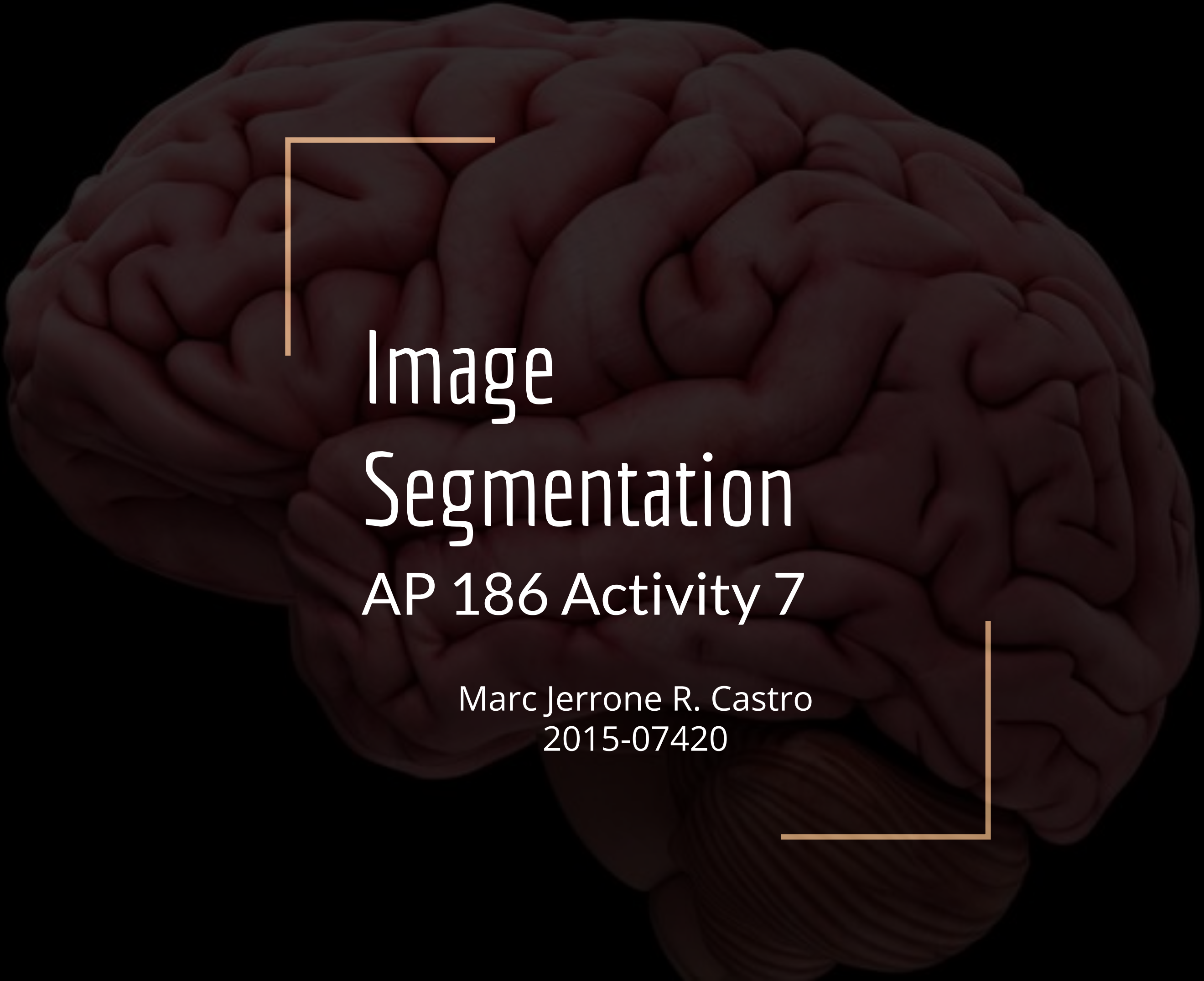


Image Segmentation

AP 186 Activity 7

Marc Jerrone R. Castro
2015-07420

Image Segmentation

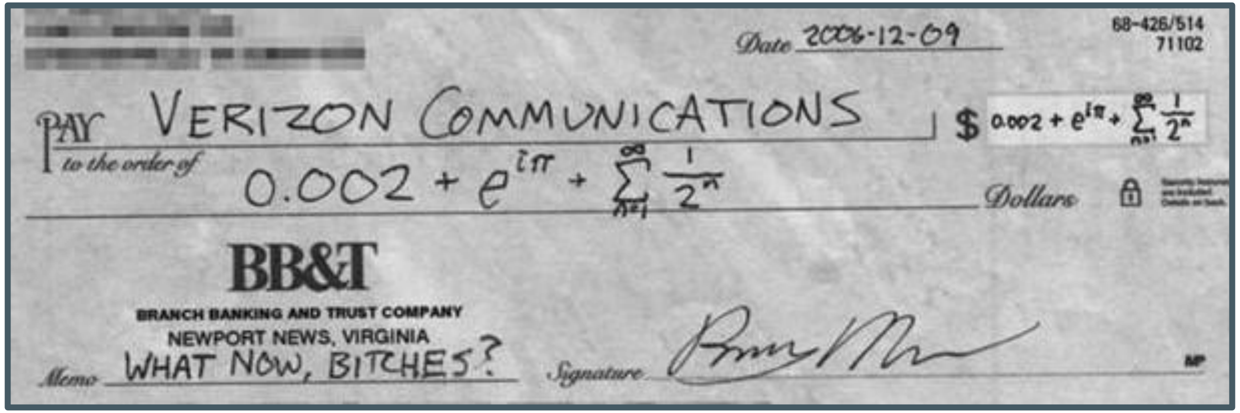
Image segmentation is the act of partitioning an input image into distinct regions which are distinct from one another on the basis of respective pixel attributes. There are several algorithms and methods to segment images such as using feature maps, color thresholds, region analysis, and a lot more.

Images are typically segmented to isolate regions of interest (ROIs) which would later be used in analysis and in some cases as ground truths.

In this study, we use color thresholds to segment a monochromatic object from its background via Parametric and Non-Parametric Probability Distribution Estimates.

Grayscale Image Segmentation

We use the image shown in Figure 1 as our input for gray scale segmentation. As observed, there are only distinct shades of black and white within the image and upon looking at its' histogram as shown in Figure 2,



[redacted]

Date 2006-12-09

68-426/514
71102

PAY to the order of VERIZON COMMUNICATIONS $ $0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$

$0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$ Dollars

BB&T

BRANCH BANKING AND TRUST COMPANY
NEWPORT NEWS, VIRGINIA

Memo WHAT NOW, BITCHES? Signature [illegible]

MP

Figure 1.0 Grayscale image of a check

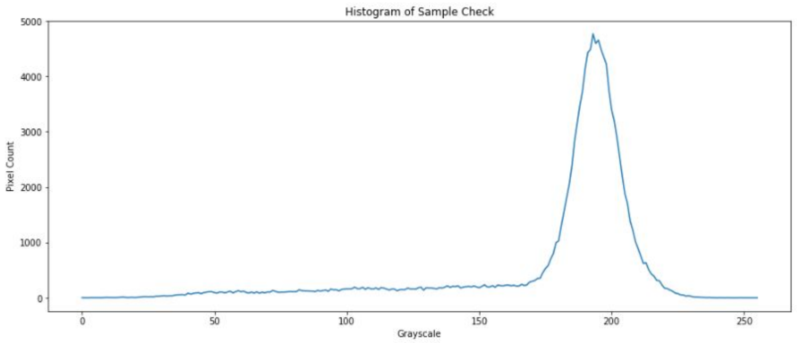
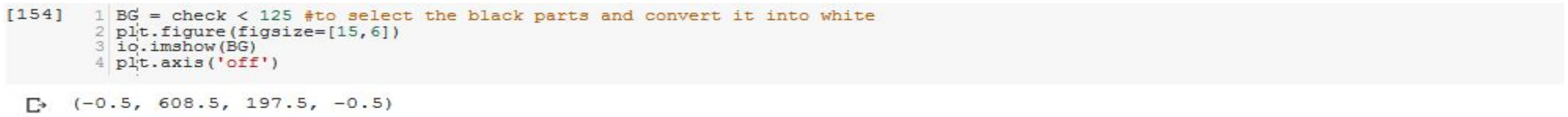


Figure 2.0 Generated histogram of the check.

