Task 1: Image Thresholding and Binary Operations

Create a Python script that reads an image from the file system using OpenCV.

Implement different thresholding techniques, such as simple thresholding, adaptive thresholding, and Otsu's thresholding, to convert the image to binary.

[1]:
```
import cv2
import numpy as np
```

[17]:
```
img = cv2.imread('cat.jpg')
```

[18]:
```
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

[3]:
```
def show_img(imgs):
    for k, img in imgs.items():
        cv2.imshow(k, img)

    cv2.waitKey(0)
    cv2.destroyAllWindows()
    cv2.waitKey(1)
```

[4]:
```
show_img({'cat':img})
```

[19]:
```
_,simple_thresholding = cv2.threshold(img_gray,127,255,cv2.THRESH_BINARY)
```

[26]:
```
adaptive_thresholding = cv2.adaptiveThreshold(img_gray,255,cv2.ADAPTIVE_THRESH
 cv2.THRESH_BINARY,11,2)
```

[25]:
```
adaptive_gaussian_thresholding = cv2.adaptiveThreshold(img_gray,255,cv2.ADAPTI
 cv2.THRESH_BINARY,11,2)
```

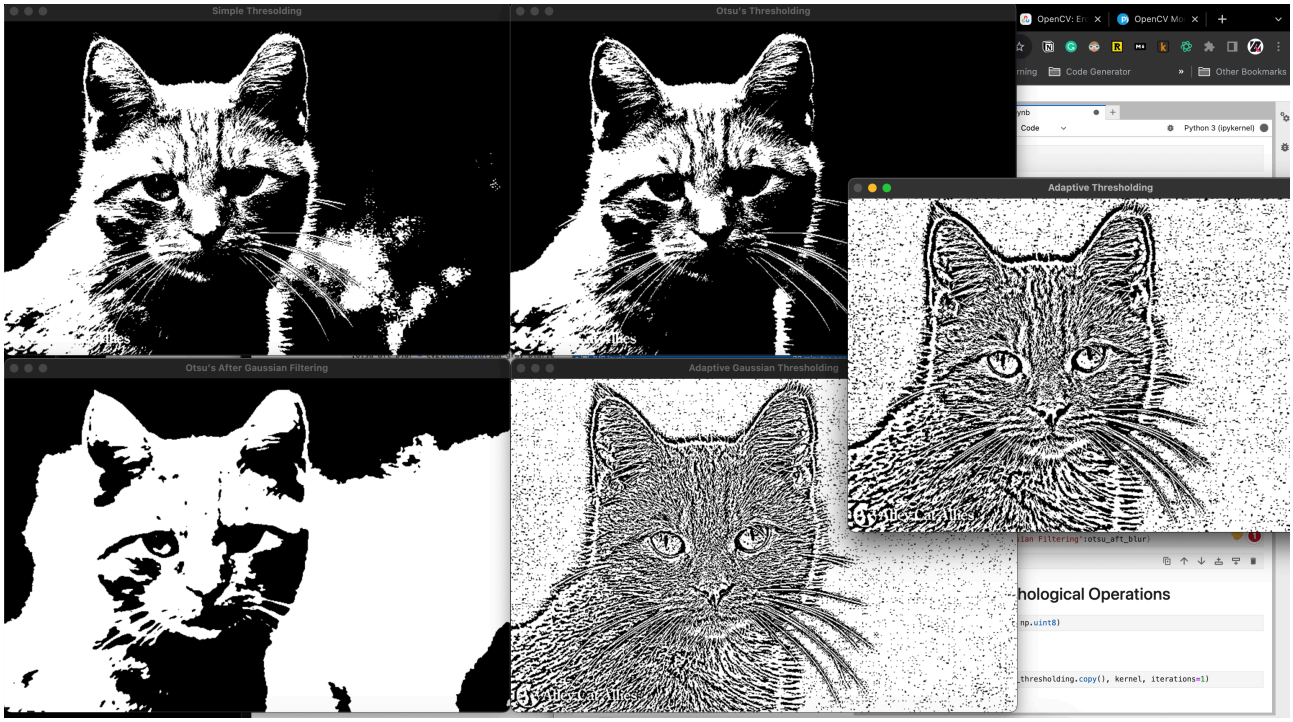
[32]:
```
_, otsu_thresholding = cv2.threshold(img_gray,0,255,cv2.THRESH_BINARY+cv2.THRE
```

[44]:
```
# Otsu's thresholding after Gaussian filtering
img_gray_blur = cv2.GaussianBlur(img_gray,(7,7),0)
_,otsu_aft_blur = cv2.threshold(img_gray_blur,0,255,cv2.THRESH_BINARY+cv2.THRE
```

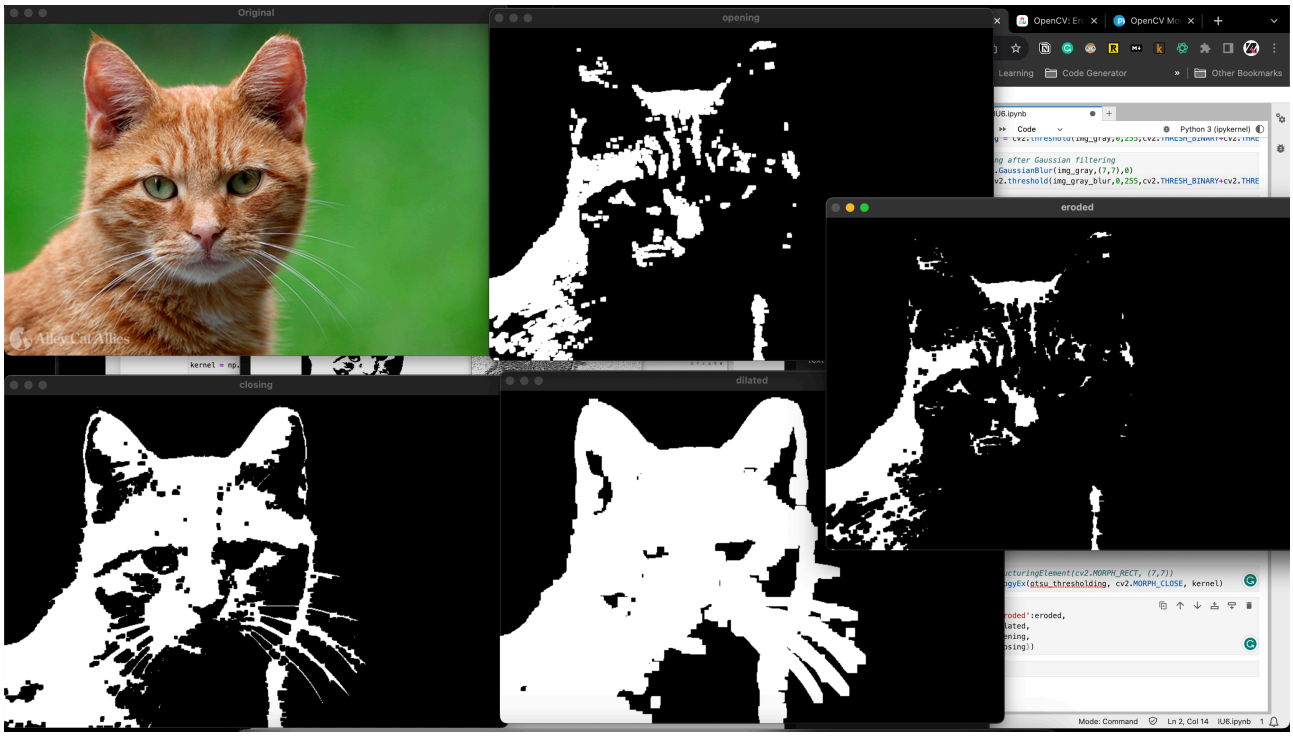
