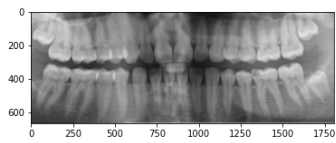


```
In [23]: import cv2
import numpy as np
from matplotlib import pyplot as plt

image=cv2.imread('7.png')
plt.imshow(image, cmap = 'gray', interpolation = 'bicubic')
plt.show()
```



```
In [21]:
```

```
In [20]: import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('7.png')

blur = cv2.blur(img,(25,25))

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(blur),plt.title('Blurred')
plt.xticks([]), plt.yticks([])
plt.show()
```

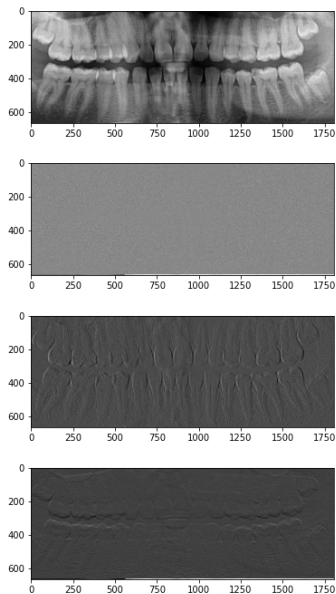


```
In [19]: import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('7.png',0)

laplacian = cv2.Laplacian(img,cv2.CV_64F)
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)

plt.imshow(img,cmap = 'gray')
plt.show()
plt.imshow(laplacian,cmap = 'gray')
plt.show()
plt.imshow(sobelx,cmap = 'gray')
plt.show()
plt.imshow(sobely,cmap = 'gray')
plt.show()
```



```
In [29]: import cv2
import numpy as np
from matplotlib import pyplot as plt

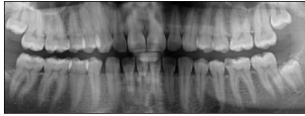
img = cv2.imread('7.png',0)
img = cv2.medianBlur(img,5)

ret,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,\
    cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,\
    cv2.THRESH_BINARY,11,2)

titles = ['Original Image', 'Global Thresholding (v = 127)',
    'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding']
images = [img, th1, th2, th3]

for i in range(4):
    plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
    plt.show()
```

Original Image



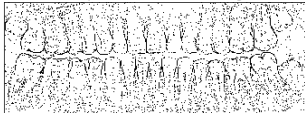
Global Thresholding (v = 127)



Adaptive Mean Thresholding



Adaptive Gaussian Thresholding

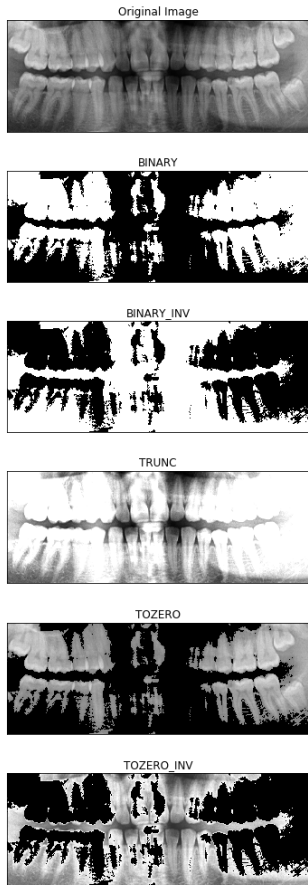


```
In [31]: import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('7.png',0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)

titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in range(6):
    plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
    plt.show()
```



```
In [33]: import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('7.png',0)

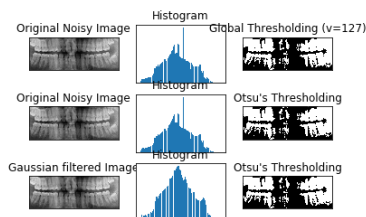
# global thresholding
ret1,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)

# Otsu's thresholding
ret2,th2 = cv2.threshold(img,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

# Otsu's thresholding after Gaussian filtering
blur = cv2.GaussianBlur(img,(5,5),0)
ret3,th3 = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

# plot all the images and their histograms
images = [img, 0, th1,
          img, 0, th2,
          blur, 0, th3]
titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)',
          'Original Noisy Image','Histogram',"Otsu's Thresholding",
          'Gaussian filtered Image','Histogram',"Otsu's Thresholding"]

for i in range(3):
    plt.subplot(3,3,i*3+1),plt.imshow(images[i*3], 'gray')
    plt.title(titles[i*3]), plt.xticks([], plt.yticks([]))
    plt.subplot(3,3,i*3+2),plt.hist(images[i*3].ravel(),256)
    plt.title(titles[i*3+1]), plt.xticks([], plt.yticks([]))
    plt.subplot(3,3,i*3+3),plt.imshow(images[i*3+2], 'gray')
    plt.title(titles[i*3+2]), plt.xticks([], plt.yticks([]))
plt.show()
```



```

In [36]: import numpy as np
import cv2
from matplotlib import pyplot as plt

dst = cv2.imread('7.png')

img = cv2.fastNlMeansDenoisingColored(dst, None, 10, 10, 7, 21)

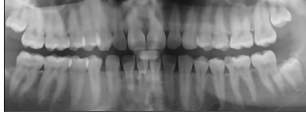
ret, thresh1 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)
ret, thresh2 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY_INV)
ret, thresh3 = cv2.threshold(img, 127, 255, cv2.THRESH_TRUNC)
ret, thresh4 = cv2.threshold(img, 127, 255, cv2.THRESH_TOZERO)
ret, thresh5 = cv2.threshold(img, 127, 255, cv2.THRESH_TOZERO_INV)

titles = ['Original Image', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in range(6):
    plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([], plt.yticks([]))
    plt.show()

