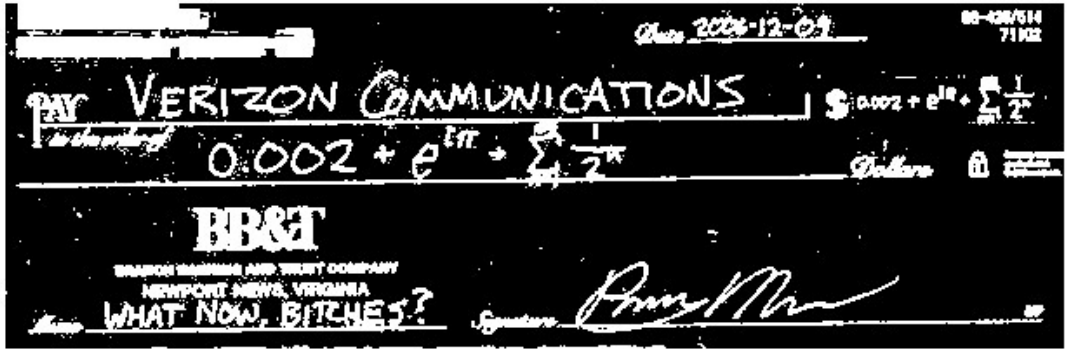


Figure 4

SEGMENTED GRAYSCALE IMAGE WITH VARYING THRESHOLD VALUES

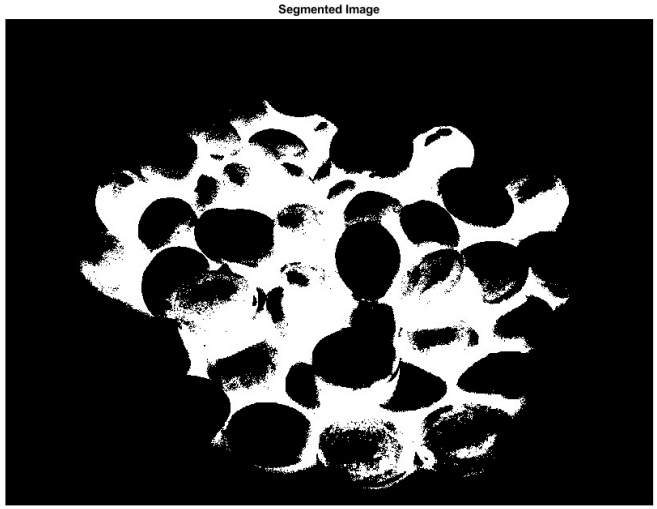
The preceding slide shows the segmented image with different threshold values. The values range from 50 to 175, meaning values below the peak of the histogram to avoid the background to be white. Here, we can observe that the lower threshold values (e.g. 50 to 100) resulted to permissive segmentation, which lead to less accurate separation of desired regions since significant portions of noise were included in the segmented image. On the other hand, higher threshold values (e.g. 150 to 175) resulted to a restrictive segmentation that lead to exclusion of some lighter or darker area within the desired regions. With this, we can say that the value of 125 gave us more accurate segmentation of the handwritten and printed parts.

WHAT HAPPENS IF WE GRAYSCALE SEGMENT A COLORED IMAGE?

https://supermarketkid.files.wordpress.com/2013/07/img_4848.jpg

(a) Colored Image

(b) Grayscale Image



(c) Segmented Image

Figure 5

Here, the image was converted to grayscale and then segmented. Figure b shows that the colors are indistinguishable because their grayscale values are nearly identical. And as can be seen plainly in Figure c, the segmentation algorithm only considered the intensity values of the grayscale image and not the original color information, resulting in the loss of color details during segmentation as only the intensity values of the grayscale image are used to separate regions. Herein lies the role of color segmentation.

COLOR IMAGE SEGMENTATION

Instead of RGB, color segmentation requires the use of a color space that represents brightness and chromaticity, such as the Normalized Chromaticity Coordinates (NCC). By normalizing the RGB channels, new coordinates (r and g) that define the color of each pixel are obtained.

$$r = \frac{R}{R+G+B} = \frac{R}{I}, g = \frac{G}{I}, b = \frac{B}{I}.$$

Color segmentation is performed by estimating the probability of a pixel belonging to the color distribution in a specific region of interest (ROI). Color segmentation is essential for accurately separating and analyzing regions or objects in colored images, taking into consideration shading variations and capturing both brightness and chromaticity information. This can be accomplished using either parametric or nonparametric methods for estimating probability distributions. [3]

PARAMETRIC PROBABILITY DISTRIBUTION ESTIMATION

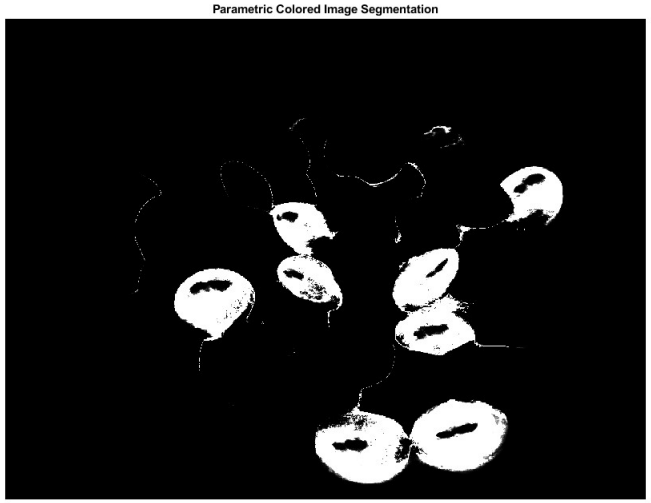


Figure 6

(a) Colored Image **(b) Parametric Colored Image Segmentation** **(c) ROI**

This shows the segmentation of a color image based on a parametric estimate of the probability distribution. Here, I wanted to segment the violet nip colors. So, I extracted the r and g components from the RGB channels of the input image and normalized them based on the intensity values. Then, the mean and standard deviation are computed for the r and g components, which serve as parameters for the assumed normal distributions. Using these parameters, probability density functions (PDFs) are created, and the segmentation is completed by multiplying the PDFs. The resulting segmentation, as seen in Figure b, is presented as a grayscale image and it appears like it effectively separates the selected ROI in the image, but there are some detected noise that contributed to the imperfection of the segmentation.

