Computer Vision in HealthCare Application Lab-2

Topic: Image Thresholding

Name: Sagnik Chatterjee

Reg.No: 19BAI1153

Dataset used: Skin Cancer HAM10000

Libraries

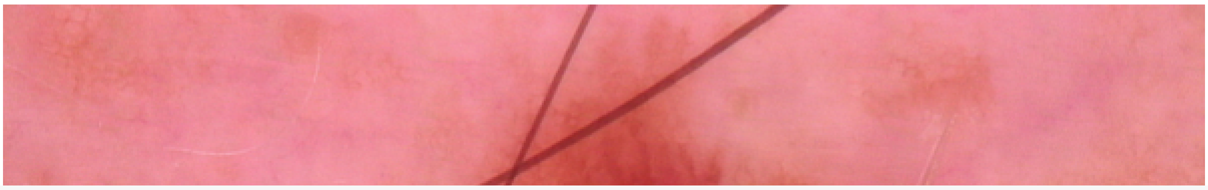
```
import cv2
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
%matplotlib inline
```

First we read Image and convert it to grayscale

Then we add various threshlding techniques to the image

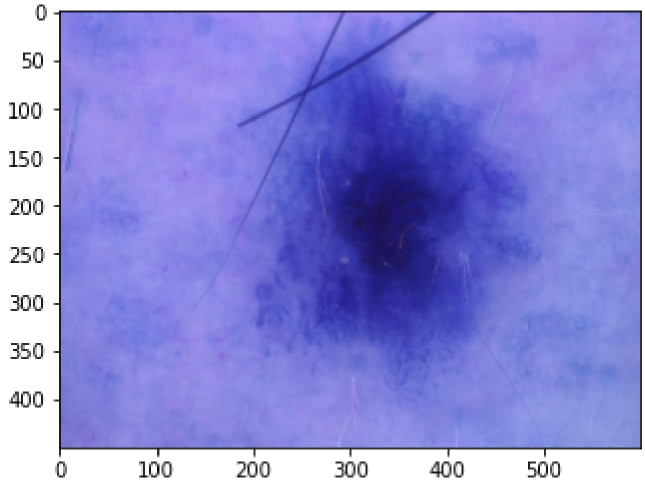
```
img = cv2.imread('/content/ISIC_0024306.jpg')
print("Shape of the image: ",img.shape)
print("Original Image: \n")
cv2_imshow(img)
```

```
Shape of the image:  (450, 600, 3)
Original Image:
```



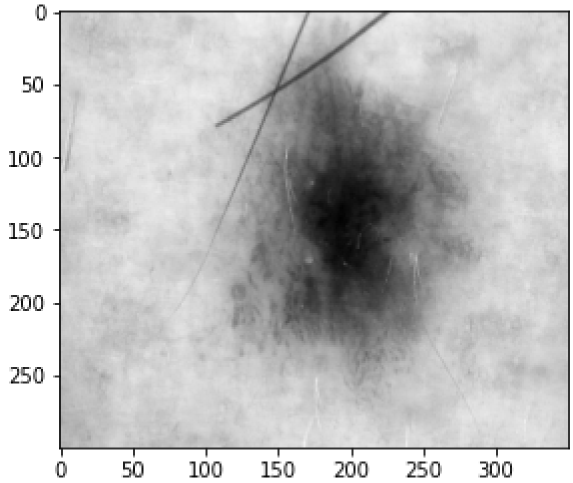
```
img = cv2.imread('/content/ISIC_0024306.jpg')
print(img.shape)
plt.imshow(img,cmap='gray')
```

```
(450, 600, 3)
<matplotlib.image.AxesImage at 0x7f5f46542790>
```



```
img = cv2.resize(img, (350, 300))
img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
print(img.shape)
plt.imshow(img,cmap='gray')
```

```
(300, 350)
<matplotlib.image.AxesImage at 0x7f5f46003710>
```



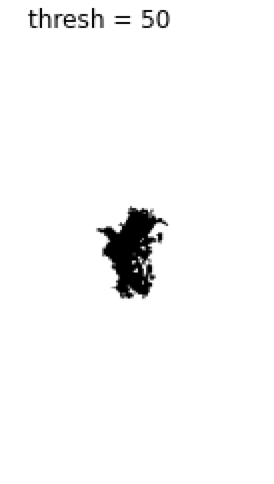
Simple Thresholding

Binary Thresholding

In this binary threshold, the function converts the pixel values to be 0 for values less than the given threshold and 255 for more than threshold.

