

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

Институт цифрового развития
Кафедра инфокоммуникаций

«Проект»

ОТЧЕТ
по лабораторной работе №15
дисциплины
«Технологии распознавания образов»

Выполнила:
Кувшин Ирина Анатольевна
2 курс, группа ПИЖ-б-о-21-1,
010.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

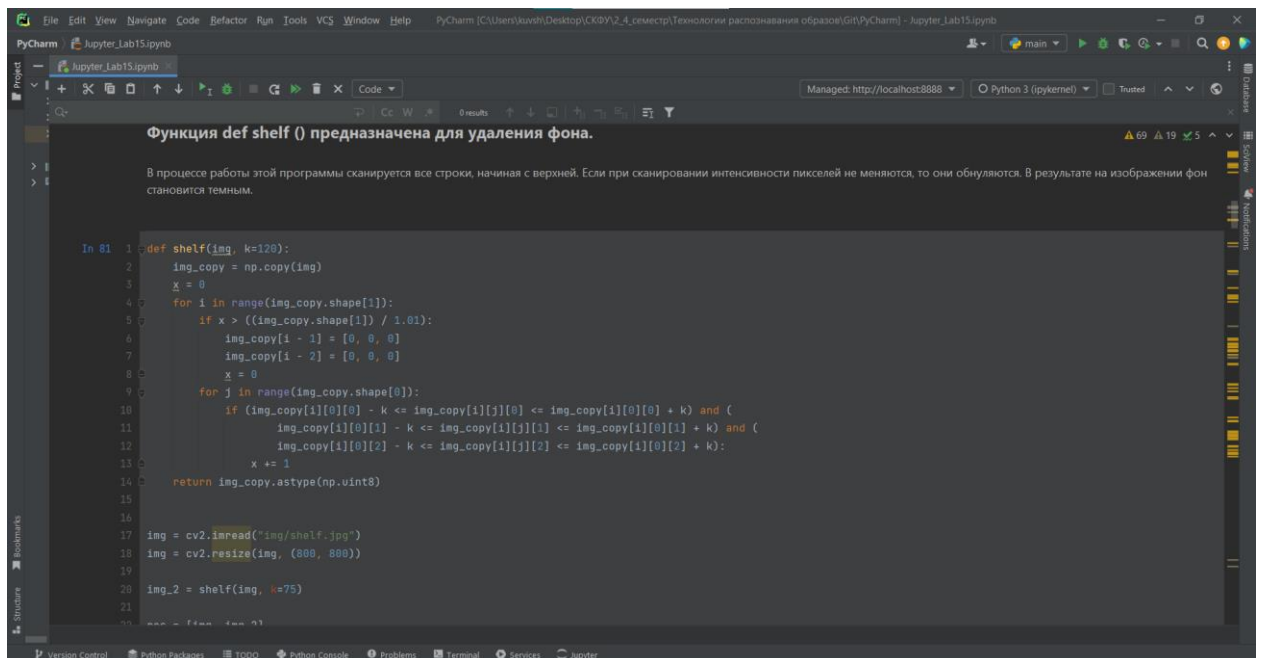
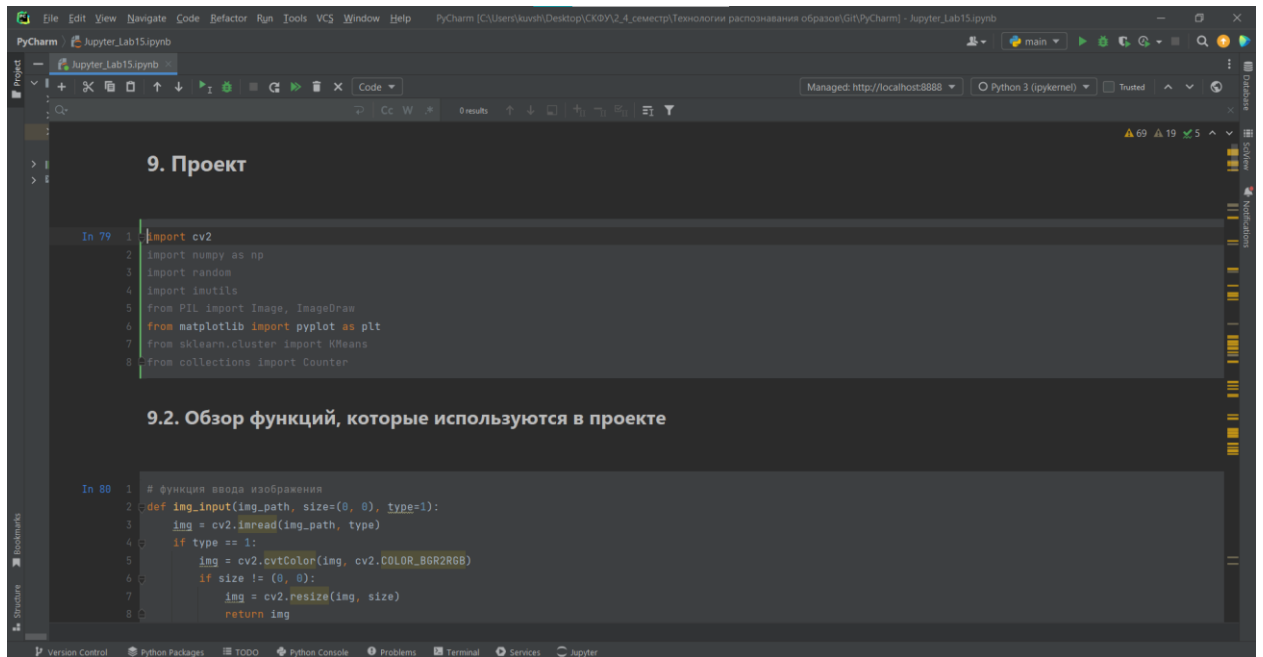
Цель работы:

Выделить на фоне всего изображения интересующие нас объекты. Создать таблицу признаков для сформированного набора объектов. Провести распознавание этих объектов с помощью нейронной сети

В этой таблице объектов используются следующие обозначения: s – площадь, p – периметр, w – ширина, h – высота, w/h – отношение ширины к высоте, $s/(wh)$, отношение, d – эквивалент диаметр, $m00$, $m01$, $m10$, $m11$ – моменты, определяющие площадь, центр масс объекта, и другие моменты более высокого порядка.

	Объекты					
Признаки	Объект 0	Объект 1	Объект 2	Объект 3	Объект 4	Объект 5
s	131.5	140.5	281.5	4315.0	171.0	727.5
p	49.6	70.43	102.38	287.65	102.48	138.07
w	14	10	9	47	7	16
h	16	27	46	99	48	57
w/h	0.875	0.37	0.19	0.47	0.14	0.28
s/wh	0.587	0.52	0.67	0.92	0.50	0.79
d	12.94	13.37	18.9	74.12	14.75	30.43
$m00$	131.5	140.5	281.5	4315.0	171.0	727.5
$m10$	699.16	462.33	962.16	99289.83	406.5	5253.0
$m01$	740.17	1500.16	5609.16	211426.66	2762.0	18065.5
$m11$	3389.21	4822.79	18422.04	4863732.41	5351.25	128150.95
Результаты						

Ход выполнения:



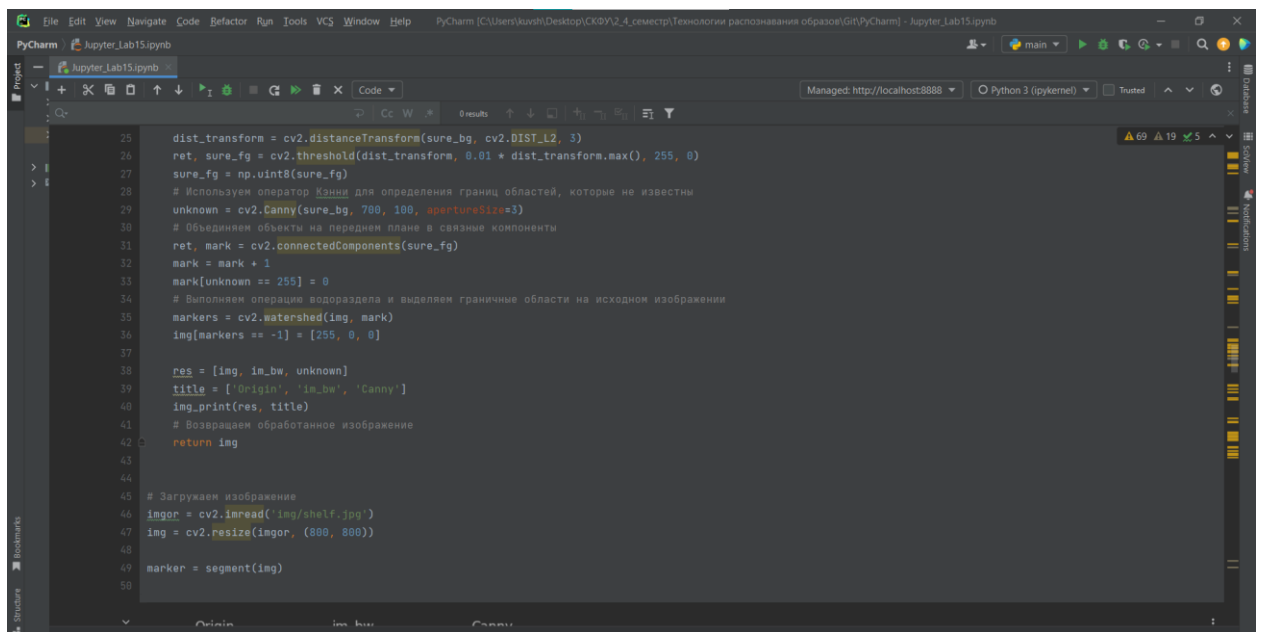
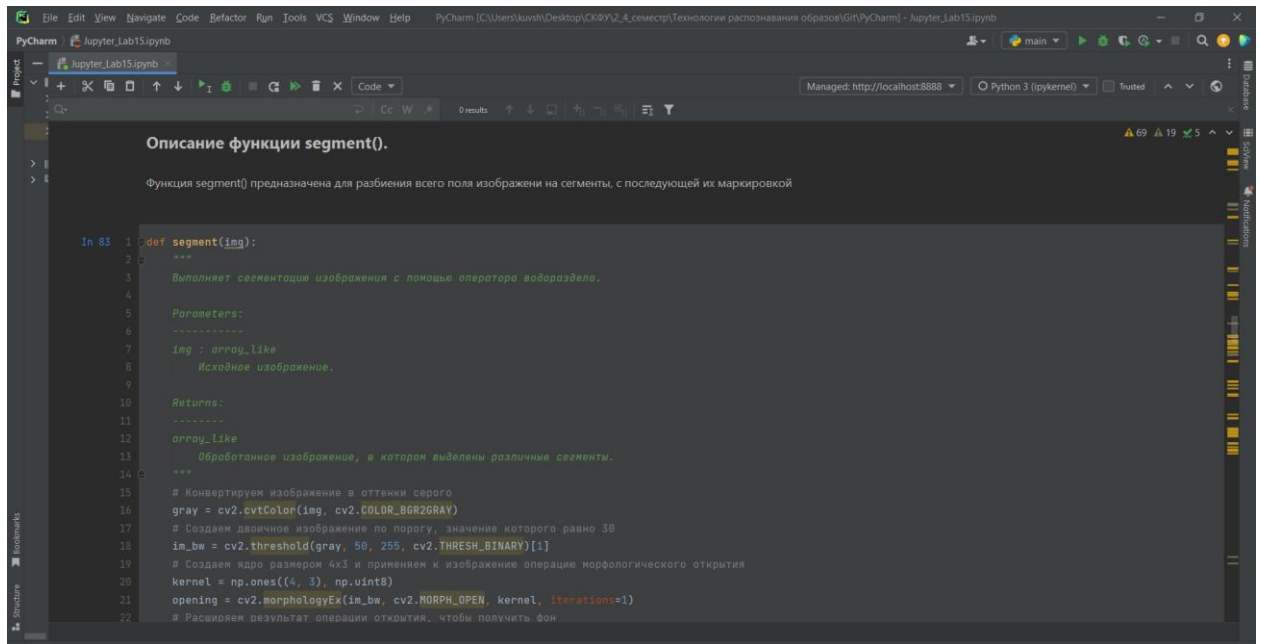
```
PyCharm [C:\Users\kuvsh\Desktop\СКОФ\2_4_семестр\Технологии распознавания образов\Git\PyCharm] - Jupyter_Lab15.ipynb
Jupyter_Lab15.ipynb
0 results
Managed: http://localhost:8888 Python 3 (pykernel) Trusted

Функция удаления фона 2.

In 82: 1 import numpy as np
2 from sklearn.cluster import KMeans
3
4
5 def get_color(image):
6     """
7     Получает доминирующий цвет изображения.
8
9     Parameters:
10    -----
11    image: array
12        RGB-изображение в виде numpy массива
13
14    Returns:
15    -----
16    dominant_color : list
17        RGB-значения доминирующего цвета
18    """
19    # Преобразуем изображение в одномерный массив и приводим тип данных к float32
20    pixels = np.float32(image.reshape(-1, 3))
21    # Задает количество цветов для извлечения
22    n_colors = 1
23    # Возвращает KMeans на основании pixels array
```

```
PyCharm [C:\Users\kuvsh\Desktop\СКОФ\2_4_семестр\Технологии распознавания образов\Git\PyCharm] - Jupyter_Lab15.ipynb
Jupyter_Lab15.ipynb
0 results
Managed: http://localhost:8888 Python 3 (pykernel) Trusted

41
42
43 Parameters:
44 -----
45 imr : array_like
46     Исходное изображение.
47 p : int, optional
48     Погрешность в определении доминирующего цвета. Значение по умолчанию - 48.
49
50 Returns:
51 -----
52 array_like
53     Изображение с удаленным фоновым цветом.
54 """
55 # Преобразуем изображение в формат HSV для определения доминирующего цвета
56 hsv_image = cv2.cvtColor(imr, cv2.COLOR_BGR2HSV)
57 # Получаем доминирующий цвет в формате HSV
58 dom_color = get_color(hsv_image)
59 # Создаем массив с заданными размерами и заполненный белым цветом
60 dom_color_hsv = np.full(imr.shape, [255, 255, 255], dtype='uint8')
61 # Определяем нижнюю и верхнюю границы доминирующего цвета с учетом погрешности
62 lower = np.array([dom_color[0] - p, dom_color[1] - p, dom_color[2] - p])
63 upper = np.array([dom_color[0] + p, dom_color[1] + p, dom_color[2] + p])
64 # Создаем маску для удаления фона
65 mask = cv2.inRange(hsv_image, lower, upper)
66 # Обновляем значения на противоположные для выделения фона
67 res = cv2.bitwise_not(dom_color_hsv, hsv_image, mask=mask)
68 # Конвертируем обратно в формат BGR
69 return cv2.cvtColor(res, cv2.COLOR_HSV2BGR)
```



```
PyCharm | Jupyter_Lab15.ipynb
Jupyter_Lab15.ipynb
0 results
Managed: http://localhost:8888
