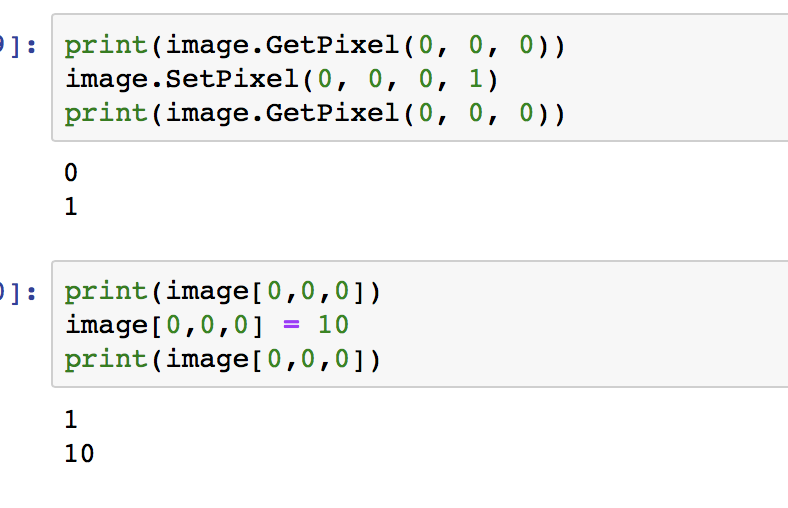
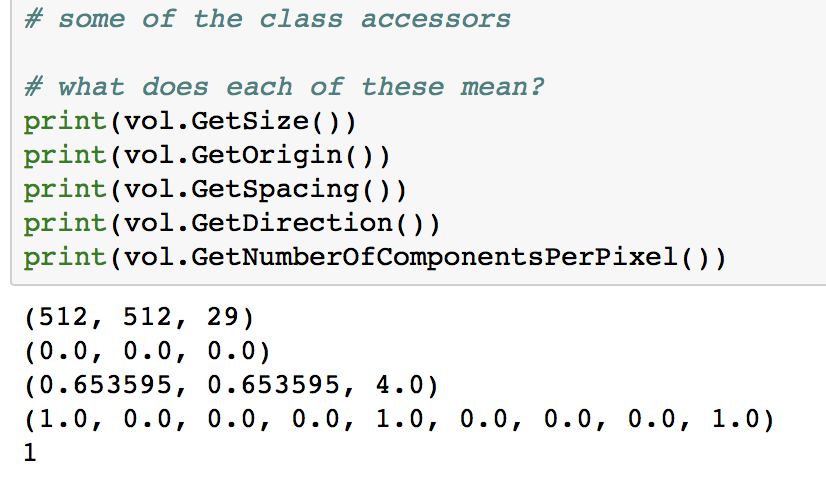
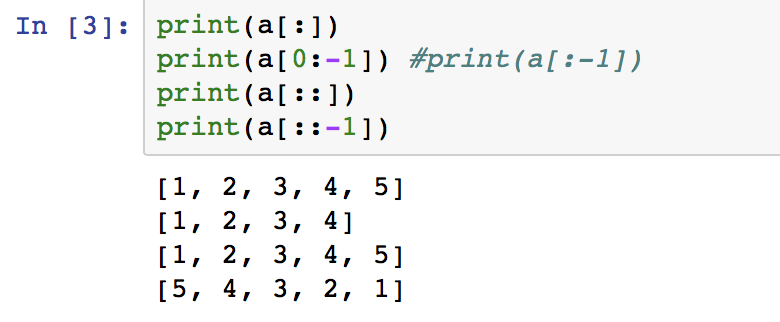
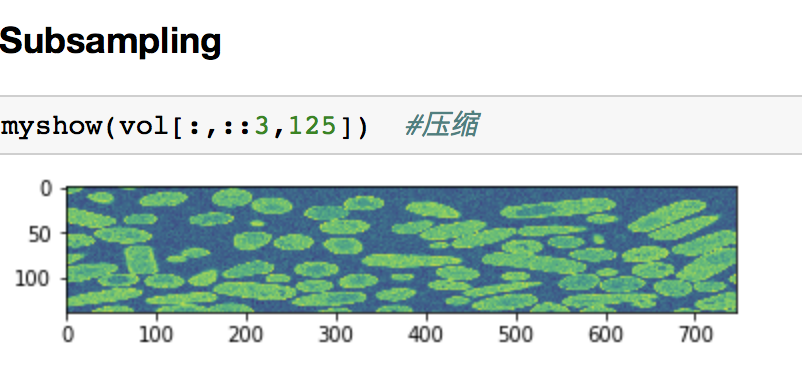
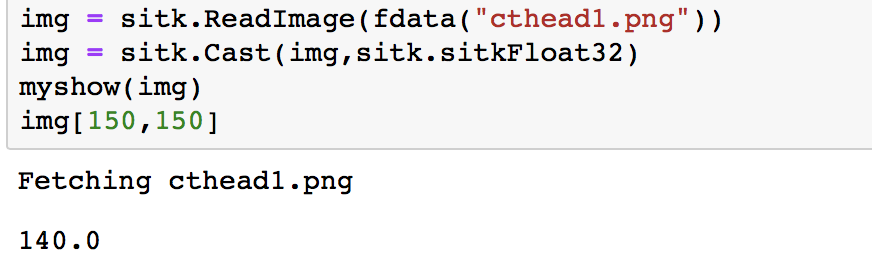
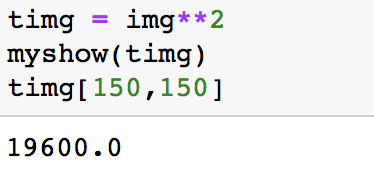
****