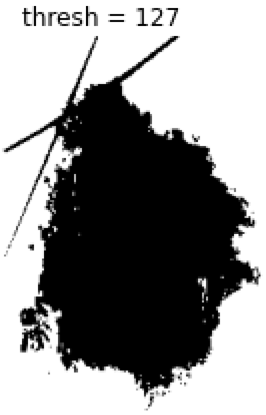
```
_,th1 = cv2.threshold(img,50,255,cv2.THRESH_BINARY)
plt.imshow(th1,cmap='gray')
plt.axis('off')
plt.title('thresh = 50')
```

```
Text(0.5, 1.0, 'thresh = 50')
```



```
_,th2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
plt.imshow(th2,cmap='gray')
plt.axis('off')
plt.title('thresh = 127')
```

```
Text(0.5, 1.0, 'thresh = 127')
```



```
_,th3 = cv2.threshold(img,150,255,cv2.THRESH_BINARY)
plt.imshow(th3,cmap='gray')
plt.axis('off')
plt.title('thresh = 150')
```

```
Text(0.5, 1.0, 'thresh = 150')
```

thresh = 150

```
_,th4 = cv2.threshold(img,200,255,cv2.THRESH_BINARY)
plt.imshow(th4,cmap='gray')
plt.axis('off')
plt.title('thresh = 200')
```

```
Text(0.5, 1.0, 'thresh = 200')
```

thresh = 200

Thus, we can see that the cancer part was kind of in the dark side in original image and when we applied the threshold as 50 the center part of the cancer patch part gets to be entirely black with neighboring space as white. Same kind of went away with 127 but the cancer patch is more darkened with no white patches.The 200 threshold entirely made the image black. 150 darkened the patch abit and kept the background same from the original image by bringing more black spots in the background.

```
def plot_images(th1,th2,th3,th4):
      fig = plt.figure(figsize=(10, 7))
      fig.add_subplot(2, 2, 1)
      plt.imshow(th1,cmap='gray')
      plt.axis('off')
      plt.title('thresh = 50')
      fig.add_subplot(2, 2, 2)
      plt.imshow(th2,cmap='gray')
      plt.axis('off')
```

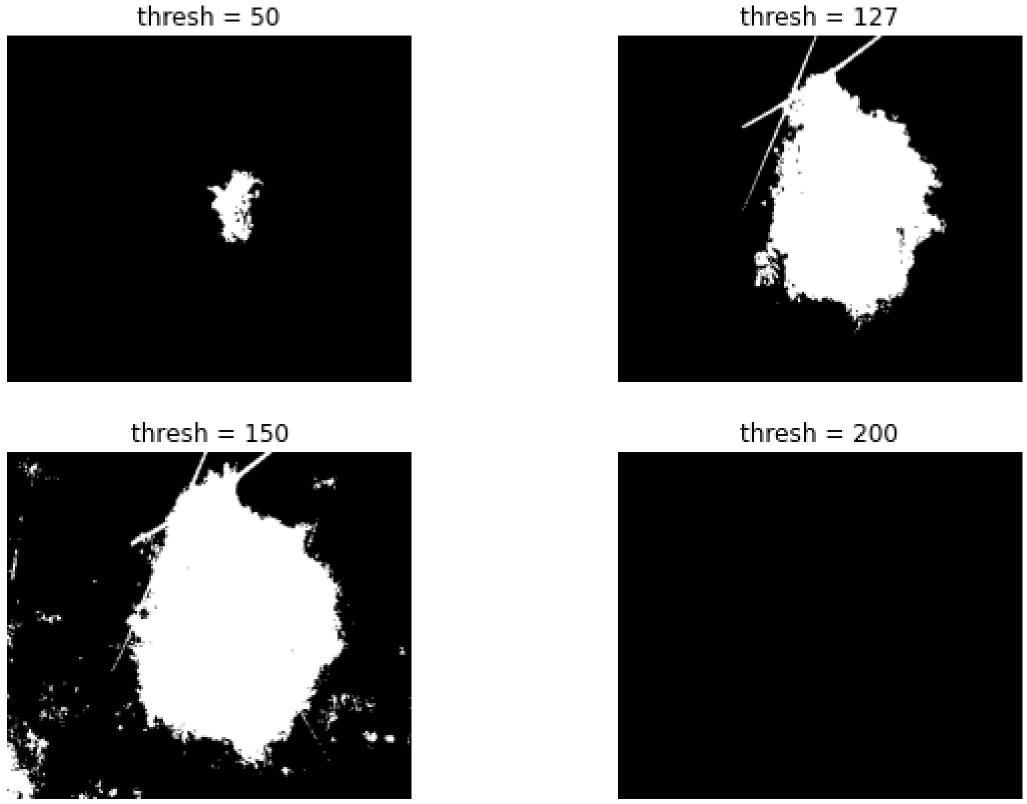
```
    plt.title('thresh = 127')
    fig.add_subplot(2, 2, 3)
    plt.imshow(th3,cmap='gray')
    plt.axis('off')
    plt.title('thresh = 150')
    fig.add_subplot(2, 2, 4)
    plt.imshow(th4,cmap='gray')
    plt.axis('off')
    plt.title('thresh = 200')
```