We can observe that there is a significant difference between the text and writings on the check as with the background. We can also observe that a majority of the image contains non-ROIs and as such the use of image segmentation would limit the amount of information the computer would have to process.

Grayscale Image Segmentation



```
[154]  BG = check < 125 #to select the black parts and convert it into white
       plt.figure(figsize=[15,6])
       io.imshow(BG)
       plt.axis('off')
```

```
(-0.5, 608.5, 197.5, -0.5)
```

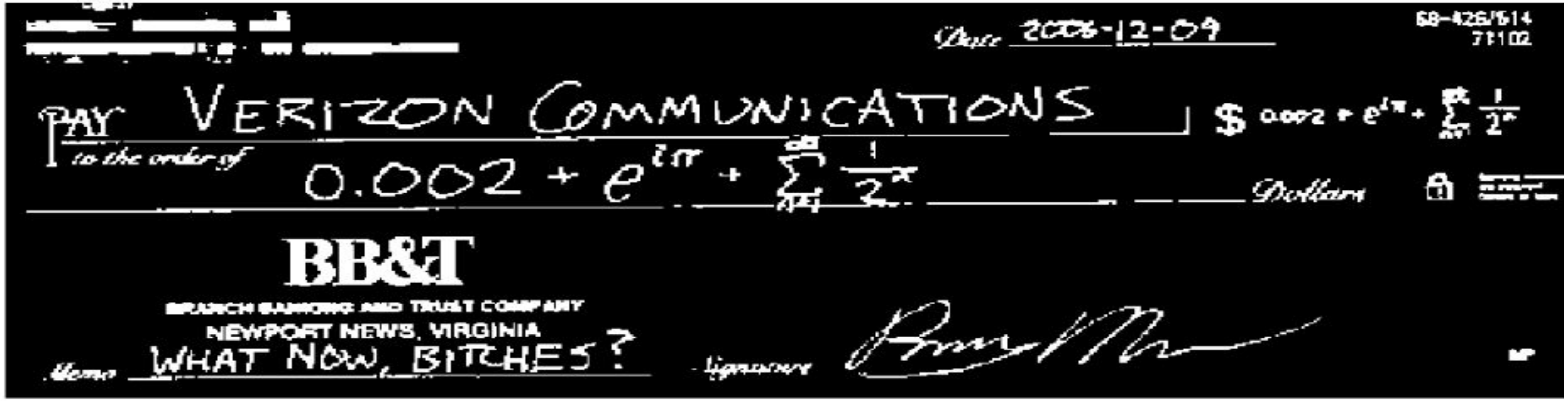


Figure 2.0 Segmented text of the input image on the basis of the pixel values located below a threshold of 125.

Shown in Figure 3, are the segmented texts which we wanted to retrieve from the input image. We limited the image to pixel values which had intensities smaller than 125. With this, we are left with a binary image which was formed on the basis that a specific pixel would have a value of *True* or 1 if it had a value less than 125 and *False* or 0, otherwise.

Grayscale Image Segmentation

```
[155]  BG = check > 125 #to select the white parts (background) and convert it into white
       plt.figure(figsize=[15,6])
       io.imshow(BG)
       plt.axis('off')
```

```
(-0.5, 608.5, 197.5, -0.5)
```

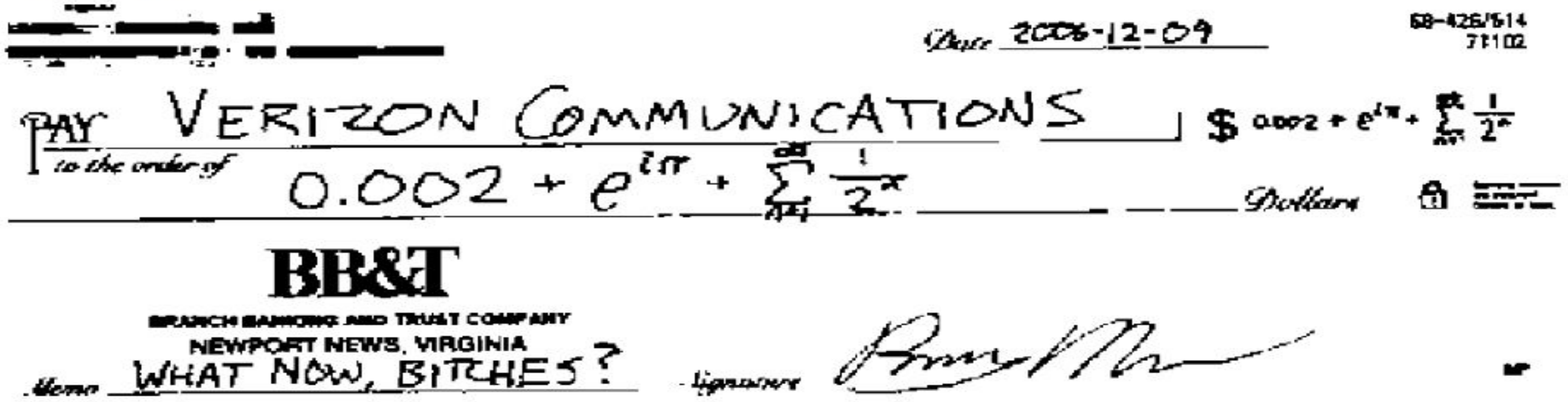


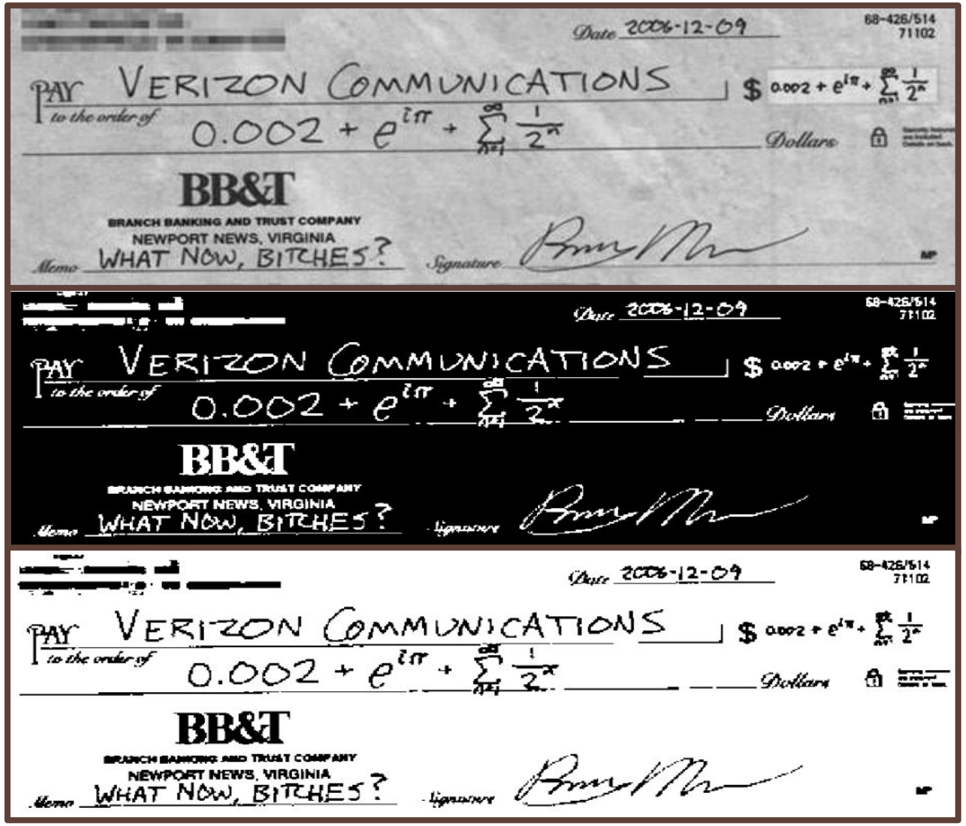
Figure 3.0 Segmented text of the input image on the basis of the pixel values located above a threshold of 125.