Python 3 (pykernel)
Trusted

Метод заключается в поиске объекта на большом изображении, который соответствует выбранному шаблону. Поиск проводится с помощью функции cv2.matchTemplate(). Шаблон перемещается по исходному изображению, его пиксели сравниваются с пикселями этого изображения. На выходе получаем изображение в градациях серого.

In 84: 1 # Загружаем изображение и шаблон
2 img_rgb = cv2.imread('img/shelf.jpg')
3 template = cv2.imread('img/1.jpg', 0)
4 # Копируем исходное изображение для вывода результата с выделенными объектами
5 res_img = img_rgb.copy()
6 # Преобразуем изображение в оттенки серого, чтобы применить поиск шаблона
7 img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
8 # Получаем размеры шаблона
9 w, h = template.shape[::-1]
10 # Выполняем соответствие шаблона на исходном изображении
11 res = cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED)
12 # Задаем порог срабатывания
13 threshold = 0.8
14 # Получаем координаты срабатываний, где значение больше порога
15 loc = np.where(res >= threshold)
16 # Выделяем области срабатываний на изображении
17 for pt in zip(*loc[::-1]):
18     cv2.rectangle(res_img, pt, (pt[0] + w, pt[1] + h), (255, 0, 0), 1)
19 # Отображаем и сохраняем результаты
20 cv2.imwrite('img/res.png', res_img)
21 res = [img_rgb, res_img]
22 title = ['Оригинал', 'Результат']
23 img_print(res, title)
```

```
PyCharm | Jupyter_Lab15.ipynb
Jupyter_Lab15.ipynb
0 results
Managed: http://localhost:8888
Python 3 (pykernel)
Trusted

In 86: 1 img = cv2.imread('img/5.jpg', 0)
2 th = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, 2)
3 contours, hierarchy = cv2.findContours(th, 5, 5)
4 cnt = contours[0] # создание контура
5 area = cv2.contourArea(cnt)
6 prm = cv2.arcLength(cnt, True)
7 hull = cv2.convexHull(cnt) # выпуклая оболочка
8 #print(hull)
9 x, y, w, h = cv2.boundingRect(cnt)
10 print(x, y, w, h)
11 image = cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
12
13 res = [th, image]
14 title = ['Binary', 'Rectangle']
15 img_print(res, title)
16
17 # параметры бинарного изображения
18 x, y, w, h = cv2.boundingRect(cnt)
19 epsilon = 0.1 * cv2.arcLength(cnt, True)
20 aspect_ratio = float(w) / h # соотношение сторон
21 rect_area = w * h
22 extent = float(area) / rect_area
23 hull_area = cv2.contourArea(hull) # площадь выпуклой оболочки
24 equi_diameter = np.sqrt(4 * area / np.pi)
25 mask = np.zeros(img.shape, np.uint8)
```

