Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Пороговая обработка изображений»

ОТЧЕТ
по лабораторной работе №11
дисциплины
«Технологии распознавания образов»

Выполнил:
Мизин Глеб Егорович
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения

_____
(подпись)

Проверил:
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

ЛР_5(11) Пороговая обработка изображений

Задание 5.1. Провести пороговую обработку полутонового изображения с плавным изменением интенсивности

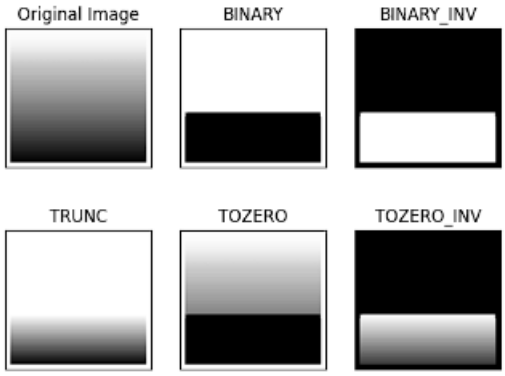
In [1]:
```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
```

In [2]:
```
img = cv.imread ("pic/Grad.jpg" , 0)
ret, thresh1 = cv.threshold (img, 127, 255, cv.THRESH_BINARY)
ret, thresh2 = cv.threshold (img, 127, 255, cv.THRESH_BINARY_INV)
ret, thresh3 = cv.threshold (img, 127, 255, cv.THRESH_TRUNC)
ret, thresh4 = cv.threshold (img, 127, 255, cv.THRESH_TOZERO)
ret, thresh5 = cv.threshold (img, 127, 255, cv.THRESH_TOZERO_INV)

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

for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(title[i])
    plt.xticks([]),plt.yticks([])

plt.show ()
```



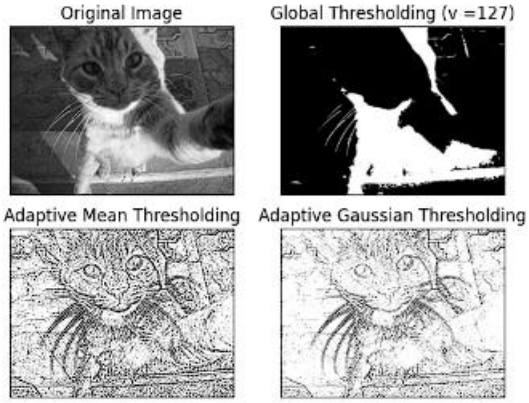
Задание 5.2 Протестировать функции с адаптивным порогом.

In [3]:
```
img = cv.imread('pic/CatWithShadow2.jpg',0)
img = cv.medianBlur(img,5)

ret1,th1 = cv.threshold(img,127,255, cv.THRESH_BINARY)
th2 = cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_MEAN_C,cv.THRESH_BINARY,11,2)
th3 = cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.THRESH_BINARY,11,2)

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

for i in range(4):
    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
```



Задание 5.3 Загрузить дополнительные модули random, PIL. Создать зашумленное изображение.

In [4]:
```
import random
from PIL import Image, ImageDraw
```

In [5]:
```
import random
from PIL import Image, ImageDraw
```

In [6]:
```
image = Image.open('pic/Cat.jpg')
draw = ImageDraw.Draw(image)
width = image.size[0]
height = image.size[1]
pix = image.load()
for i in range(width):
    for j in range(height):
        rand = random.randint(0, 150)
        a = pix[i, j][0] + rand
        b = pix[i, j][1] + rand
        c = pix[i, j][2] + rand
        if (a > 255):
            a = 255
        if (b > 255):
            b = 255
        if (c > 255):
            c = 255
        draw.point((i, j), (a, b, c))
image.save("pic/median.png", "JPEG")


img = cv.imread("pic/median.png", 1)
plt.imshow(cv.cvtColor(img, cv.COLOR_BGR2RGB));
plt.axis('off');
```

Задание 5.4 На вход программы пороговой обработки подается зашумленное изображение. Это изображение обрабатывается тремя способами. В первом случае используется глобальный порог со значением 127. Во втором случае напрямую применяется порог Оцу. В третьем случае изображение сначала удаляет шум фильтром с гауссовым ядром 5х5, затем применяется пороговая обработка Оцу.

Загрузим изображение и сделаем пошаговую обработку

In [7]:
```
img = cv.imread('pic/median.png',0)

ret1, th1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
```

Выполним обработку Оцу

In [8]:
```
ret2, th2 = cv.threshold(img, 0, 255,cv.THRESH_BINARY+cv.THRESH_OTSU)
```

Удалим шум с Гауссовым ядром 5х5 и сделаем обработку Оцу

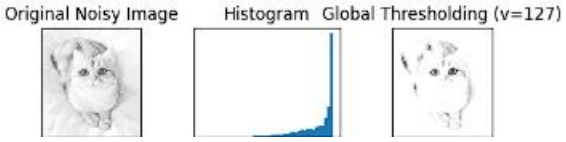
In [9]:
```
blur = cv.GaussianBlur(img,(5,5),0)
ret3, th3 = cv.threshold(blur, 0, 255, cv.THRESH_BINARY+cv.THRESH_OTSU)
```

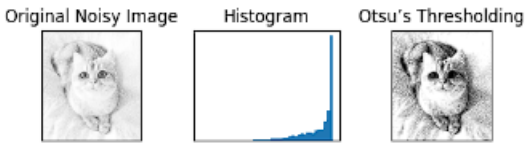
Выведем оригинальное изображение, изображение после обработки и гистограмму обработанного изображения

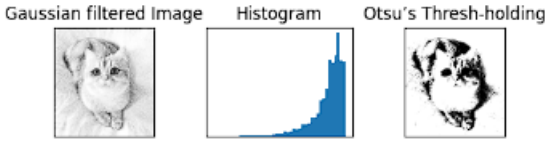
In [10]:
```
images = [img, 0, th1,
          img, 0, th2,
          blur, 0, th3]

titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)','Original Noisy Image', 'Histogram','Otsu's Threshold

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(), 45)
    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()
```







Контрольные вопросы

1. Каким образом происходит получение бинарного изображения?

Бинарное изображение получается путем сравнения интенсивности каждого пикселя с пороговым значением, удаления старого значения интенсивности и присвоения нового значения в зависимости от того, больше или меньше интенсивность пикселя порогового значения.

2. Что происходит с интенсивностью пикселя, если ее значение больше порогового значения?

Если интенсивность пикселя больше порогового значения, то новое значение интенсивности будет равно 255, а при меньшей интенсивности порогового значения - новое значение интенсивности будет равно 0.

3. Для чего в функции cv.threshold(img,127, 255, cv.THRESH) используется третий и четвертый аргументы?

Третий аргумент в функции cv.threshold(img,127, 255, cv.THRESH) задает значение интенсивности на выходе функции, когда значение пикселя больше порогового значения, а четвертый параметр задает используемый тип порогового значения.

4. Чем отличаются функции cv2.adaptiveThreshold с параметром cv2.ADAPTIVE_THRESH_MEAN_C и cv2.ADAPTIVE_THRESH_GAUSSIAN _C?

Функция cv2.adaptiveThreshold с параметром cv2.ADAPTIVE_THRESH_MEAN_C берет в качестве порогового значения среднее арифметическое всех пикселей в окрестности выделенного пикселя, а функция cv2.adaptiveThreshold с параметром cv2.ADAPTIVE_THRESH_GAUSSIAN_C берет взвешенную сумму значений

окрестностей, где весовые коэффициенты определяются с помощью функции Гаусса.

5. Какие аргументы принимают на вход функции cv2.adaptiveThreshold?

На вход функции cv2.adaptiveThreshold необходимо подать исходное изображение, желаемое значение интенсивности на выходе, метод расчета порогового значения (cv2.ADAPTIVE_THRESH_MEAN_C или cv2.ADAPTIVE_THRESH_GAUSSIAN_C), размер окрестности и константу C, используемую для настройки порогового значения.

6. Для чего используется последний аргумент в функциях cv2.adaptiveThreshold и как он помогает настроить пороговое значение?

Последний аргумент - константа C, вычитаемая из вычисленного среднего или взвешенного среднего. Это позволяет точно настроить пороговое значение и сделать его более или менее чувствительным к изменениям в значениях пикселей вокруг выделенного пикселя.

7. Что такое порог и как он помогает разделить изображение на объект и фон?

Порог - это яркостное значение, которое используется для разделения изображения на объект и фон. Объект - это множество пикселей с яркостью выше порога, а фон - множество остальных пикселей с яркостью ниже порога.

8. Как метод Оцу помогает вычислить оптимальное пороговое значение?

Метод Оцу использует гистограмму изображения, которая показывает, сколько пикселей имеют определенную яркость. Если гистограмма имеет два пика, то это означает наличие двух различных классов пикселей: полезные и фоновые. Метод Оцу вычисляет оптимальное пороговое значение, которое разделяет эти два класса.

9. Что означает флаг cv.THRESH_OTSU в функции cv2.threshold и зачем он нужен?

Флаг cv.THRESH_OTSU в функции cv2.threshold указывает на использование метода Оцу для вычисления оптимального порогового значения. Этот флаг позволяет автоматически выбрать наилучшее пороговое значение для разделения объекта и фона.

10. Какими способами можно использовать порог для обработки изображений?

Порог можно использовать для бинаризации изображения, выделения объектов, удаления шума и т.д., например, функция cv2.threshold может использоваться вместе с пороговыми флагами для различных методов бинаризации. Также пороговые методы могут быть использованы вместе с другими методами обработки изображений, такими как морфологические операции.