Shown in Figure 4, on the other hand, is an isolation of the background whereas the values are reversed such that all the background pixels have a value of *True* or 1 and all text pixels have a value of *False* or 0.

Grayscale Image Segmentation

From this activity we learned that:

- Using a distinct contrast between colors, we are able to segment images on the basis of their pixel values.
- Although a majority of the desired image is segmented, it is almost never the case that all desired portions (e.g. thin lines under signature is not segmented) are captured.
- Alpha values and other shades are loss in this type of segmentation as we limit the values to either *True* or *False* values, and as such it fails to capture the varying shades of black within the image.



[redacted]

Date 2006-12-09

68-426/514
71102

PAY to the order of VERIZON COMMUNICATIONS $ $0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$

$0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$ Dollars

BB&T
BRANCH BANKING AND TRUST COMPANY
NEWPORT NEWS, VIRGINIA

Memo WHAT NOW, BITCHES?

Signature [illegible]

Figure 4.0 Image segmentation performed on a grayscale image of a check. (Top to Bottom) Input Image, Segmentation of text, Segmentation of background.

Image Segmentation via Parametric Probability Distribution Estimation

We can also perform segmentation on colored objects by determining the probability that a pixel belongs to a certain color distribution of interest.

By extracting an ROI from the image, we isolate our specific range by determining a corresponding histogram value regarding the pixel values located in that patch. As we are dealing with a 3D colored image, we must first separate the RGB channels and flatten down the intensities of each channel such that we are left with a version of the original image narrating the chromaticity and brightness of each pixel.

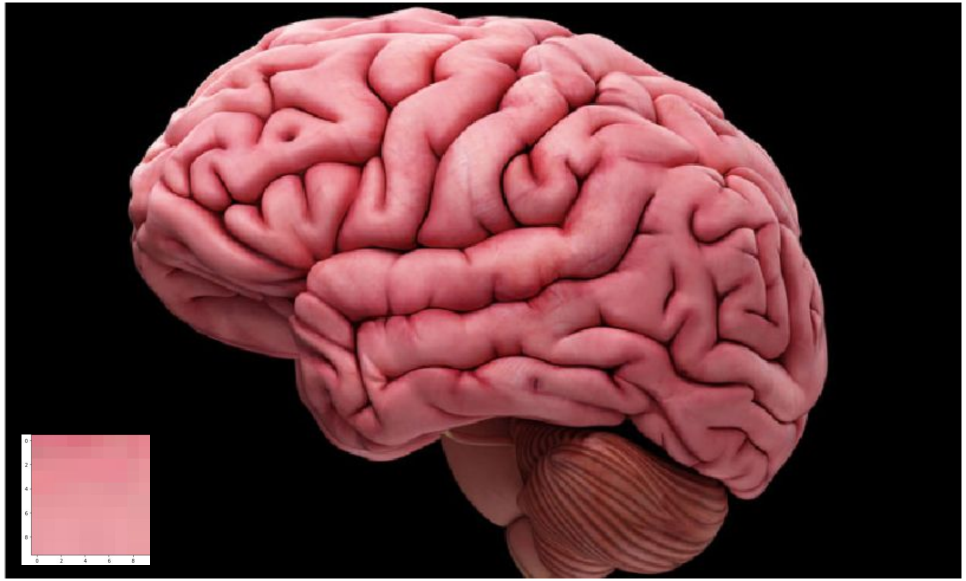
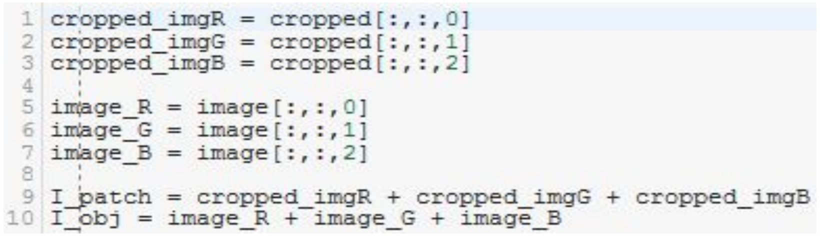


Figure 5.0 Colored image of a model of a human brain retrieved from [1]. Located on the bottom-left corner of the image is the ROI we extracted from the image.

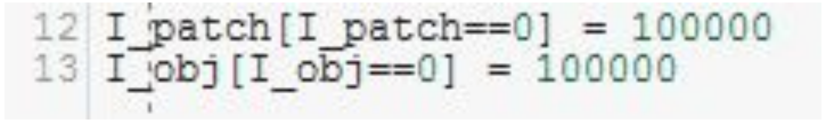


```
cropped_imgR = cropped[:,:,0]
cropped_imgG = cropped[:,:,1]
cropped_imgB = cropped[:,:,2]

image_R = image[:,:,0]
image_G = image[:,:,1]
image_B = image[:,:,2]

I_patch = cropped_imgR + cropped_imgG + cropped_imgB
I_obj = image_R + image_G + image_B
```

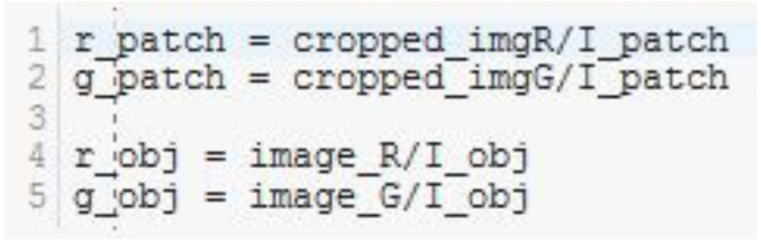
Image Segmentation via Parametric Probability Distribution Estimation



```
I_patch[I_patch==0] = 100000
I_obj[I_obj==0] = 100000
```

We then replace all *null* or background pixel values with 100000 as to prevent errors when we normalized our image using the following snippet of code.

Afterwards, we determine the normalized chromaticity coordinates of each pixel using the following snippet of code,



```
r_patch = cropped_imgR/I_patch
g_patch = cropped_imgG/I_patch

r_obj = image_R/I_obj
g_obj = image_G/I_obj
```

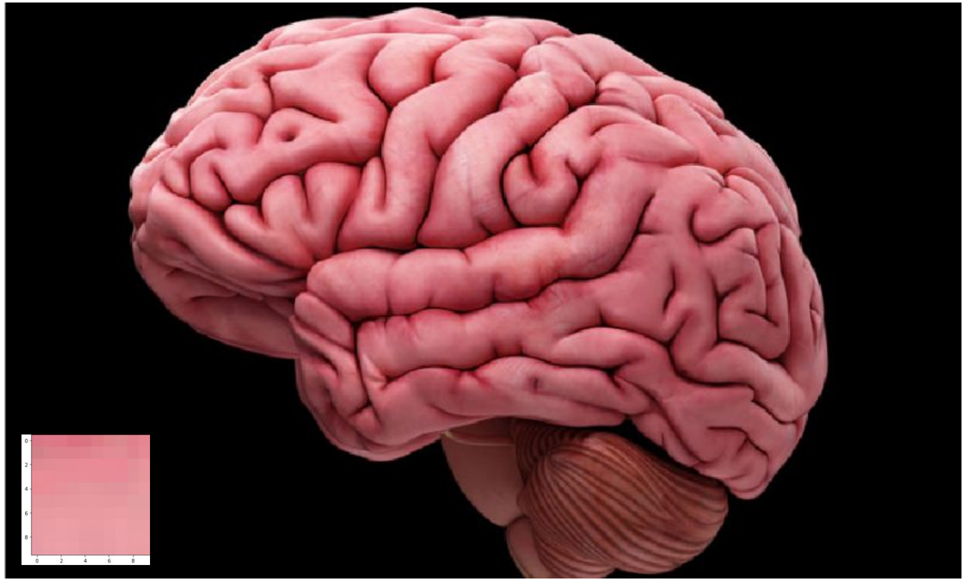


Figure 5.0 Colored image of a model of a human brain retrieved from [1]. Located on the bottom-left corner of the image is the ROI we extracted from the image.

Image Segmentation via Parametric Probability Distribution Estimation

Since we normalized the values by the chromaticities and brightness of each channel. It is enough to use only two channel (r and g channels) as our frames of reference. Thus we now have information of the red and green channel which we shall use as the basis for chromaticity, as well as the intensities of each pixel.

We then plot the chromaticities of our image such that we can qualitatively determine if our patch or ROI minimizes the amount of data we need to process.

We also perform this on the original image as a point of reference.

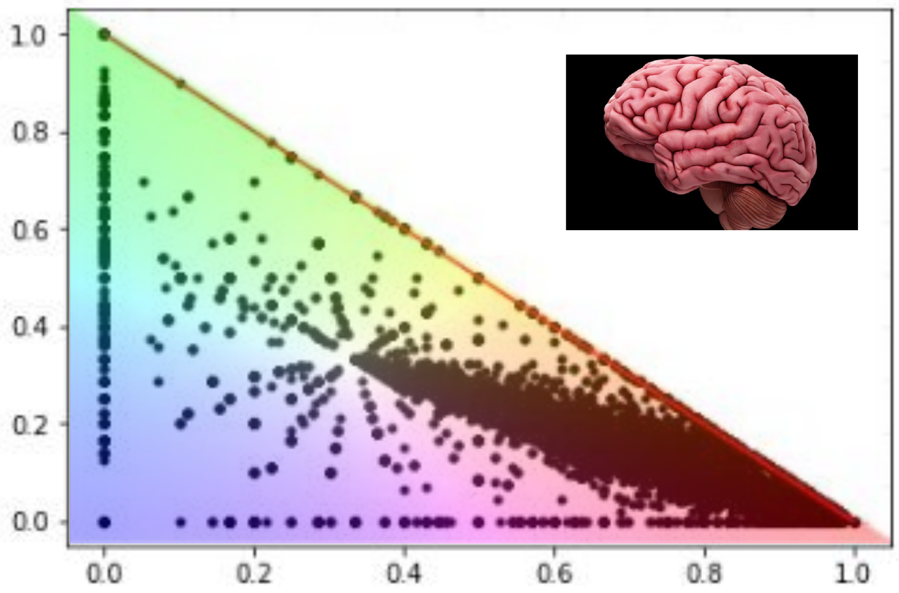


Figure 6.0 Normalized chromaticity coordinates (NCCs) of pixels located in the original input image whereas the x-coordinates correspond to chromaticities in the r channel and the y-coordinates correspond to the chromaticities in the g channel. The projected NCCs are as expected as majority of our input image is of the red to pink color.

Image Segmentation via Parametric Probability Distribution Estimation

Since we normalized the values by the chromaticities and brightness of each channel. It is enough to use only two channel (r and g channels) as our frames of reference. Thus we now have information of the red and green channel which we shall use as the basis for chromaticity, as well as the intensities of each pixel.

We then plot the chromaticities of our image such that we can qualitatively determine if our patch or ROI minimizes the amount of data we need to process.

We also perform this on the original image as a point of reference.

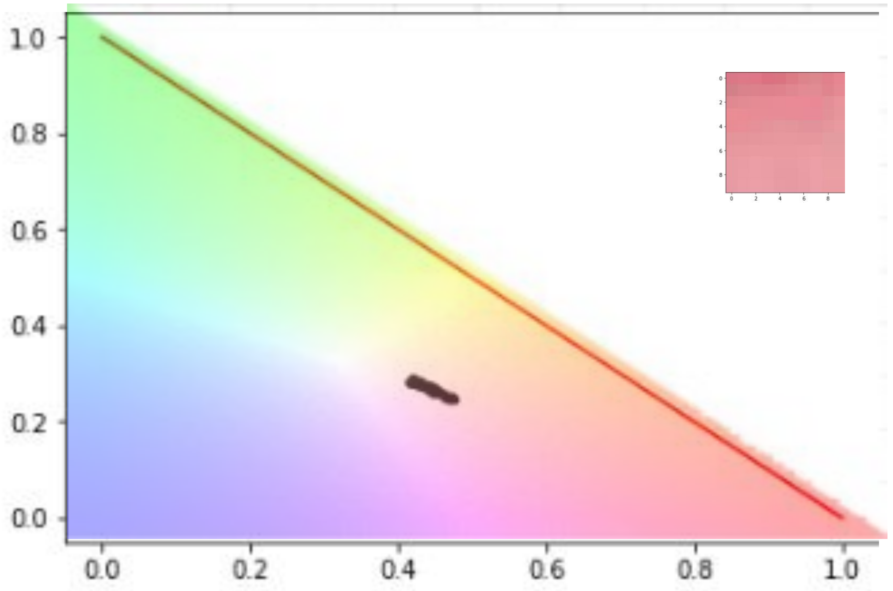


Figure 7.0 Normalized chromaticity coordinates (NCCs) of pixels located in the cropped image whereas the x-coordinates correspond to chromaticities in the r channel and the y-coordinates correspond to the chromaticities in the g channel. As this a smaller portion of the image, the resulting NCCs is as expected.

Image Segmentation via Parametric Probability Distribution Estimation

Using the following equations for parametric probability distribution estimation, we then calculate for the probabilities at both r and g,

$$p(r)=\frac{1}{\sigma_r\sqrt{2\pi}}\exp\left\{-\frac{(r-\mu_r)^2}{2\sigma_r^2}\right\}$$

$$p(g)=\frac{1}{\sigma_g\sqrt{2\pi}}\exp\left\{-\frac{(g-\mu_g)^2}{2\sigma_g^2}\right\}$$

We then determine their joint probability by performing element-wise matrix multiplication on each other. Afterwards, we visualize the correspond joint probability of the image as to visualize the resulting segmented image.

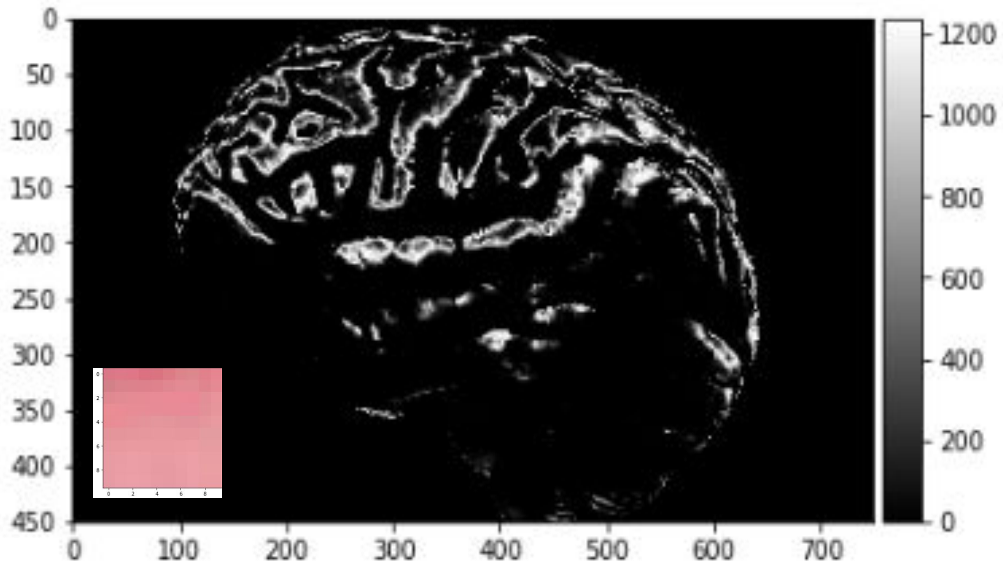


Figure 8.0 Segmented image using parametric probability distribution estimation.Located on the bottom-left corner is the patch or ROI utilized to calculate the NCCs of the corresponding image.

Image Segmentation via Parametric Probability Distribution Estimation

As a means for exploration, I performed the segmentation multiple times using varying sizes of ROIs. The following images are the results of each trial.