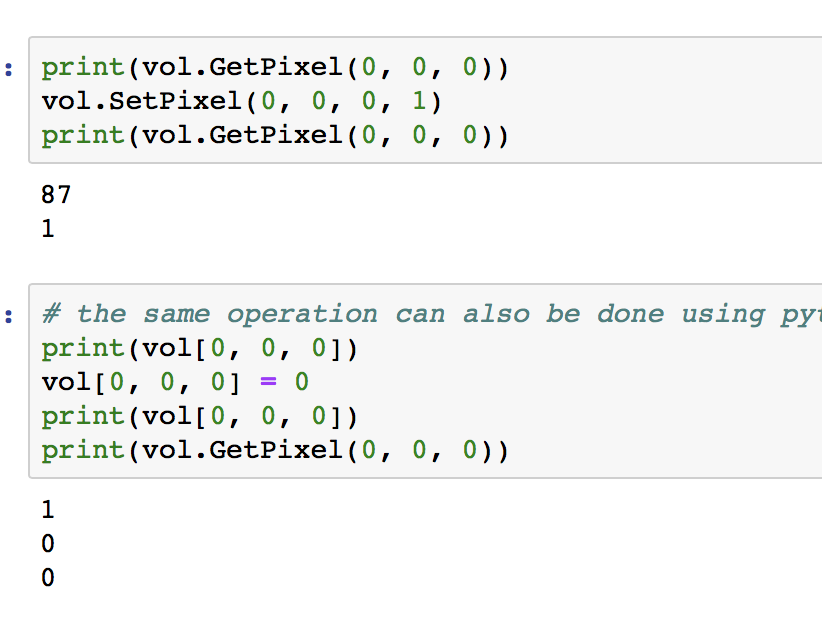
**** ****

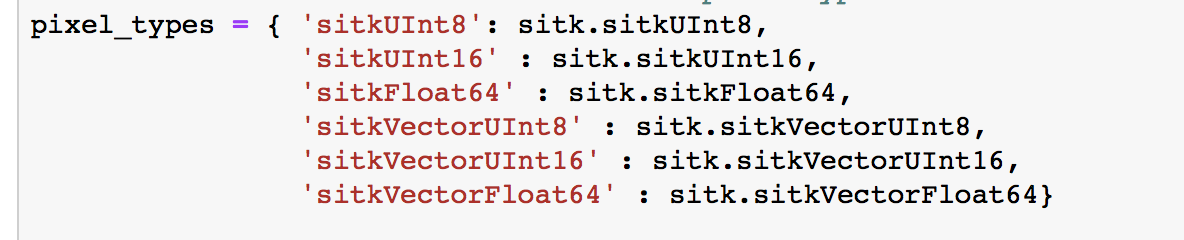
****

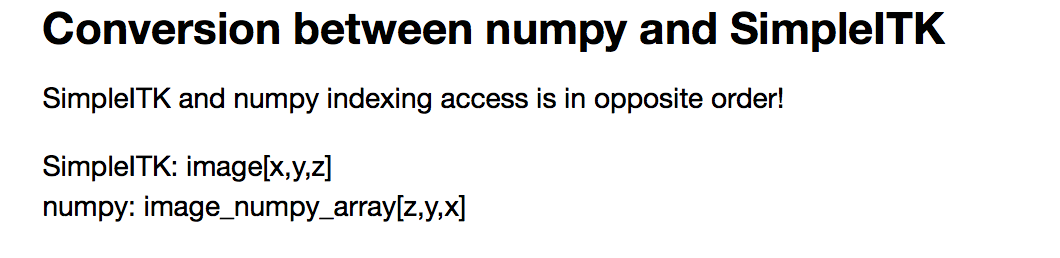
****

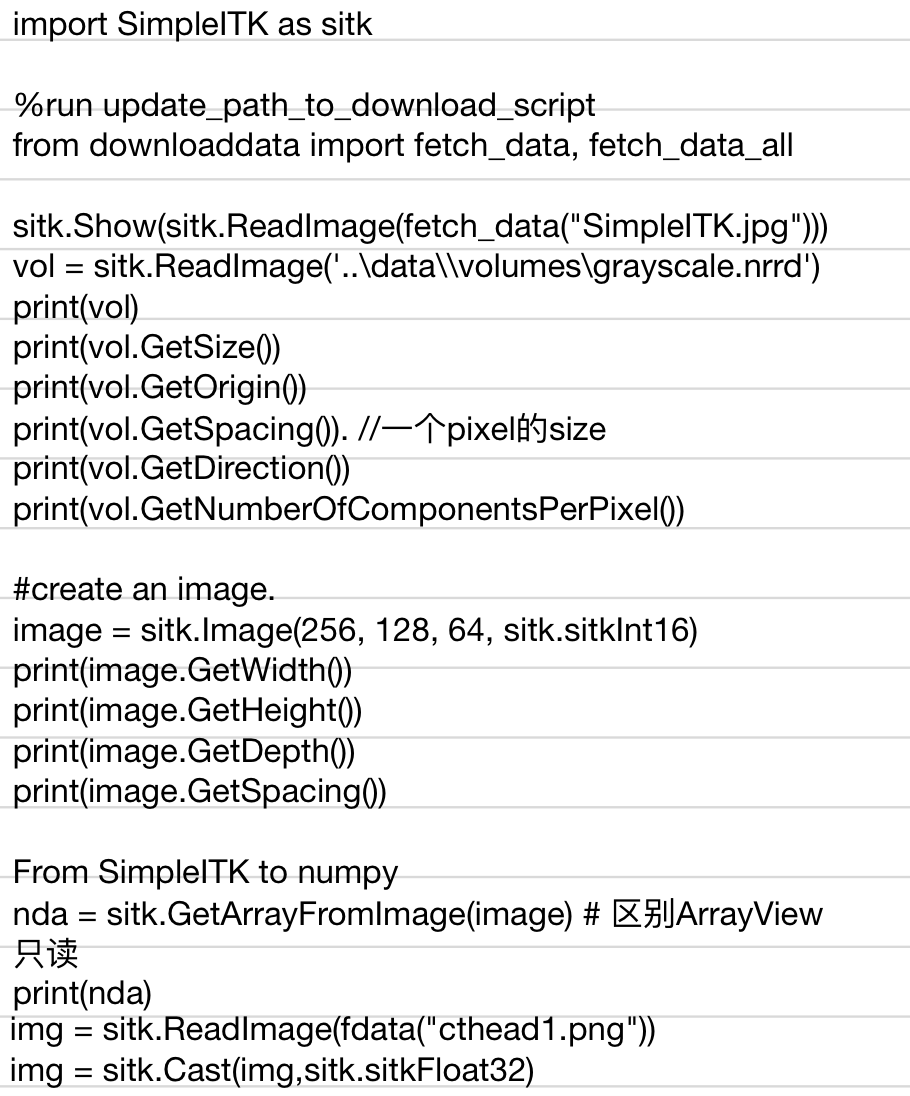
****

**** ****

****



****

****

**writeimage：**

output\_file\_name\_3D = os.path.join(OUTPUT\_DIR, '3DImage.mha')

sitk.WriteImage(original\_image, output\_file\_name\_3D)

**getimagefromarry:**

nda = np.zeros((10,20,3))

#if this is supposed to be a 3D gray scale image [x=3, y=20, z=10]

img = sitk.GetImageFromArray(nda)

fig.add\_subplot(1,3,1)

plt.imshow(npa\_zslice)

plt.title('default colormap')

plt.axis(‘off')

plt.title("Histogram")