Рисунок 15.1– Код программы

Контрольные вопросы:

1. Что имеет большое значение при изучении любой дисциплины?

При изучении любой дисциплины большое значение имеет практическое применение накопленных знаний, формирование навыков их использования.

2. Что происходит на первом этапе алгоритма создания таблицы признаков для множества объектов согласно теории из учебника?

На первом этапе проводится предварительная обработка изображений, которая включает в себя удаление шума, повышение резкости изображений. С помощью этой обработки выделяются характерные детали, подавляется шум, повышается быстродействие, уменьшается объем информации.

3. Какая операция используется для выделения объектов?

Для выделения объектов используется операция сегментации изображения методом водораздела, с последующей маркировкой результата сегментации.

4. Что необходимо сделать с объектами для их более точного выделения?

Для более точного выделения объектов их нужно разнести, увеличив расстояние между ними.

5. Каким образом работает функция `def shelf()`, приведенная в теории?

Функция `def shelf ()` предназначена для удаления фона. В процессе работы этой программы сканируется все строки, начиная с верхней. Если при сканировании интенсивности пикселей не меняются, то они обнуляются. В результате на изображении фон становится темным.

6. Что такое сегментация изображения?

Сегментация изображения – это разбиение изображения на множество покрывающих его областей, называемых сегментами.

7. Что делают с помощью сегментации?

С ее помощью можно выделить объекты, создавая границы между ними.

8. В чем заключен метод распознавания метода по шаблону?

Метод заключается в поиске объекта на большом изображении, который соответствует выбранному шаблону.

9. С помощью какой функции проводится распознавание объекта по шаблону?

Поиск проводится с помощью функции `cv2.matchTemplate()`. Шаблон перемещается по исходному изображению, его пиксели сравниваются с

пикселями этого изображения. На выходе получаем изображение в градациях серого.

10. Что такое признак?

Признак – это количественная характеристика того или иного свойства исследуемого объекта.

11. Что такое образ?

Образ – это совокупность признаков, с помощью которых объект идентифицируется с большой вероятностью.

12. Что такое класс образов и что он характеризует?

Класс образов – это совокупность образов с общими свойствами.

Класс образов – характеризует наличие «схожести» объектов.