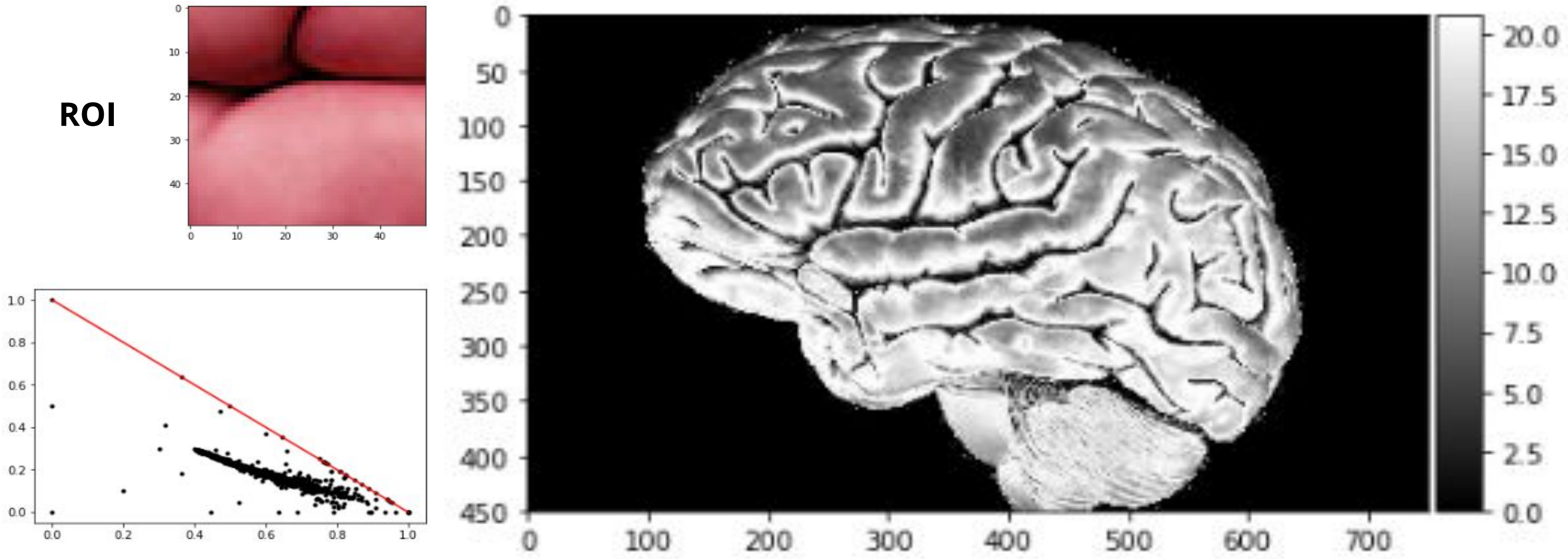
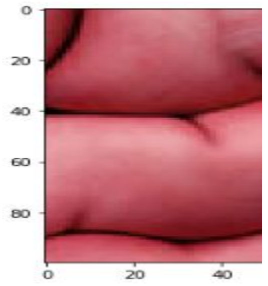


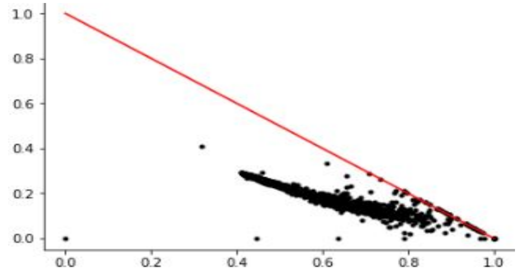
Figure 9.0 Segmented image using parametric probability distribution estimation. Located on the upper left is the ROI selected for this specific segmentation, while the plot on the bottom left of the image represent the NCCs for this particular ROI.

Image Segmentation via Parametric Probability Distribution Estimation

As a means for exploration, I performed the segmentation multiple times using varying sizes of ROIs. The following images are the results of each trial.

ROI





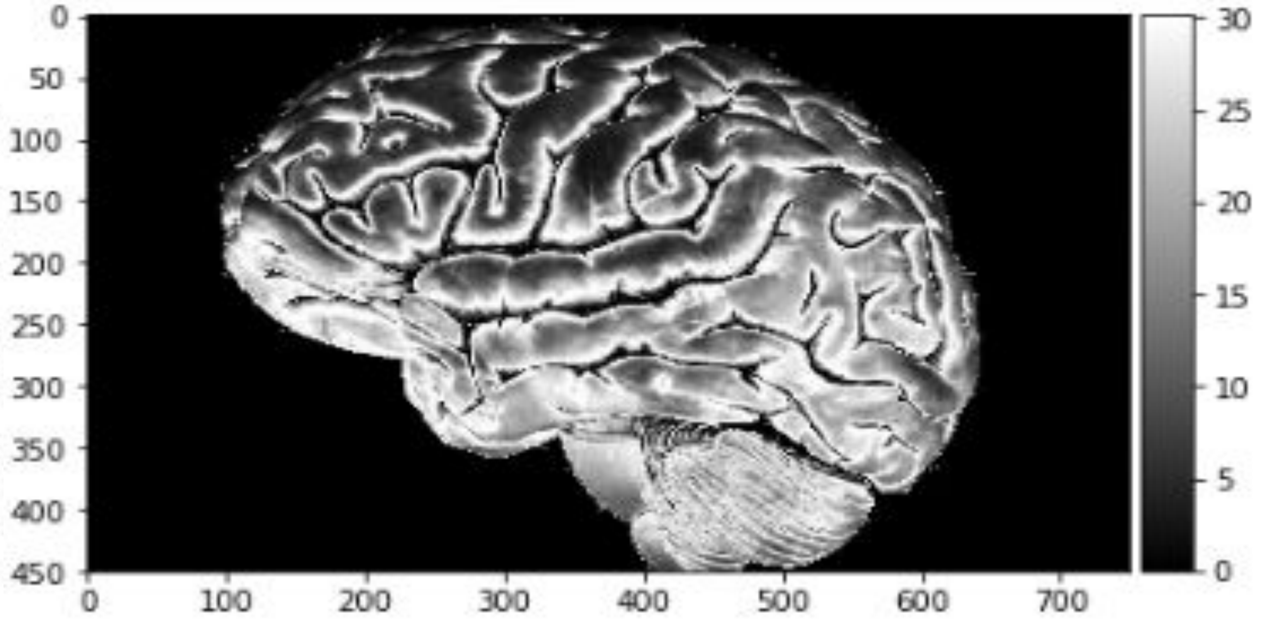


Figure 10.0 Segmented image using parametric probability distribution estimation. Located on the upper left is the ROI selected for this specific segmentation, while the plot on the bottom left of the image represent the NCCs for this particular ROI.

Image Segmentation via Parametric Probability Distribution Estimation

As a means for exploration, I performed the segmentation multiple times using varying sizes of ROIs. The following images are the results of each trial.

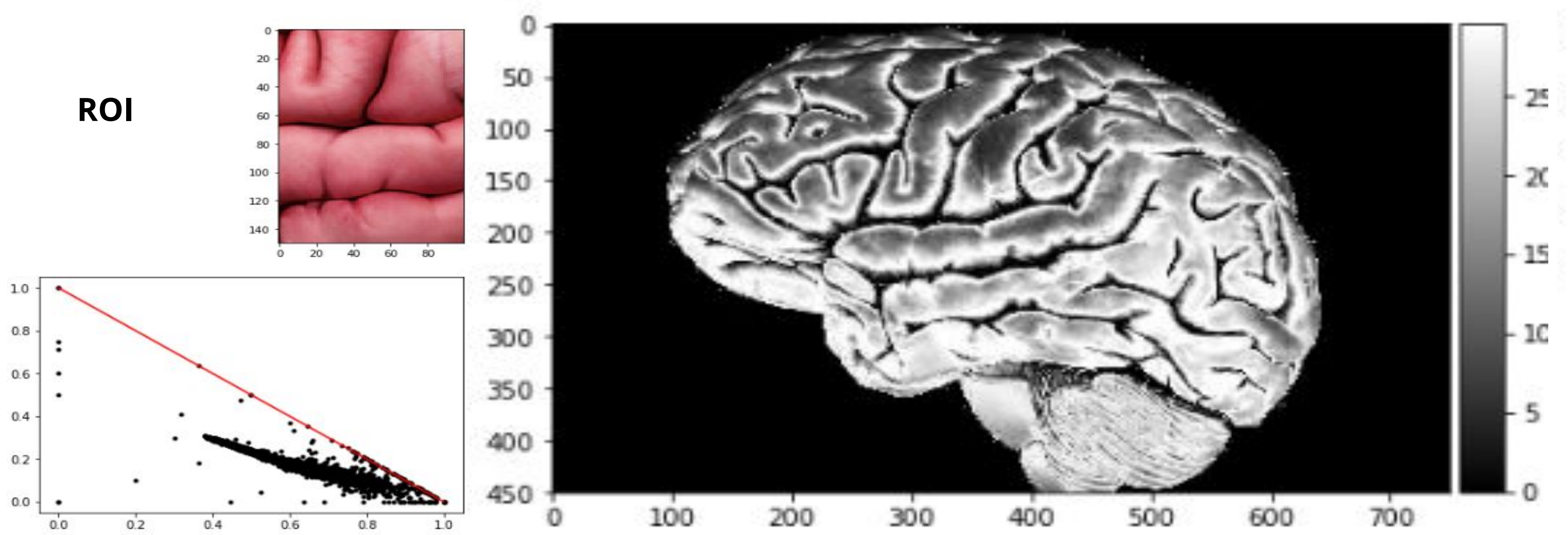


Figure 11.0 Segmented image using parametric probability distribution estimation. Located on the upper left is the ROI selected for this specific segmentation, while the plot on the bottom left of the image represent the NCCs for this particular ROI.

Image Segmentation via Parametric Probability Distribution Estimation

From my observations from increasing the size of the ROI,

1. The segmentation via color is highly dependent on the selected ROI.
2. Increasing the size of the ROI also increases the amount of NCCs for the ROI.
3. Altering the ROI affects the resulting segmented image, whereas for this example, as I included additional brighter portions of the image, the resulting segmentation became brighter (as observed from Figure 11). On the other hand, by including more darker shades of pink, the resulting segmentation turned comparatively darker (as observed From Figure 10).

Image Segmentation via Non-Parametric Probability Distribution Estimation

We utilize the same initialization methods for segmenting colored images in non-parametric probability distribution estimations. Such as,

1. From an initial input image select an ROI
2. Isolate r and g channels of the cropped and original image
3. Replace all null or zero pixel values at each channel with 100000
4. Determine the NCCs for the cropped image.

Afterwards, we determine the corresponding histogram of the NCCs and scale it with a predetermined bin size (for the sake of examples we utilize a bin size of 32).

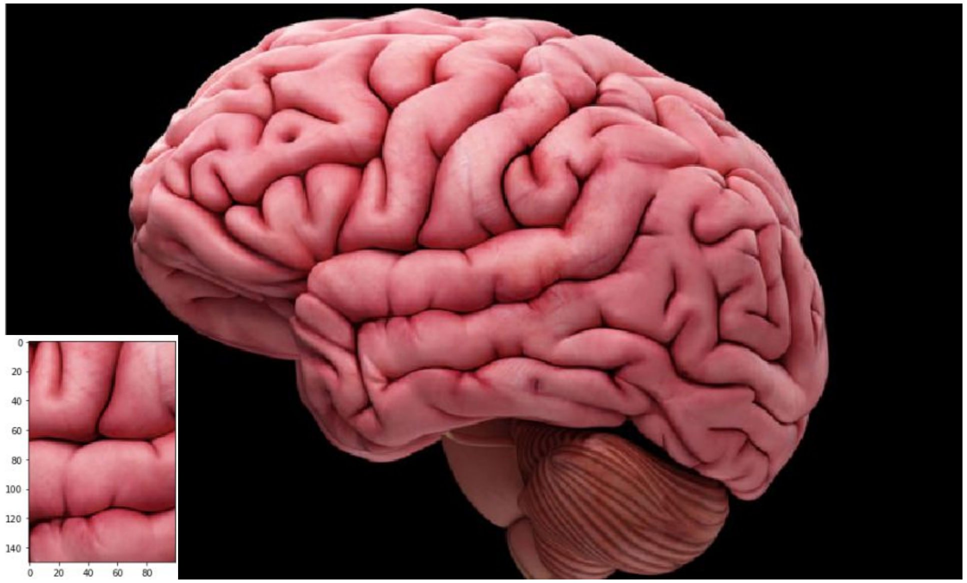
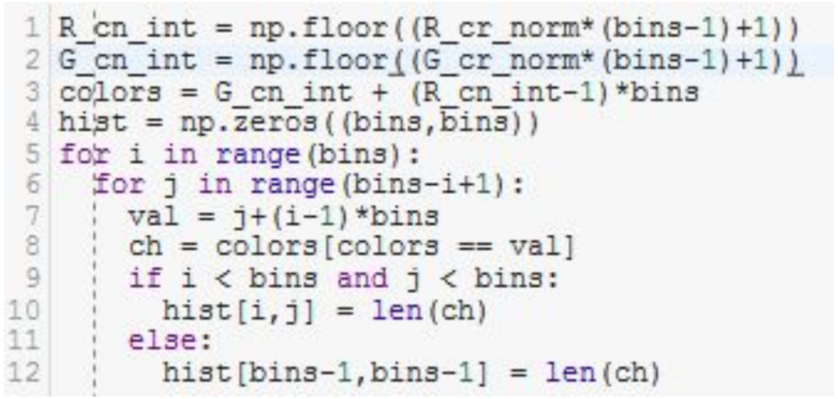


Figure 12.0 Colored image of a model of a human brain retrieved from []. Located on the bottom-left corner of the image is the ROI we extracted from the image.

Image Segmentation via Non-Parametric Probability Distribution Estimation

As for the purpose of faster processing, the non-parametric method utilizes minimal computing by using histogram backprojection. Such that, by generating a 2D histogram and binning it into a matrix. We can utilize this matrix as a look-up table with the indices being a function of the bin indices for r and g. This was implemented using the following snippet of code.



```
R_cn_int = np.floor((R_cr_norm*(bins-1)+1))
G_cn_int = np.floor((G_cr_norm*(bins-1)+1))
colors = G_cn_int + (R_cn_int-1)*bins
hist = np.zeros((bins,bins))
for i in range(bins):
  for j in range(bins-i+1):
    val = j+(i-1)*bins
    ch = colors[colors == val]
    if i < bins and j < bins:
      hist[i,j] = len(ch)
    else:
      hist[bins-1,bins-1] = len(ch)
```

