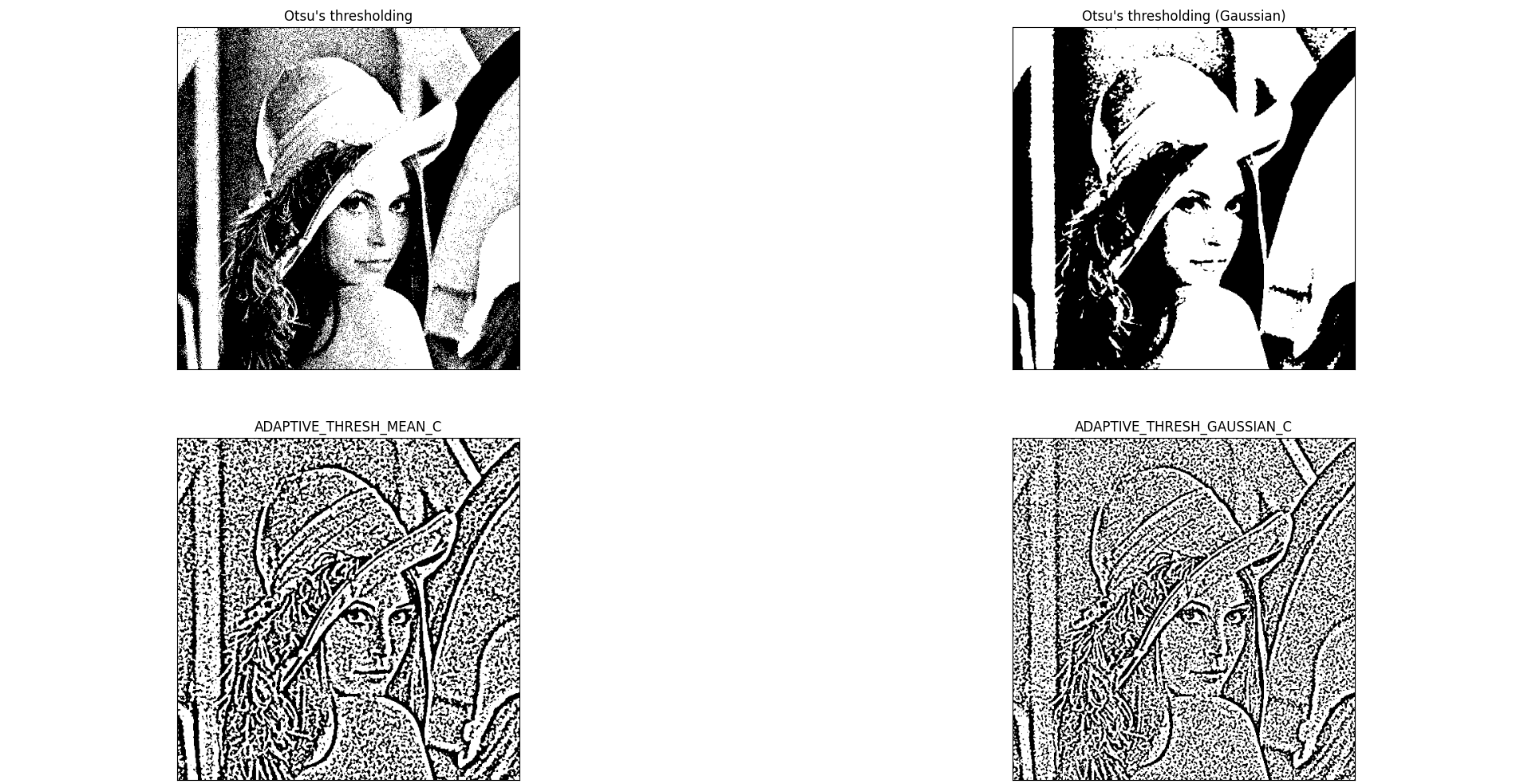
**Computer vision – HW7**

# Source code

|  |
| --- |
| # WeiWen Wu 2023-10-25 14:00:33  **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**()** |

# Result map



# 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