```

Original Image



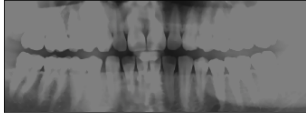
BINARY



BINARY_INV



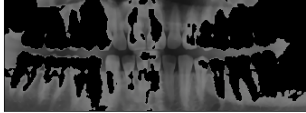
TRUNC



TOZERO



TOZERO_INV



```

In [37]: img = cv2.imread('7.png',0)
img = cv2.medianBlur(img,5)

ret,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,\
    cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,\
    cv2.THRESH_BINARY,11,2)

titles = ['Original Image', 'Global Thresholding (v = 127)',
    'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding']

ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)

titles1 = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images1 = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in range(6):
    plt.imshow(images1[i], 'gray')
    plt.title(titles1[i])
    plt.xticks([],plt.yticks([]))
    plt.show()

ret,thresh1 = cv2.threshold(th1,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(th1,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(th1,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(th1,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(th1,127,255,cv2.THRESH_TOZERO_INV)

titles2 = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images2 = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in range(6):
    plt.imshow(images2[i], 'gray')
    plt.title(titles2[i])
    plt.xticks([],plt.yticks([]))
    plt.show()

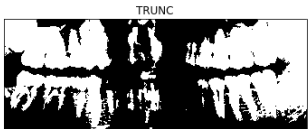
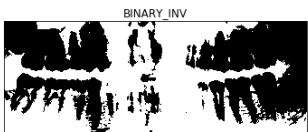
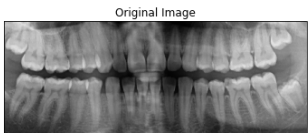
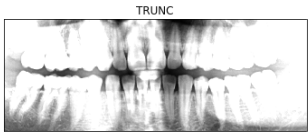
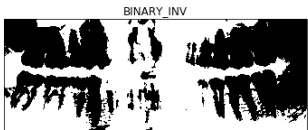
ret,thresh1 = cv2.threshold(th2,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(th2,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(th2,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(th2,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(th2,127,255,cv2.THRESH_TOZERO_INV)

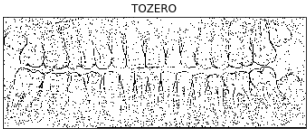
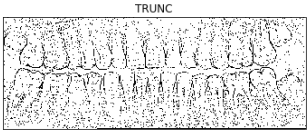
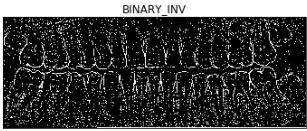
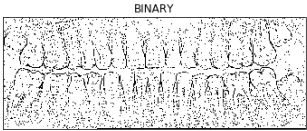
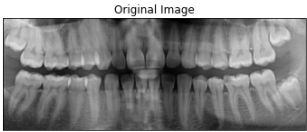
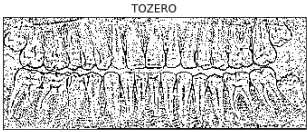
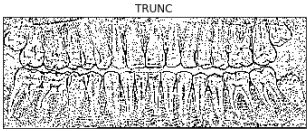
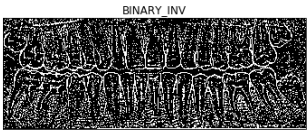
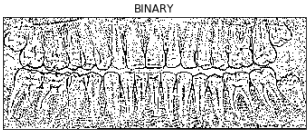
titles3 = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images3 = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in range(6):
    plt.imshow(images3[i], 'gray')
    plt.title(titles3[i])
    plt.xticks([],plt.yticks([]))
    plt.show()

ret,thresh1 = cv2.threshold(th3,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(th3,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(th3,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(th3,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(th3,127,255,cv2.THRESH_TOZERO_INV)

titles4 = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images4 = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in range(6):
    plt.imshow(images4[i], 'gray')
    plt.title(titles4[i])
    plt.xticks([],plt.yticks([]))
    plt.show()

```





In []: