

Computer vision – HW7

A 、 Source code

```
# WeiWen Wu 2023-10-25 14:00:55
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('lena_noise2.jpg')          # Read image.
img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)    # BGR to gray.

n:int=1
def plot(name:str,dst,cmap:str='gray',save:bool=1):    # Make pictures.
    global n                                           # Global variables.
    plt.subplot(2,2,n), plt.imshow(dst,cmap), plt.title(name)
    plt.xticks([]), plt.yticks([]) # Do not show scale.
    if save: cv.imwrite(f"./HW7/{name}.png",dst,[int(cv.IMWRITE_PNG_COMPRESSION),0])
    n+=1

### Otsu's thresholding ###
_,OTSU = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
plot("Otsu's thresholding",OTSU)

### Otsu's thresholding (Gaussian) ### Otsu's thresholding after Gaussian filtering
blur = cv.GaussianBlur(img,(5,5),0)
_,OTSUG = cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
plot("Otsu's thresholding (Gaussian)",OTSUG)

### ADAPTIVE_THRESH_MEAN_C ###
ATMC = cv.adaptiveThreshold(blur,255,cv.ADAPTIVE_THRESH_MEAN_C,cv.THRESH_BINARY,11,2)
plot("ADAPTIVE_THRESH_MEAN_C",ATMC)

### ADAPTIVE_THRESH_GAUSSIAN_C ###
ATGC =
cv.adaptiveThreshold(blur,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.THRESH_BINARY,11,2)
plot("ADAPTIVE_THRESH_GAUSSIAN_C",ATGC)

### Show matplotlib. ###
plt.show()
```

B 、 Result map

Otsu's thresholding



Otsu's thresholding (Gaussian)



ADAPTIVE THRESH MEAN C



ADAPTIVE THRESH GAUSSIAN C



C、 Appendix (original picture)



Figure 1 Otsu's thresholding



Figure 2 Otsu's thresholding (Gaussian)



Figure 3 ADAPTIVE_THRESH_MEAN_C



Figure 4 ADAPTIVE_THRESH_GAUSSIAN_C