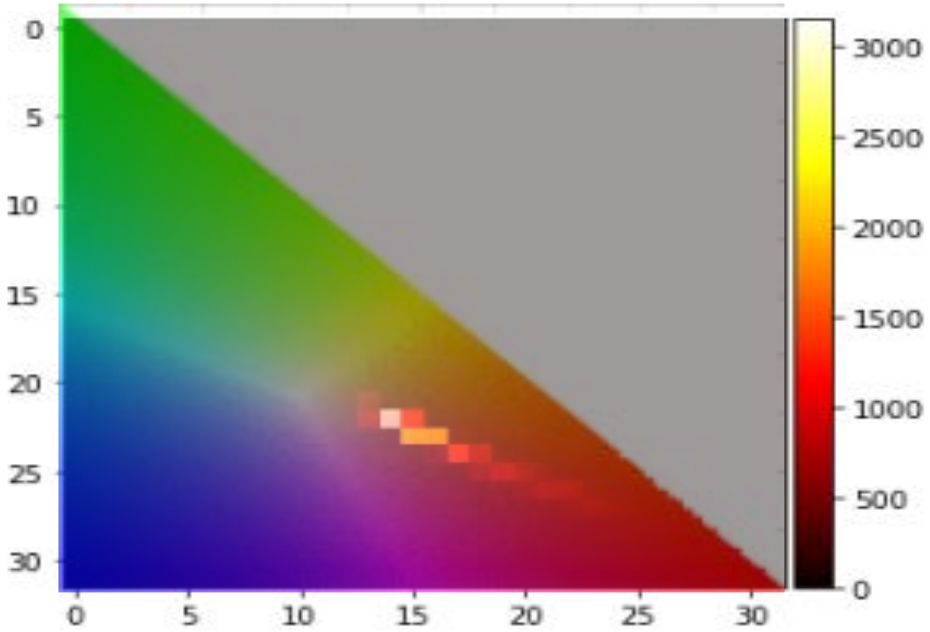
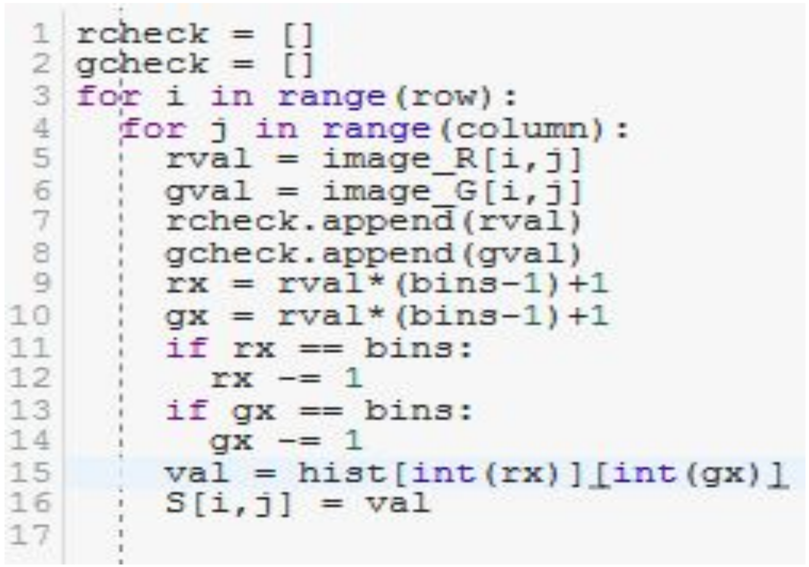


Figure 12.0 Generated 2D histogram of the ROI using 32 bins. Overlaying this with the NCC color space, we can verify that the resulting histogram depicts the chromaticity (light pink to dark red) of our selected ROI.

Image Segmentation via Parametric Probability Distribution Estimation

Using histogram backprojection, we project the corresponding histogram values to our image as to reproduce the resulting segmented image. The following snippet of code was used to backproject the matrix of the 2D histogram to our image.



```python
rcheck = []
gcheck = []
for i in range(row):
    for j in range(column):
        rval = image_R[i,j]
        gval = image_G[i,j]
        rcheck.append(rval)
        gcheck.append(gval)
        rx = rval*(bins-1)+1
        gx = rval*(bins-1)+1
        if rx == bins:
            rx -= 1
        if gx == bins:
            gx -= 1
        val = hist[int(rx)][int(gx)]
        S[i,j] = val
```

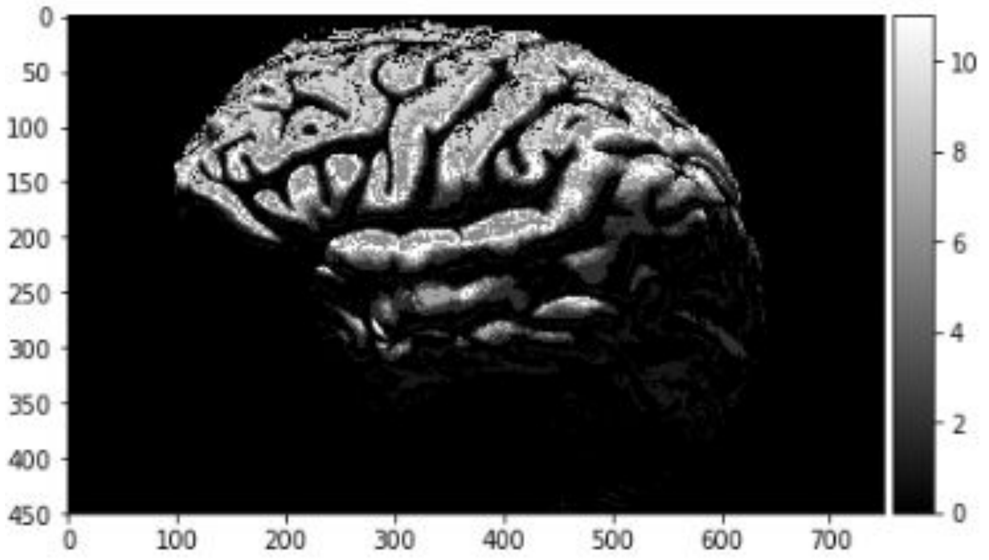


Figure 13.0 Segmented image using non-parametric probability distribution estimation with a bin size of 32.

Image Segmentation via Non-parametric Probability Distribution Estimation

As a means for exploration, I performed the segmentation multiple times using varying sizes of ROIs. The following images are the results of each trial.

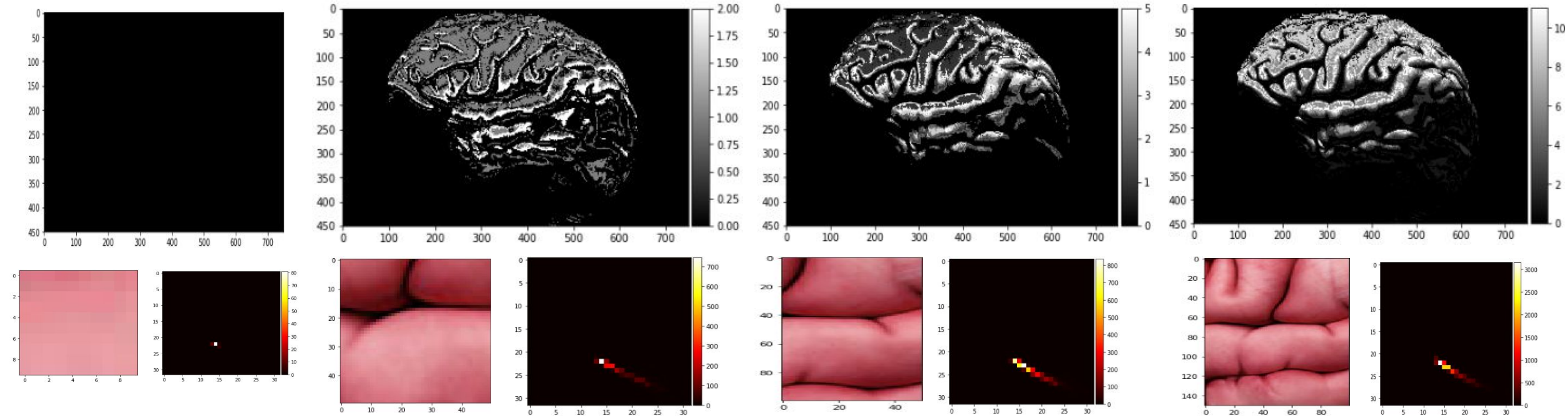
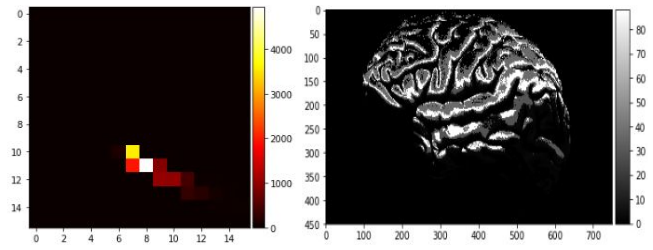


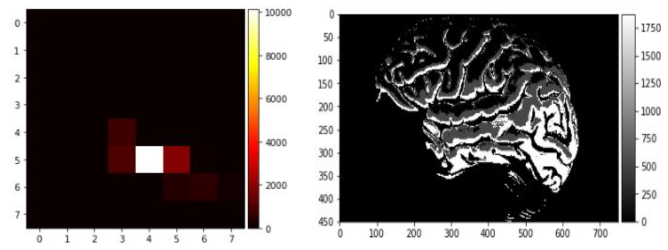
Figure 14.0 Segmented images using non-parametric probability distribution estimation with a bin size of 32. A darker segmentation is observed for increasing amounts of darker shades present within the ROI, while the opposite is true for lighter shades. The segmented image on the extreme right was able to isolate an equal amount of bright and darker shades as its ROI was an equal balance of the two as seen in its corresponding 2D histogram. It was also able to isolate the bottom portion of the image as it does not fit the desired chromaticity which I desire.