Binary Inverse Thresholding

The binary inverse thresholding does exactly the inverse of a binary filter

```
_,th1 = cv2.threshold(img,50,255,cv2.THRESH_BINARY_INV)
_,th2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
_,th3 = cv2.threshold(img,150,255,cv2.THRESH_BINARY_INV)
_,th4 = cv2.threshold(img,200,255,cv2.THRESH_BINARY_INV)

plot_images(th1,th2,th3,th4)
```

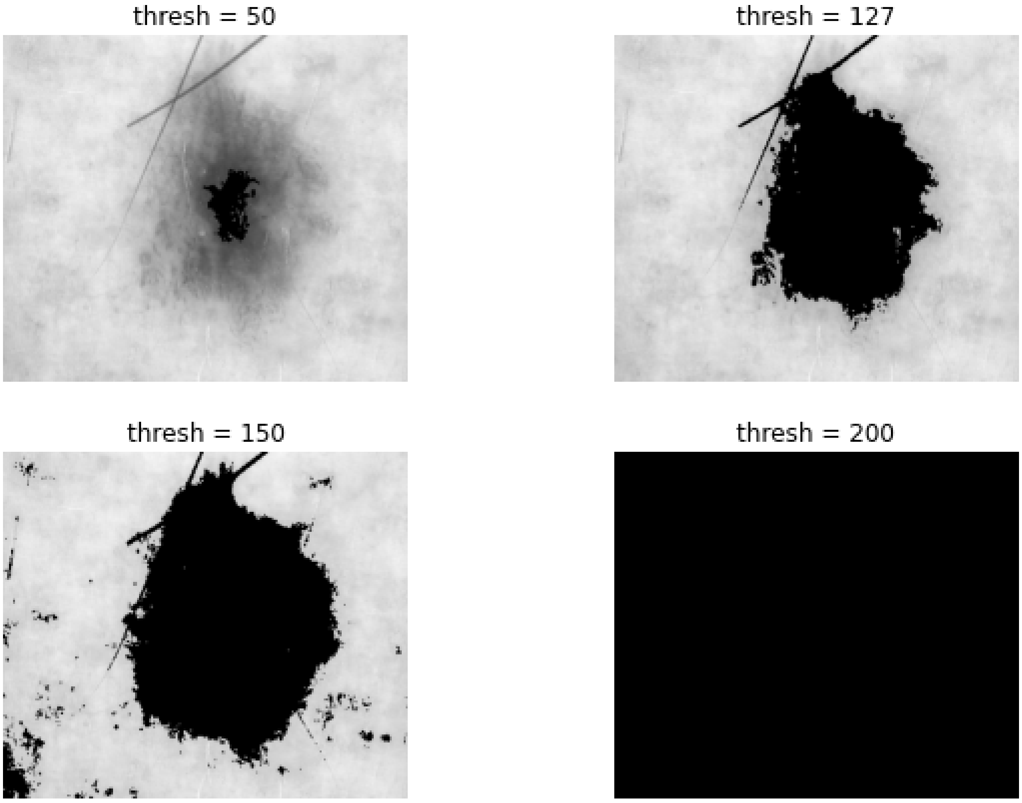


Tozero Thresholding

Tozero thresholding changes the value of the pixel to zero if it is less than the threshold and changes the value to the threshold value if greater.

```
_,th1 = cv2.threshold(img,50,255,cv2.THRESH_TOZERO)
```

```
_,th2 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
_,th3 = cv2.threshold(img,150,255,cv2.THRESH_TOZERO)
_,th4 = cv2.threshold(img,200,255,cv2.THRESH_TOZERO)
plot_images(th1,th2,th3,th4)
```