NON-PARAMETRIC PROBABILITY DISTRIBUTION ESTIMATION

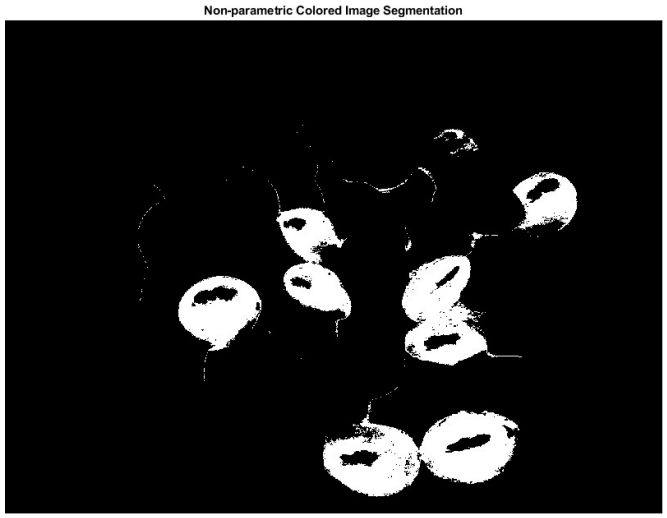


Figure 7

(a) Colored Image **(b) Non-parametric Colored Image Segmentation** **(c) ROI**

Similar to the previous slide, I wished to segment the violet nip colors on the image, but this time I applied non-parametric probability distribution estimation. Here, the r-g color space ROI from the original image is cropped. After converting the r and g components to discrete indices, the two-dimensional rg-histogram is computed. This histogram shows the hue combination distribution within the ROI. The histogram is then backprojected onto the entire image by assigning the r and g components of each pixel to the histogram indices. The resulting backprojected histogram image depicts the probability that each pixel belongs to the color distribution in the ROI. As shown in Figure b, the result was nearly identical to that of parametric segmentation. To better distinguish the differences between them, let's segment more!



Figure 8

(a) Colored Image **(b) Parametric Colored Image Segmentation** **(c) Non-parametric Colored Image Segmentation** **(d) ROI**

Comparing the results, I can conclude that the parametric technique produced a significantly smoother and cleaner segmentation than the non-parametric technique here. This may be because the image's color distribution within the ROI closely matched the Gaussian distribution, which is the assumed parametric distribution. This indicates that pixel values display a comparatively uniform and seamless variation, devoid of significant outliers and complex patterns. This uniformity enables the parametric model to capture and represent the color distribution accurately, which resulted in a clean segmentation.



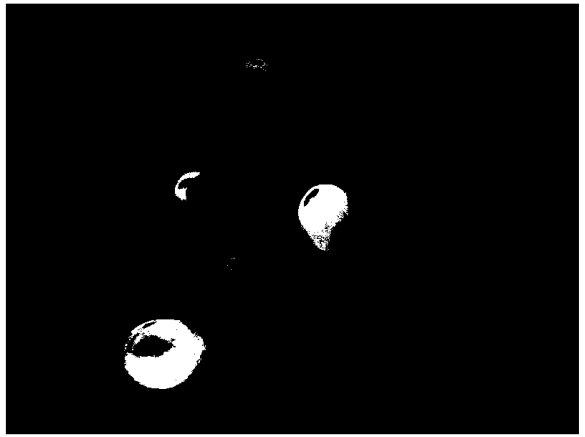


Figure 9

(a) Colored Image **(b) Parametric Color Image Segmentation** **(c) Non-parametric Color Image Segmentation** **(d) ROI**

Here, I also used the same colored image but with a different region of interest, and as you can see, the non-parametric technique produced a segmentation that was smoother and neater than the other technique. But how does it happen? Always bear in mind that the specified region of interest has a substantial impact on the segmentation result. It is possible that the ROI here had a more complex color distribution that the presumed parametric distribution did not accurately show. Non-parametric method, on the other hand, can adjust to unique color distributions and manage variations more effectively, resulting in segmentations that are smoother and cleaner.

REFLECTION

I can honestly say that I was anxious about this, but I also had a great time doing it. I was tense because I had so many other tasks to complete, but I enjoyed the part that involved coding because I was able to complete it much more quickly than in the past acitivities. TThanks to the readily available code provided by Ma'am Jing. Only when attempting to perform the parametric image segmentation technique did I experience any difficulty. I encountered few errors, but the error I still vividly recall is that I missed the part where, when calculating the mean and standard deviation, I failed to linearize the matrix into a column vector so that they could operate on the individual elements of the matrix. ut after some troubleshooting and debugging, I was able to correct it. Overall, this was so enjoyable.And with the efforts I put into this activity, I believe I am deserving of **10/10!**

REFERENCES

[1]https://www.mygreatlearning.com/blog/feature-extraction-in-imageprocessing/#:~:text=Feature%20extraction%20is%20a%20part,a%20large%20number%20of%20variables.
[2]https://datagen.tech/guides/image-annotation/image-segmentation/#:~:text=Image%20segmentation%20is%20a%20method,analysis%20of%20each%20image%20segment.
[3] Activity 5 Lab Manual

End