Image Segmentation via Non-parametric Probability Distribution Estimation

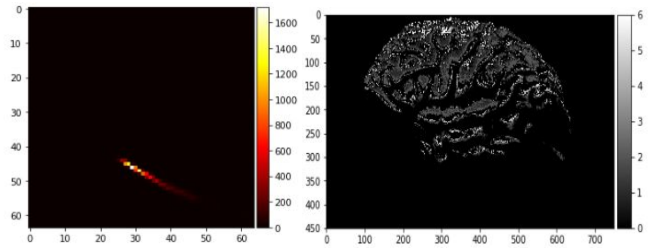
As an additional means for exploration, I also performed the segmentation using varying bin sizes as to determine if there is any effect on using smaller or larger number of bins. I performed this on the best image of the non-parametric ROI test.



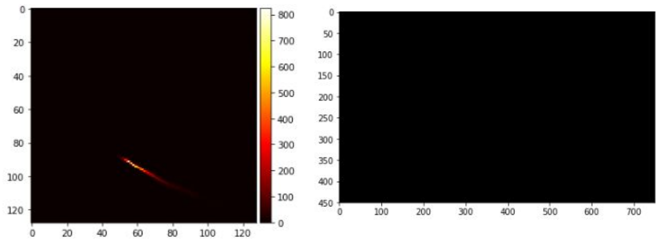
8 bins



16 bins



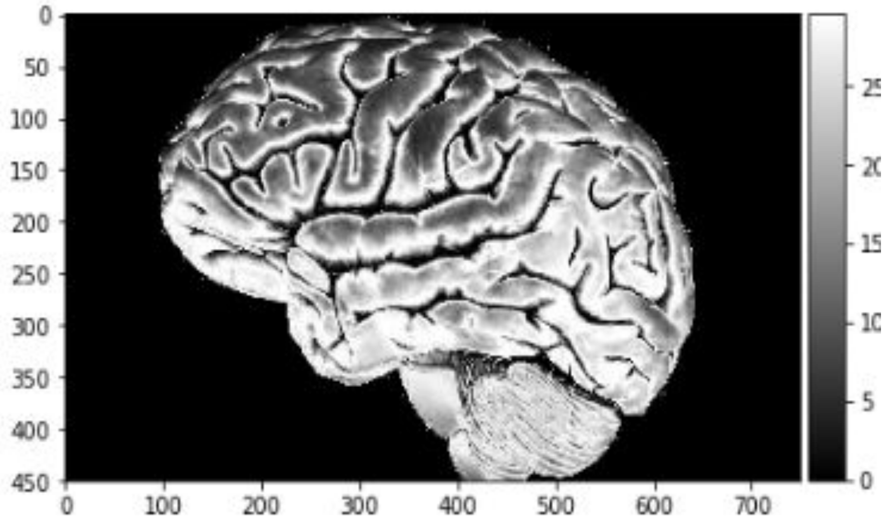
64 bins



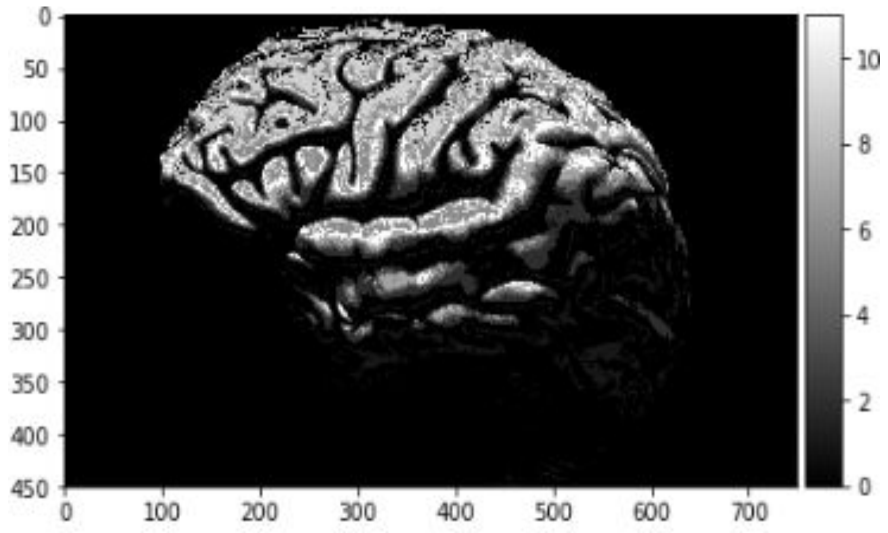
128 bins

Figure 15.0 Segmented images using non-parametric probability distribution estimation with varying bin sizes. As I decreased the amount of bins, one may observe that the corresponding intensities of each 2D histogram scales correspondingly, whereas, increasing the amount of bins would minimize the range at each color gradient covers while it also projects into the segmented image such that each shade of gray now covers a smaller portion of the spectrum. As such, at 128 bins, the probability distribution is too spread out for any image to form.

Image Segmentation (Parametric vs Non-Parametric)



PARAMETRIC



NON- PARAMETRIC

Image Segmentation (Parametric vs Non-Parametric)

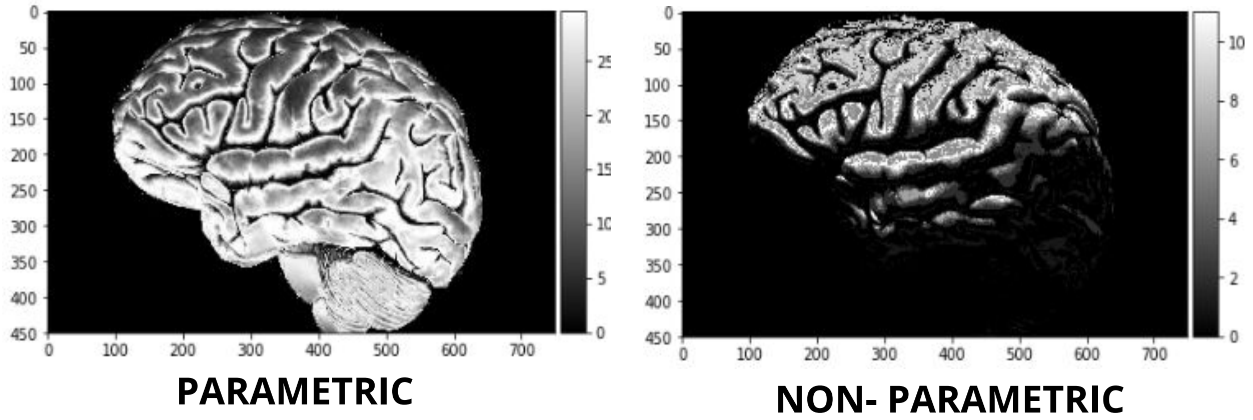


Figure 16.0 Side-by-side comparison of parametric method and non-parametric method for segmentation using a bin size of 32 and the same ROI.

The parametric method for segmenting resulted to a smoother segmentation. This can best observed at the edges between the segmented portions and the unsegmented ones. It also resulted to a brighter segmentation as it was able to estimate the coordinates of all non-background pixels. In contrast, the non-parametric segmentation excelled in specific segmentations - such that it limited the segmentation to strictly shades present within the ROI. As such, I believe that each method has its specific uses. Whereas, the parametric method is best for segmenting a monochromatic object against totally different chromaticities. On the other hand, non-parametric method excels at isolating specific portions of a monochromatic object which is highly dictated by the patch captured by the ROI.

Self-Evaluation

Technical Correctness: 5

Quality of Presentation: 5

Initiative: 2

References

[1]https://www.pinterest.ph/pin/42221933792570471/

[2]Soriano, M., “Activity 7 - Image Segmentation,” 2019.