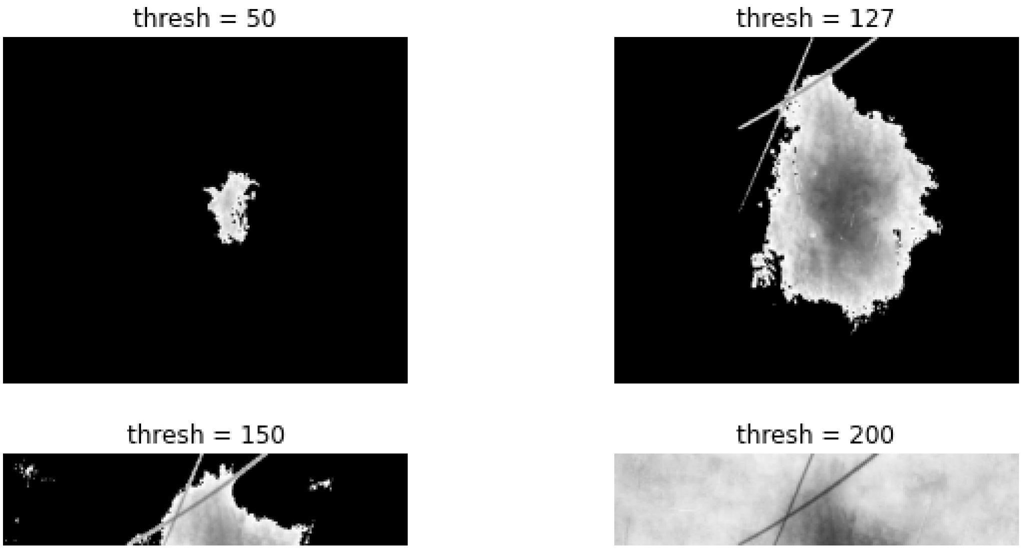


threshold val 50 mostly retained the features of the original image. the other three values does pretty much the same work with little patches of white and lack colors here and there and mostly black. The 200 one is entirely black because as the threshold value increases, the number of pixels with 0 increses.

Inverse Tozero Thresholding

This is the exact iverse of tozero

```
_,th1 = cv2.threshold(img,50,255,cv2.THRESH_TOZERO_INV)
_,th2 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)
_,th3 = cv2.threshold(img,150,255,cv2.THRESH_TOZERO_INV)
_,th4 = cv2.threshold(img,200,255,cv2.THRESH_TOZERO_INV)
plot_images(th1,th2,th3,th4)
```

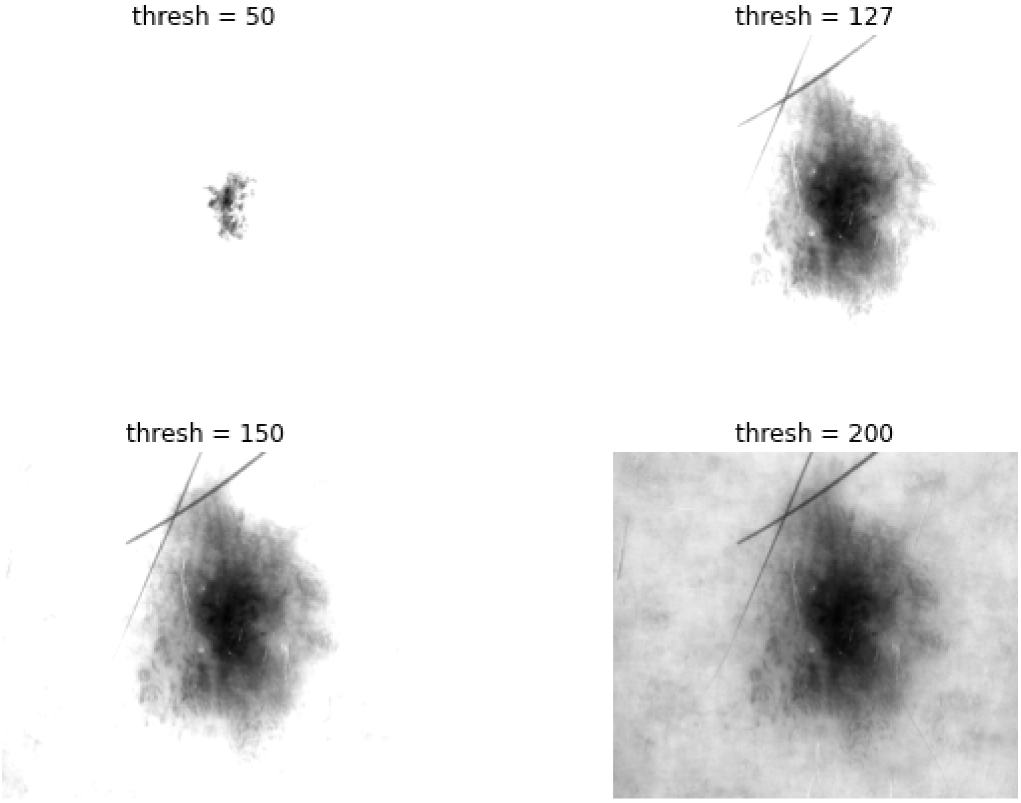


Trunc Thresholding

Trunc Threshold basically keeps the same pixel value if it is less than given threshold and if its is more than the pixel value is changed to threshold value.

```
_,th1 = cv2.threshold(img,50,255,cv2.THRESH_TRUNC)
_,th2 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
_,th3 = cv2.threshold(img,150,255,cv2.THRESH_TRUNC)
_,th4 = cv2.threshold(img,200,255,cv2.THRESH_TRUNC)

plot_images(th1,th2,th3,th4)
```