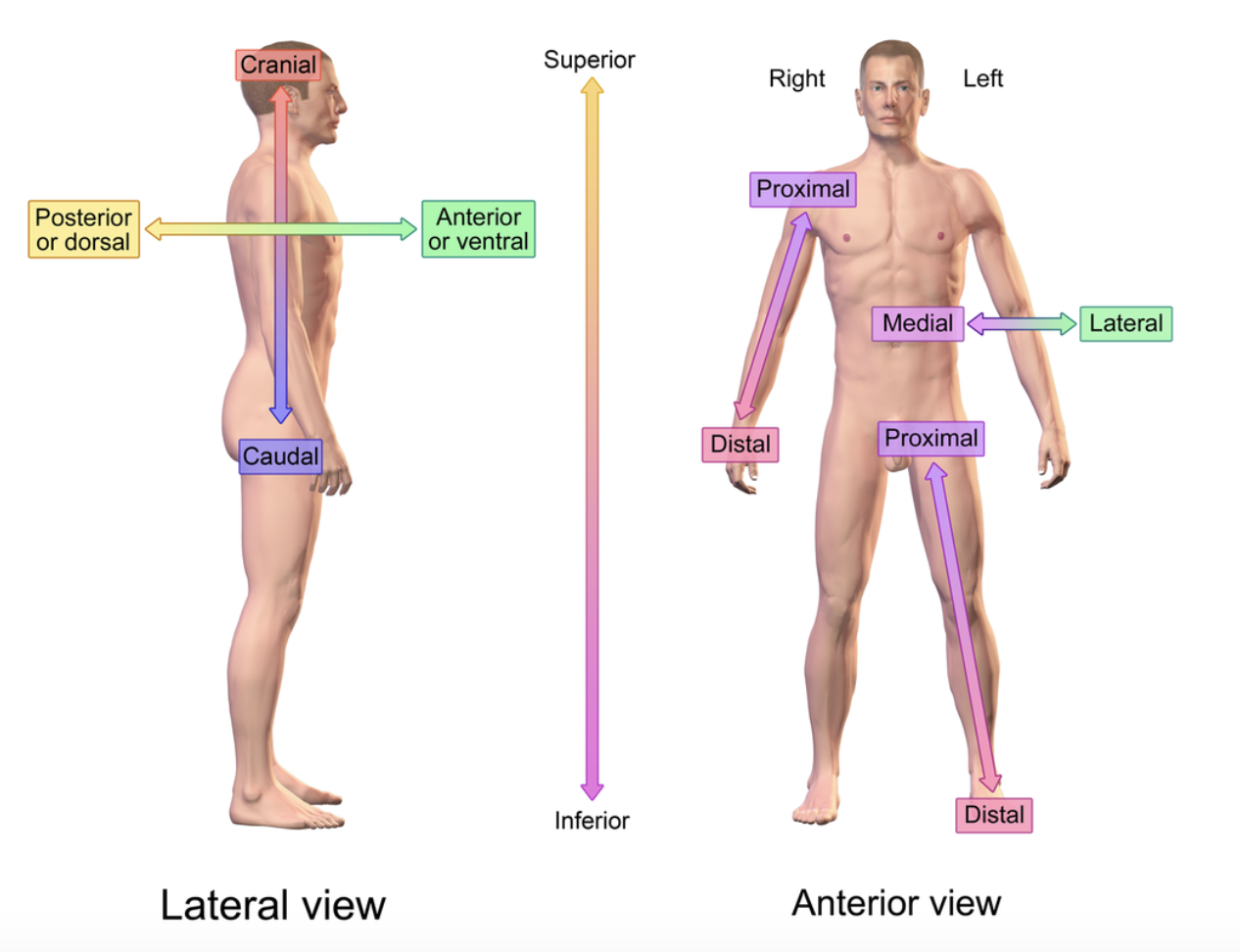
# X label Hounsfield unit (HU) scale, the range is automatically set

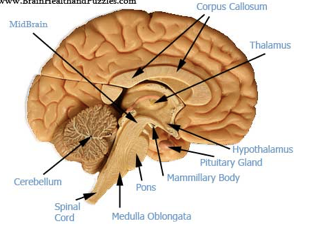
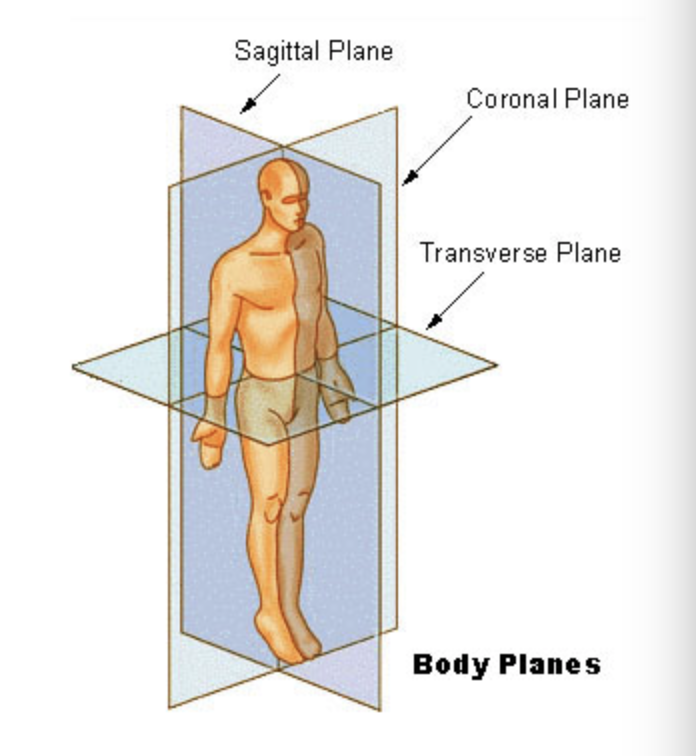
plt.xlabel("Pixel Intensity")

