

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

Институт цифрового развития
Кафедра инфокоммуникаций

«Процессы дискретизации и квантования изображения»

ОТЧЕТ
по лабораторной работе №8
дисциплины
«Основы распознавания образов»

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Процессы дискретизации и квантования изображения

Цель работы: изучение функций, используемых для моделирования процессов квантования и дискретизации изображения на языке Python.

Ход работы:

Задание 2.1

```
: image = cv2.imread('mustang.jpg')
img = image.copy()
K = 10
s = img.shape
h1, w1 = s[0], s[1]
h = (s[0] - s[0] % K)
w = (s[1] - s[1] % K)
img = cv2.resize(img, (w,h))

for y in range(0, h-1, K):
    for x in range(0, w-1, K):
        if len(s) > 2:
            s = np.average(img[y:(y+K), x:(x+K)], axis=0)
            img[y:(y+K), x:(x+K)] = np.average(s, axis=0)
        else:
            s = img[y:(y+K), x:(x+K)]
            img[y:(y+K), x:(x+K)] = np.average(s)
img = cv2.resize(img, (w1, h1))
res = np.hstack((image, img))
cv2.imshow('Img', res)
cv2.waitKey(0)

: -1
```

Задание 2.2

```
: plt.subplot(121)
img = cv2.imread('jaguar.jpg')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.subplot(122)
plt.axis("off")
Z = img.reshape((-1,3))
Z = np.float32(Z)
crt=(cv2.TERM_CRITERIA_EPS+cv2.TERM_CRITERIA_MAX_ITER,
      10, 1.0)
K = 4
ret, label, center = cv2.kmeans(Z, K, None, crt, 10, cv2.KMEANS_RANDOM_CENTERS)
center = np.uint8(center)
res = center[label.flatten()]
res2 = res.reshape((img.shape))
plt.imshow(res2)
cv2.waitKey(0)
```

Рисунок 1 – Задания

Задание 2.2

```
In [ ]: plt.subplot(121)
img = cv2.imread('jaguar.jpg')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.subplot(122)
plt.axis("off")
Z = img.reshape((-1,3))
Z = np.float32(Z)
crt=(cv2.TERM_CRITERIA_EPS+cv2.TERM_CRITERIA_MAX_ITER,
      10, 1.0)
K = 4
ret, label, center = cv2.kmeans(Z, K, None, crt, 10, cv2.KMEANS_RANDOM_CENTERS)
center = np.uint8(center)
res = center[label.flatten()]
res2 = res.reshape((img.shape))
plt.imshow(res2)
cv2.waitKey(0)
```

In []: -1



Рисунок 2 – Задания

Выделение объекта синего цвета

```
img = cv2.imread ( "bmw.jpg" );
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
lower_blue = np.array([90, 50, 70])
upper_blue = np.array([130, 255, 255])
mask = cv2.inRange(hsv, lower_blue, upper_blue)
res = cv2.bitwise_and(img,img, mask= mask)

plt.subplot(221)
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
plt.axis("off")

plt.subplot(222)
plt.imshow(cv2.cvtColor(mask,cv2.COLOR_BGR2RGB))
plt.axis("off")

plt.subplot(223)
plt.imshow(cv2.cvtColor(res,cv2.COLOR_BGR2RGB))
plt.axis("off")

(-0.5, 999.5, 1499.5, -0.5)
```

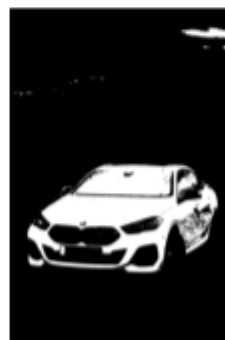


Рисунок 3 – Индивидуальное задание

Вывод: изучены функции, использующиеся для моделирования процессов квантования и дискретизации изображения на языке Python.

Ответы на контрольные вопросы:

1. Что такое интенсивность изображения? Интенсивность изображения $f(x, y)$ является функцией двух пространственных переменных x и y на ограниченной прямоугольной области.

2. Что такое квантование изображения? Процесс разбиения непрерывного динамического диапазона значений яркости на ряд дискретных уровней называется квантованием.

3. Чему равно число уровней квантования?

$$K = \lceil A/\Delta A \rceil,$$

4. Как можно изменить размер массива изображения? Функция `reshape()`.

5. Как можно создать копию изображения? Функция `copy()`.

6. Что такое дискретизация изображений? Дискретизация – это преобразование непрерывного сигнала в последовательность чисел (отсчетов), то есть представление этого сигнала по какому-либо конечномерному базису. Это представление состоит в проектировании сигнала на данный базис.

7. Приведите алгоритм дискретизации. Алгоритм дискретизации: разбиваем три матрицы цветного изображения на отдельные блоки с шагом дискретизации K . В каждом блоке вычисляем среднее значение по каждому цвету в отдельности и полагаем, что внутри блока интенсивность равна вычисленному среднему значению.

8. Как изменить цветовую модель изображения с BGR на RGB?
`Cv2.COLOR_BGR2RGB`

9. Как проверить: изображение цветное или полутоновое? Воспользуемся функцией `s = img.shape` (получим размер изображения и его тональность) Далее проверим `len(s)` В цветном строка вернёт число 3 (так как у цветного три параметра: длина, ширина и число каналов) А у полутонового вернёт только два: длина и высота

10. Функция квантования. `Cv2.kmeans()`