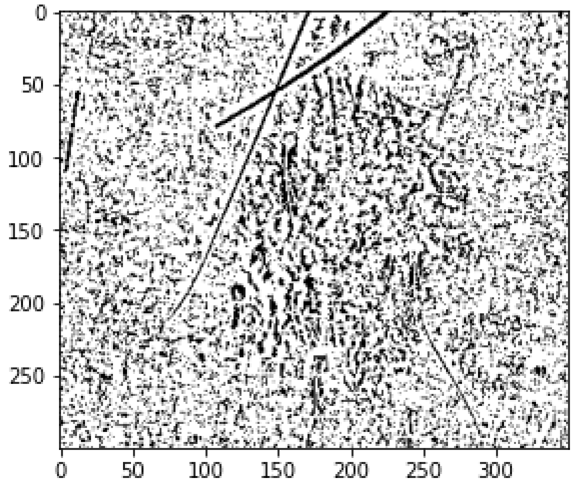
The 50 gives the central cells only that are the darkest ones. Using the thresholding value as 127 we can see the cancer patch more signified in the image with the cancer cells. The 150 threshold value gives more of the original patch. The 200 threshold is giving mostly the original image

Adaptive Thresholding

Gaussian Thresholding

```
th7 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)
plt.imshow(th7,cmap='gray')
```

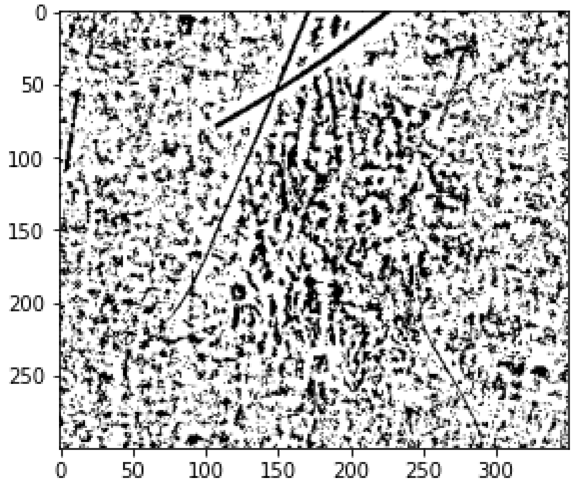
```
<matplotlib.image.AxesImage at 0x7f5f4592cb50>
```



Mean Thresholding

```
th6 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,2)
plt.imshow(th6,cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f5f45bd5e10>
```

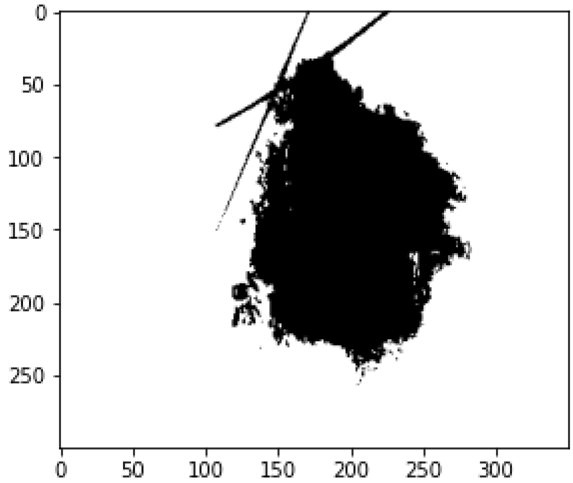


OTSU Binarization

The global threshold value provided by otsu binarization is 109.

```
val, oth = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
plt.imshow(oth,cmap='gray')
print('threshold value = ',val)
```

```
threshold value =  126.0
```



Conclusion

In this lab we applied different thresholding techniques to an image from Skin Cancer HAM10000 dataset. The simple thresholding techniques helped to signify the cancer cell parts in the images whereas the adaptive thresholding technique helped to signify the detailed cells part of the entire image.

✓ 0s completed at 11:49 PM ● ×