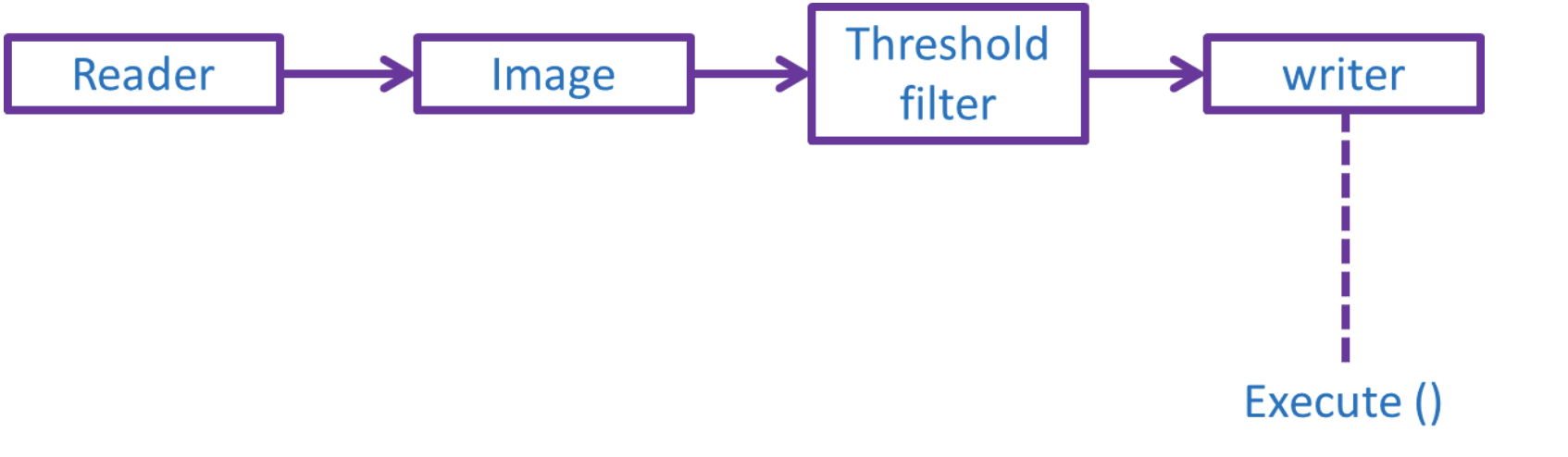
# Y label the number of pixel in this picture slice

plt.ylabel("Occurance")

anatomical position:



 sagittal



plt.imshow(sitk.GetArrayViewFromImage(vol)[z,:,:], cmap=plt.cm.Greys\_r)

plt.figure()

**plt.hist**(sitk.GetArrayViewFromImage(vol)[z,:,:])

plt.title("Histogram")

plt.show()

**matplotlib to display images in our notebooks**

import matplotlib.pyplot as plt

nda = sitk.GetArrayViewFromImage(img1)

plt.imshow(nda)

**Region Growing Segmentation**

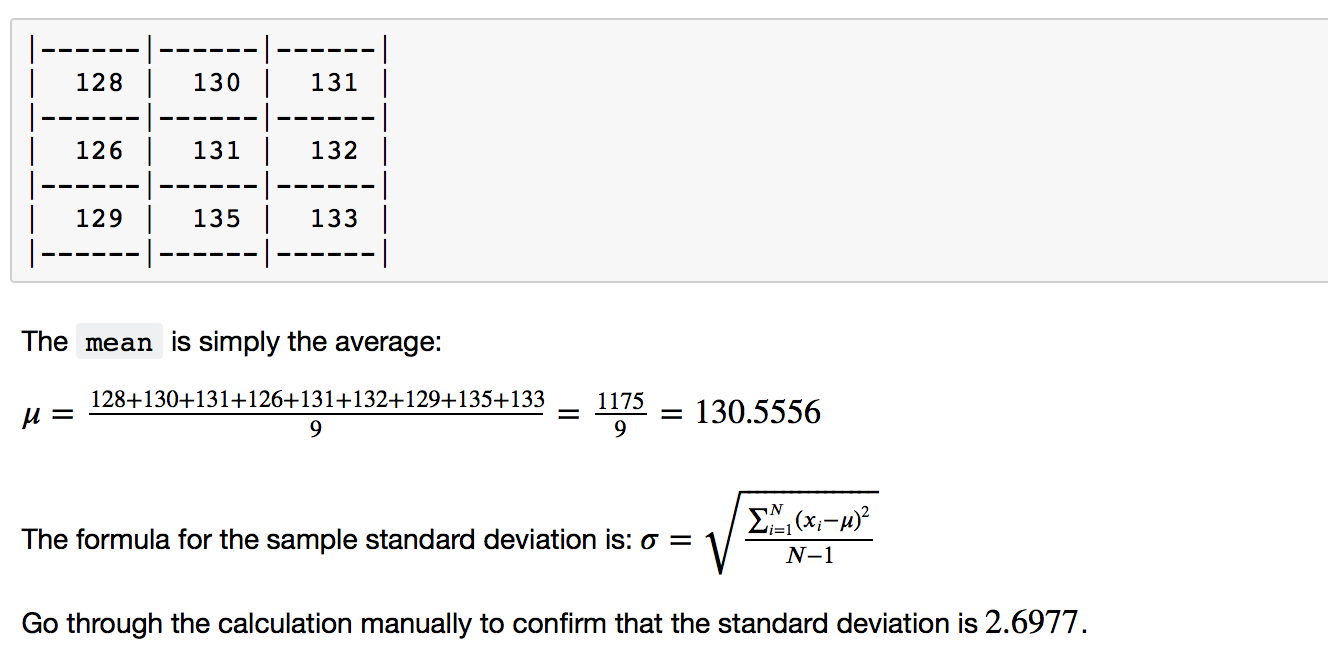
* **ConnectedThreshold**: The neighboring voxel's intensity is within explicitly specified thresholds.

seg\_explicit\_thresholds = sitk.ConnectedThreshold(img\_T1, seedList=initial\_seed\_point\_indexes, lower=120, upper=170)

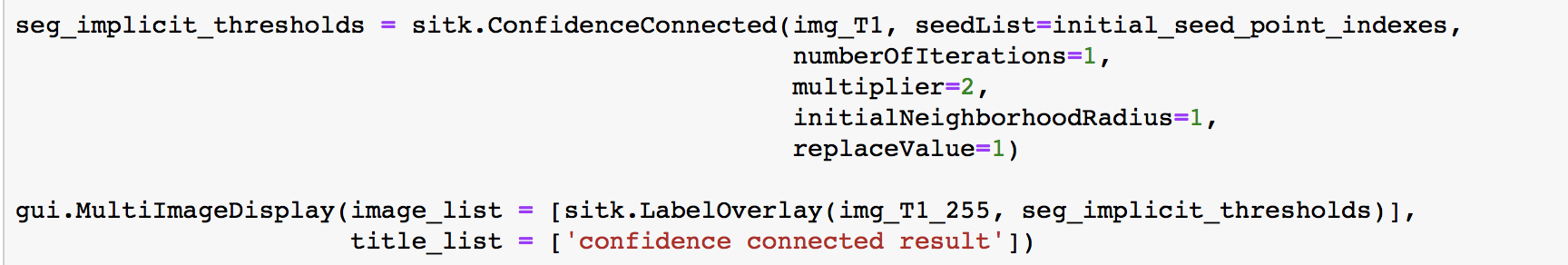
# Overlay the segmentation onto the T1 image

gui.MultiImageDisplay(image\_list = [sitk.LabelOverlay(img\_T1\_255, seg\_explicit\_thresholds)],

title\_list = ['connected threshold result'])



