

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Нахождение и обработка контуров»**

**Отчет по лабораторной работе № 13
по дисциплине «Технологии распознавания образов»**

Выполнил студент группы ПИЖ-б-о-21-1

Халимендик Я. Д. « » 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: обнаружение и выделение контуров на изображении, анализ контуров. Изучение функций `cv2.findContours()`, `cv2.drawContours()`

Ход работы:

Задание 6.1.

С помощью функции `cv2.findContours` найти все контуры изображения.

Лабораторная работа №13

"Пространственные методы обработки изображений"

Задание 7.1.

С помощью функции `cv2.findContours` найти все контуры изображения.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
img = cv2.imread('img.jpg', 0)
img = cv2.medianBlur(img, 5)
```

```
thresh = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)
plt.figure(figsize=(10,10))
plt.imshow(thresh,cmap = 'gray'),plt.title("Thresh")
plt.axis('off');
```

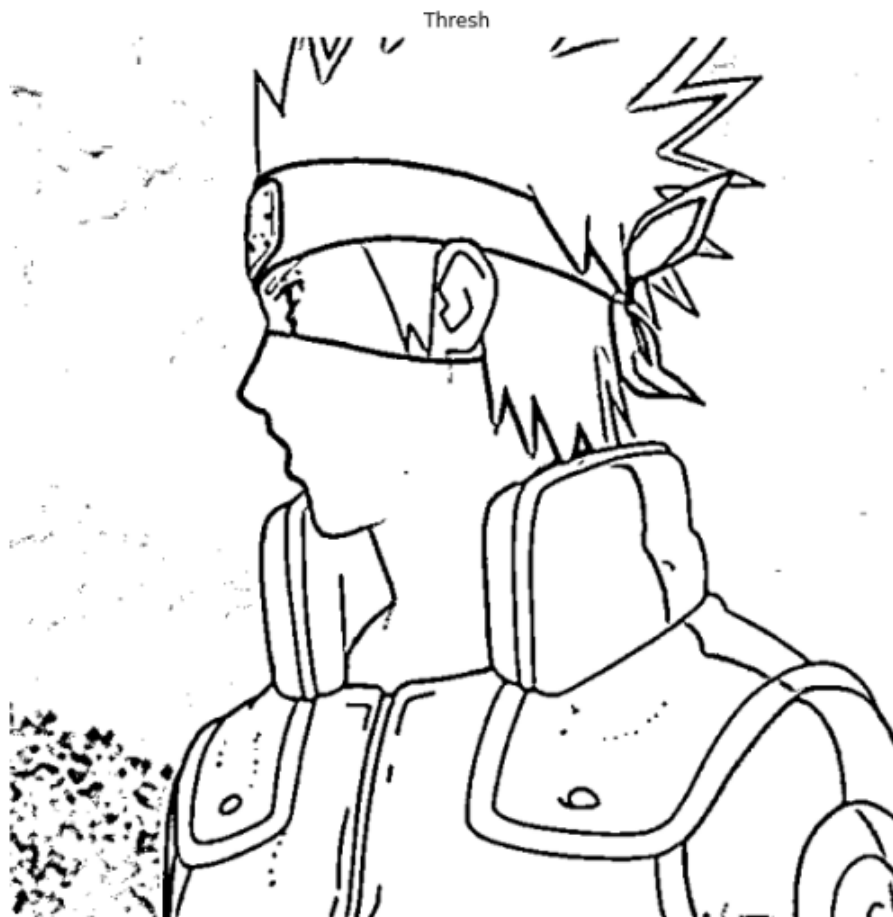


Рисунок 1 – Результат работы

```
: contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cnt = contours[4]
img = cv2.drawContours(img, [cnt], 0, (0,255,0), 3)

: plt.figure(figsize=(10,10))
plt.imshow(img,cmap = 'gray')
plt.title("Contours")
plt.axis('off');
```



Рисунок 2 – Результат работы

Задание 6.2.

Протестировать функцию поиска контура `cv2.findContours` с аргументом `cv2.CHAIN_APPROX_SIMPLE`, который экономит память.

Задание 7.2.

Протестировать функцию поиска контура cv2.findContours с аргументом cv2.CHAIN_APPROX_SIMPLE, который экономит память.

```
img = cv2.imread('img.jpg', 0)
img = cv2.resize(img, (600, 600))
image = cv2.medianBlur(img, 5)
```

```
thresh = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 3,10)
```

```
contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(image, contours, -1, (255,255,255),3);
```

```
plt.figure(figsize=(20,20))
plt.subplot(121),plt.imshow(thresh,cmap = 'gray'),plt.title('Thresh')
plt.axis('off')
plt.subplot(122),plt.imshow(image,cmap = 'gray'),plt.title('Contours')
plt.axis('off')
plt.show();
```



Рисунок 3 – Результат работы

Задание 6.3. Выделить границу методом Канни.

Задание 7.3

Выделить границу методом Канни.

```
img = cv2.imread('img.jpg', 0)
img = cv2.resize(img, (900, 600))
edges = cv2.Canny(img,700,100,apertureSize = 3)
```

```
plt.figure(figsize=(20,20))
plt.subplot(121),plt.imshow(img,cmap = 'gray'),plt.title('Original')
plt.axis('off')
plt.subplot(122),plt.imshow(edges,cmap = 'gray'),plt.title('Cannys edges')
plt.axis('off')
plt.show();
```



Рисунок 3 – Результат работы

Индивидуальное задание.

Обработать изображение, найти и подсчитать количество контуров на изображении, а также определить наибольший контур и найти контур методом Кэнни, подставляя различные пороговые значения.

Задача:

Обработать изображение, найти и подсчитать количество контуров на изображении, а также определить наибольший контур и найти контур методом Кэнни, подставляя различные пороговые значения

```
: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
: img = cv2.imread('img.jpg', 0)
ret,thresh = cv2.threshold(img,128,255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

```
: len_cont = len(contours)
print("Количество контуров: ", len_cont)
```

Количество контуров: 60

```
: plt.figure(figsize=(10,10))
plt.axis('off')
plt.title("Оригинал")
plt.imshow(img, cmap='gray');
```



Рисунок 4 – Результат работы

```

: i = 0
  cnt = []

  four_count = cv2.drawContours(mask,cnt,-1,255,2)
  mask = np.zeros(img.shape,np.uint8)

  #all_cont = cv2.drawContours(mask,contours,-1,255,2)

  max=0
  sel_countour=None
  for countour in contours:
      if countour.shape[0]>max:
          sel_countour=countour
          max=countour.shape[0]

  max_kont = cv2.drawContours(mask, [sel_countour], -1, (255,255,255), 2)

: plt.figure(figsize=(10,10))
  plt.axis('off')
  plt.title("Самый длинный контур")
  plt.imshow(max_kont, cmap='gray');

```



Рисунок 5 – Результат работы

Метод Кэнни

```
: plt.figure(figsize=(20,20))
k = 1
pose = 221
while k <=10:
    canny_edges = cv2.Canny(img,k*100,k,apertureSize = 3)
    plt.subplot(pose)
    plt.axis('off')
    plt.title(f"Значения {k*100, k}")
    plt.imshow(canny_edges, cmap='gray')
    pose +=1
    k *= 2
```



Рисунок 6 – Результат работы