[75]:
```
thresholdings = {
    'Simple Thresolding':simple_thresholding,
    'Adaptive Thresholding': adaptive_thresholding,
    'Adaptive Gaussian Thresholding': adaptive_gaussian_thresholding,
    'Otsu\'s Thresholding':otsu_thresholding,
    'Otsu\'s After Gaussian Filtering':otsu_aft_blur}
```

[76]:
```
show_img(thresholdings)
```

Perform binary operations like erosion, dilation, opening, and closing on the binary image to manipulate and enhance shapes and structures.



Display the original image, the binary image, and the results of binary operations.



OpenCV Morphological Operations

[86]:
```
kernel = np.ones((5,5), np.uint8)
```

Erosion

[87]:
```
eroded = cv2.erode(otsu_thresholding.copy(), kernel, iterations=1)
```

Dilation

[88]:
```
dilated = cv2.dilate(otsu_thresholding.copy(), kernel, iterations=3)
```

Opening

[89]:
```
# kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (7,7))
opening = cv2.morphologyEx(otsu_thresholding, cv2.MORPH_OPEN, kernel)
```

Closing

[90]:
```
# kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (7,7))
closing = cv2.morphologyEx(otsu_thresholding, cv2.MORPH_CLOSE, kernel)
```

[*]:
```
show_img({
    'Original':img,'eroded':eroded,
         'dilated':dilated,
         'opening':opening,
         'closing':closing})
```

[ ]:
```

```