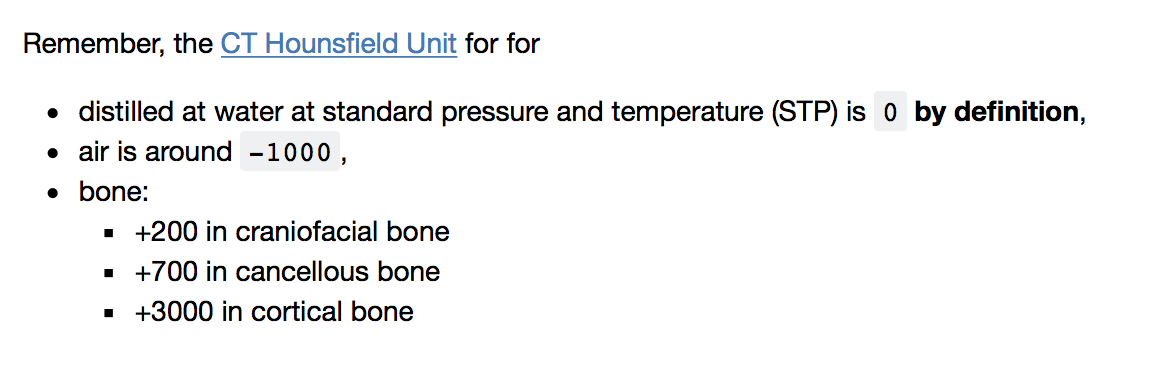
* **ConfidenceConnected**: The neighboring voxel's intensity is within the implicitly specified bounds 𝜇±𝑐𝜎, where 𝜇 is the mean intensity of the seed points, 𝜎 their standard deviation and 𝑐c a user specified constant.
* **Seedlist** -🡪 to calculate the mean and standard deviation, so , don’t keep just one。



* **VectorConfidenceConnected**: A generalization of the previous approach to vector valued images, for instance multi-spectral images or multi-parametric MRI. The neighboring voxel's intensity vector is within the implicitly specified bounds using the Mahalanobis distance where 𝜇μ is the mean of the vectors at the seed points, ΣΣ is the covariance matrix and 𝑐c is a user specified constant.
* ConnectedThresholdImageFilter使用了一种叫做Flood fill iterator的迭代器。在使用这个类时用户要指定像素值的最小值lower threshold和最大值upper threshold，像素值位于lower threshold和upper threshold的像素会被认为是位于种子区域内。

未复习的： vecterconfidenceconnect lecture 6 马氏距离

Thresholding:



img\_255 = sitk.Cast(sitk.RescaleIntensity(img\_ct), sitk.sitkUInt8) #转换单位为255

seg = img\_ct > 220

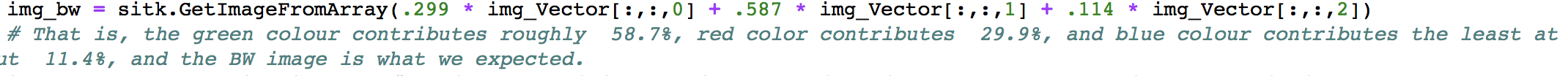
myshow(sitk.LabelOverlay(img\_255, seg), "Basic Thresholding") #标记为绿色

##

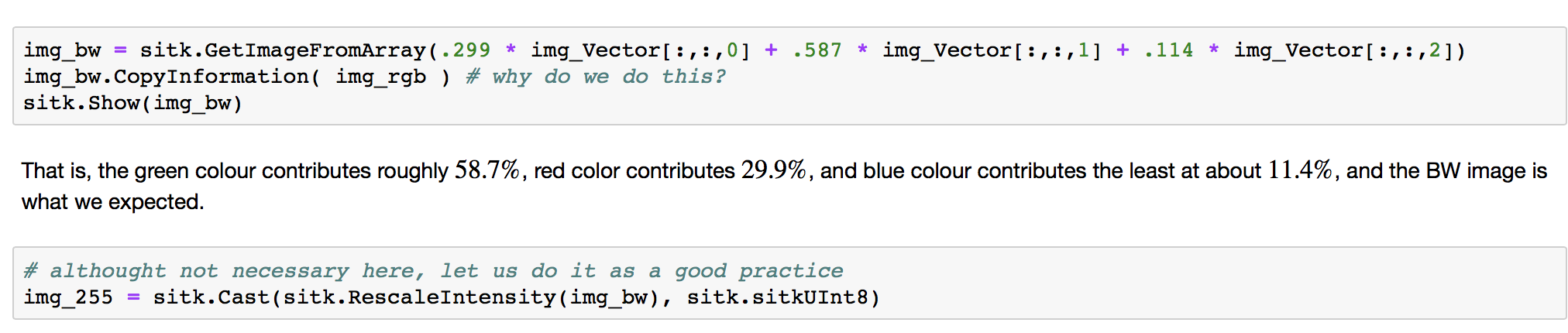
seg = sitk.BinaryThreshold(img\_ct, lowerThreshold=130, upperThreshold=155, insideValue=1, outsideValue=0)

myshow(sitk.LabelOverlay(img\_255, seg), "Binary Thresholding")

##

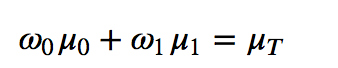
AIP\_08

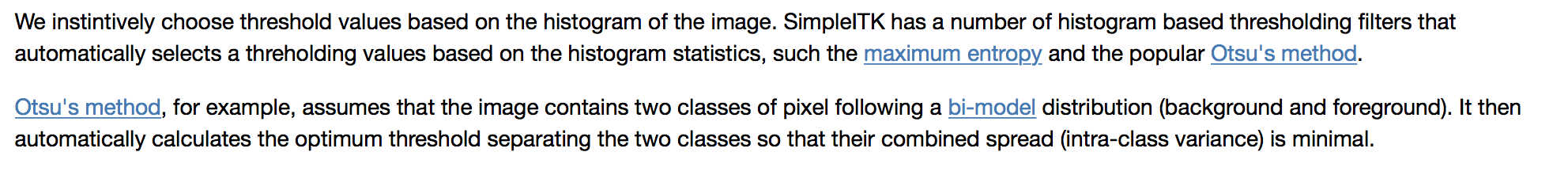
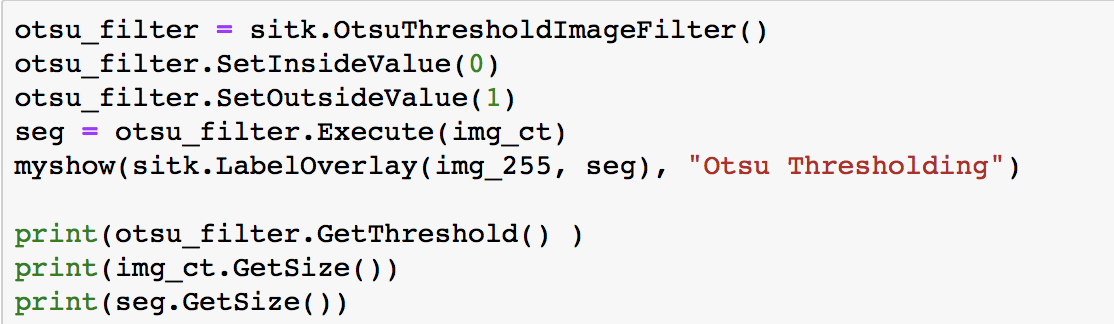
As it turns out, human vision does not preceive each color equally: [Green light contributes the most to the intensity perceived by human eyes, and blue light the least](https://en.wikipedia.org/wiki/Relative_luminance).

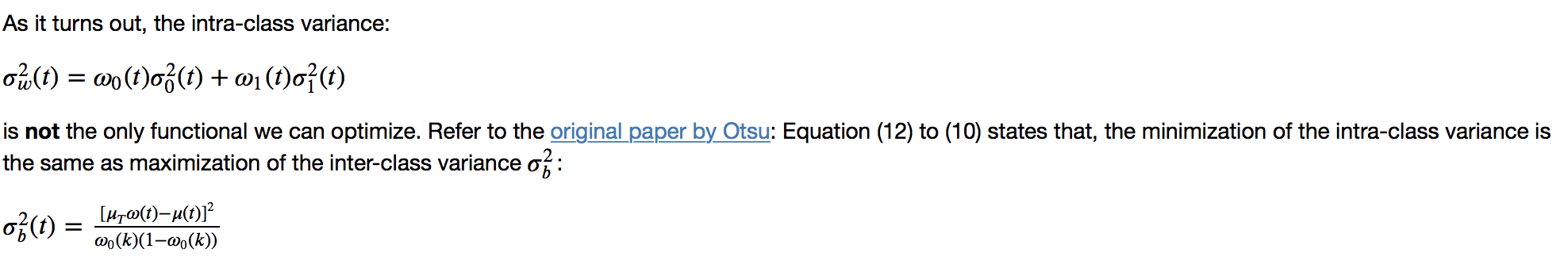


But since we are working with images, and we assumes that the pixel intensity follows a **stochastic process**, the standard deviation/variance computed by **division of**(𝑛)(n) is [biased](https://en.wikipedia.org/wiki/Bias_of_an_estimator): Dividing instead by (𝑛−1)(n−1) yields an unbiased estimator.

* Otsu：对图像Image，记t为前景与背景的分割阈值，前景点数占图像比例为w0，平均灰度为u0；背景点数占图像比例为w1，平均灰度为u1。图像的总平均灰度为：u=w0\*u0+w1\*u1。从最小灰度值到最大灰度值遍历t，当t使得值g=w0\*(u0-u)2+w1\*(u1-u)2 最大时t即为分割的最佳阈值  。对大津法可作如下理解：该式实际上就是类间方差值，阈值t分割出的前景和背景两部分构成了整幅图像，而前景取值u0，概率为w0，背景取值u1，概率为w1，总均值为u，根据方差的定义即得该式。因方差是灰度分布均匀性的一种度量，方差值越大，说明构成图像的两部分差别越大，当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小，因此使类间方差最大的分割意味着错分概率最小。

 （Ut找最大）



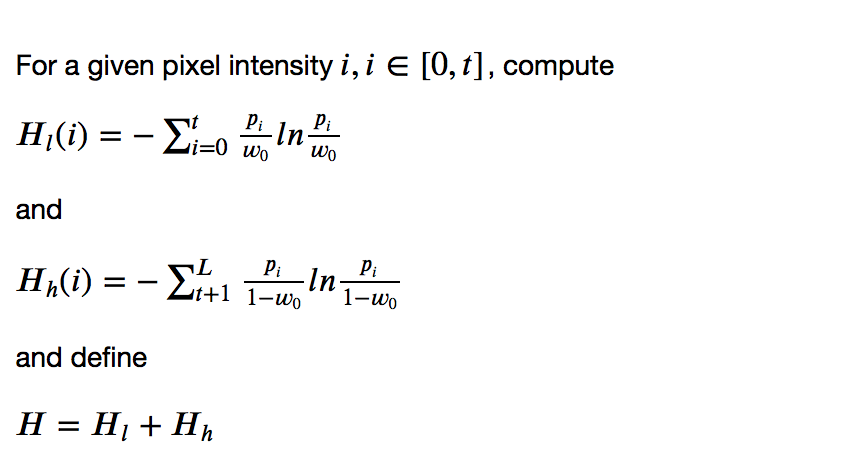


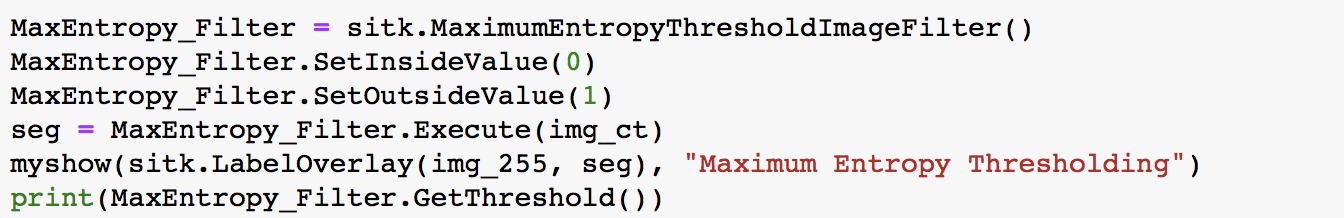
类间（inter-class）方差最大， 类内（intra）方差最小

这个竟然考了

## The maximum entropy method, on the other hand, choose a threhold value such that the entropies of distributions above and below the threshold value is maximised.

That is, 𝐻𝑙(𝑖)Hl(i) is the entropy among the pixels with intensity lower than 𝑖, and 𝐻ℎ(𝑖) is the entropy among the pixels with intensity high than 𝑖. The **maximum entropy** thresholding value 𝐻H is the maxumum of 𝐻𝑙+𝐻ℎ.





7.thresholding localization:？？？？？？？

Let us find a way to determine the exact location for these artificial fiducials (基准点).

We will use two approaches:

1. Segment the fiducial using a thresholding approach, derive the sphere's radius from the segmentation. This approach is solely based on SimpleITK,
2. Localize the fiducial's edge using the Canny edge detector and then fit a sphere to these edges using a least squares approach. This approach is a combination of SimpleITK and scipy/numpy.

要记一下方程如何直接用 Otsa， 还有一个问题 expand 怎么用

8.entropy

MaxEntropy\_Filter = sitk.MaximumEntropyThresholdImageFilter()

MaxEntropy\_Filter.SetInsideValue(1)

MaxEntropy\_Filter.SetOutsideValue(0)

seg = MaxEntropy\_Filter.Execute(img\_bw)

myshow(sitk.LabelOverlay(img\_255, seg), "Maximum Entropy Thresholding")

print(MaxEntropy\_Filter.GetThreshold())

9 计算 以及笔记

4. myshow3d

11.

#erosion

# A trivial example of Erosion

erodeFilter = sitk.BinaryErodeImageFilter()

erodeFilter.SetKernelType ( sitk.sitkCross )

erodeFilter.SetKernelRadius( [1,1]) # adjust the kernal size

img\_eroded = erodeFilter.Execute( seg )

myshow( sitk.Expand(img\_eroded, [2,2]))

# a trivial example of dilation.

dilateFilter = sitk.BinaryDilateImageFilter()

dilateFilter.SetKernelType( sitk.sitkCross )

dilateFilter.SetKernelRadius( [1,1]) # adjust the kernel size

img\_dilated = dilateFilter.Execute(seg)

myshow(sitk.Expand(img\_dilated,[2,2]))

#

# It is obviously bimodal, so Otsu's method should separate black from white easily

otsu\_filter = sitk.OtsuThresholdImageFilter()

inside\_value = 1

outside\_value = 0

otsu\_filter.SetInsideValue(inside\_value)# change into 2, turn blue

otsu\_filter.SetOutsideValue(outside\_value)

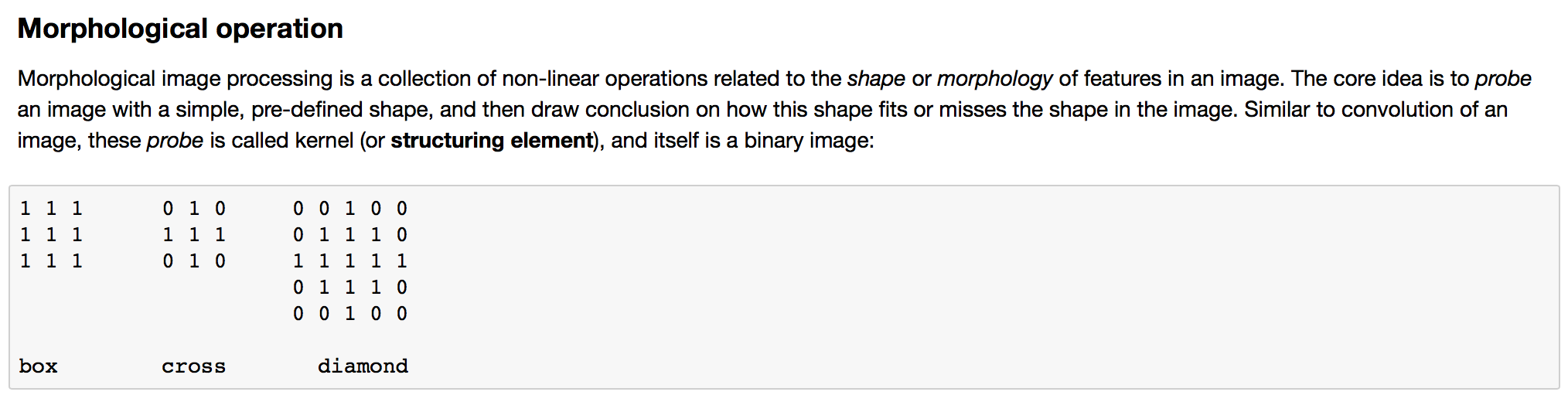
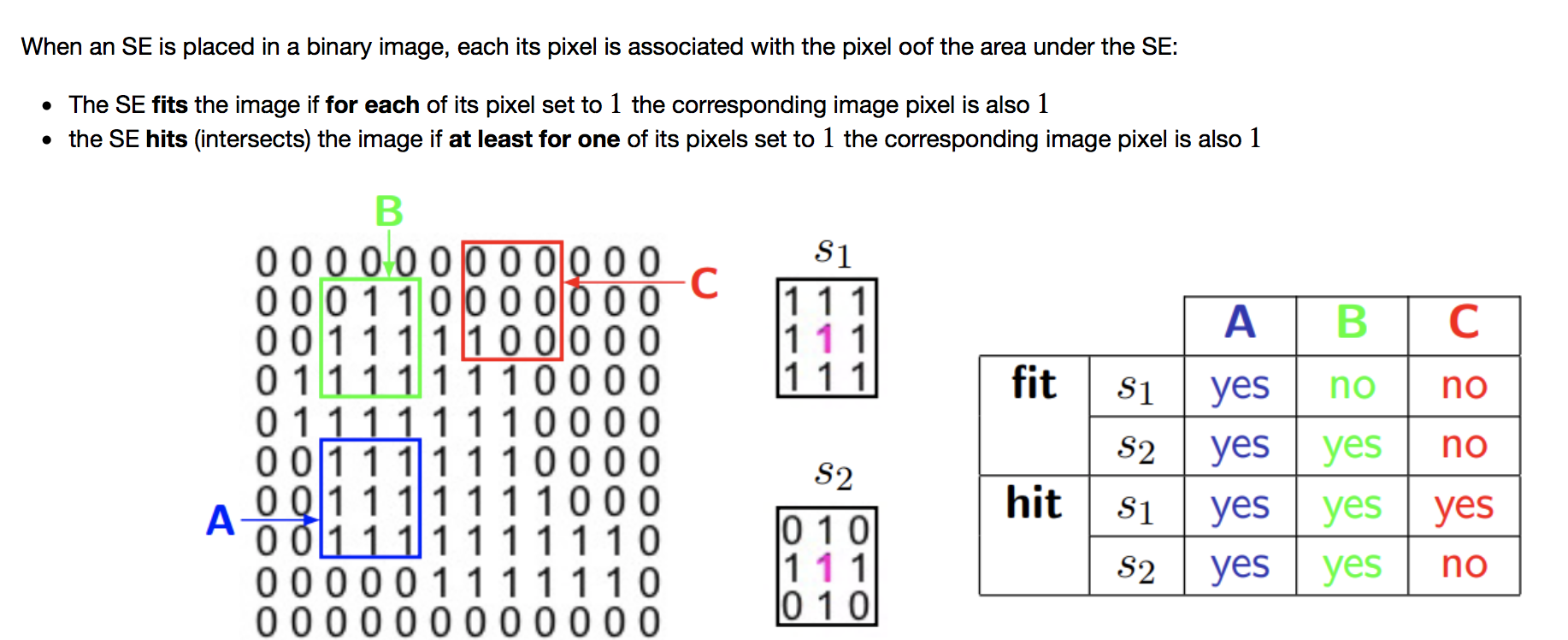
seg = otsu\_filter.Execute(img)

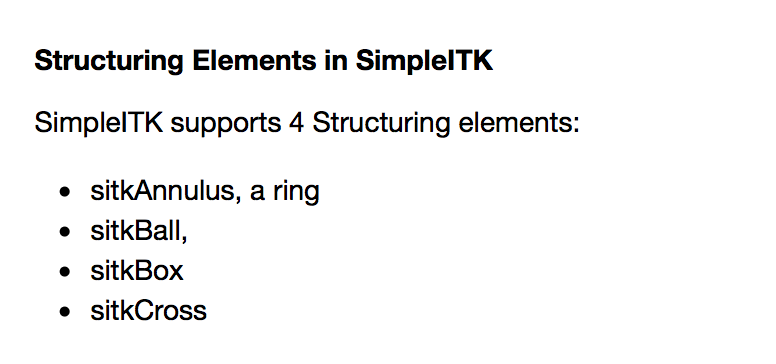
myshow(sitk.LabelOverlay(img, seg), title='Otsu Segmented Image') # the image may be too large to use myshow

print( otsu\_filter.GetThreshold())

# create a binary (0 or 1) image **调整颜色**

img\_bw = img < otsu\_filter.GetThreshold()

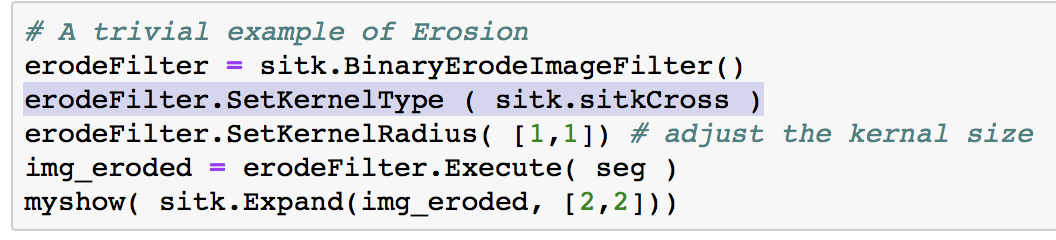
 



如何应用？

dilation—hit

erosion—fit



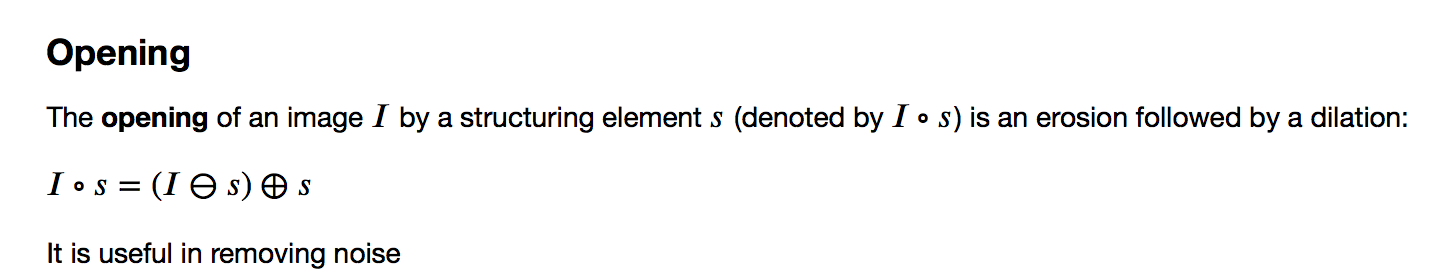
dilateFilter = sitk.BinaryDilateImageFilter()

dilateFilter.SetKernelType( sitk.sitkBall )

dilateFilter.SetKernelRadius( [5,5]) # 5\*5 kernal

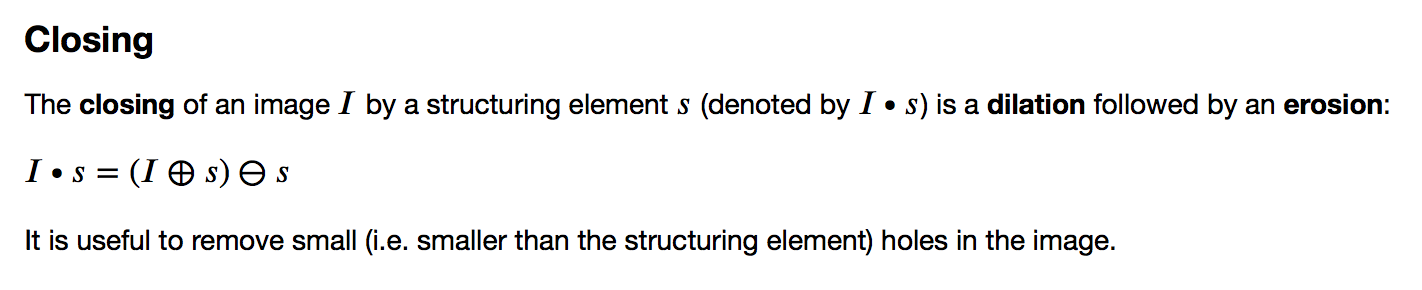
img\_dilated = dilateFilter.Execute(img\_bw)

myshow( img\_dilated )



只要是概念都可能考！

opening, 先削减， 在增长。， 最后是 hole 变为开的



原理， 定义， 方程， 公式