

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Информатика»
Тема: Введение в архитектуру компьютера

Студент гр. 3343

Старков С.А

Преподаватель

Иванов Д. В.

Санкт-Петербург

2023

Цель работы

Научится работать с библиотекой Pillow, выполнять различные графические преобразования над изображениями.

Задание

Предстоит решить 3 подзадачи, используя библиотеку Pillow (PIL). Для реализации требуемых функций студент должен использовать numpy и PIL. Аргумент `image` в функциях подразумевает объект типа `<class 'PIL.Image.Image'>`

1) Рисование пентаграммы в круге

Необходимо написать функцию `pentagram()`, которая рисует на изображении пентаграмму в круге.

Функция `pentagram()` принимает на вход:

Изображение (`img`)

координаты левого верхнего и правого нижнего угла квадрата, в который вписана окружность (`x0,y0,x1,y1`)

Толщину линий и окружности (`thickness`)

Цвет линий и окружности (`color`) - представляет собой список (`list`) из 3-х целых чисел

Функция должна вернуть обработанное изображение.

Примечание:

Вершины пентаграммы вычислять по формуле:

$$\phi_i = (\pi/5) * (2*i + 3/2)$$

$$\text{node_}i = (\text{int}(x_0 + r * \cos(\phi_i)), \text{int}(y_0 + r * \sin(\phi_i)))$$

`x0,y0` - координаты центра окружности, в который вписана пентаграмма

`r` - радиус окружности

`i` - номер вершины от 0 до 4

Подсказка: Округляйте все вычисляемые вами значения (кроме значений углов) до целых чисел.

2) Инвертирование полос

Необходимо реализовать функцию `invert`, которая делит изображение на "полосы" и инвертирует цвет нечетных полос (счёт с нуля).

Функция `invert()` принимает на вход:

Изображение (`img`)

Ширину полос в пикселах (`N`)

Признак того, вертикальные или горизонтальные полосы (`vertical` - если `True`, то вертикальные)

Функция должна разделить изображение на вертикальные или горизонтальные полосы шириной N пикселей. И инвертировать цвет в нечетных полосах (счет с нуля). Последняя полоса может быть меньшей ширины, чем N.

3) Поменять местами 9 частей изображения

Необходимо реализовать функцию `mix`, которая делит квадратное изображение на 9 равных частей (сторона изображения делится на 3), и по правилам, записанным в словаре, меняет их местами.

Функция `mix()` принимает на вход:

Изображение (`img`)

Словарь с описанием того, какие части на какие менять (`rules`)

Пример словаря `rules`:

`{0:1,1:2,2:4,3:4,4:5,5:3,6:8,7:8,8:8}`

Элементы нумеруются слева-направо, сверху-вниз.

В данном случае нулевой элемент заменяется на первый, первый на второй, второй на четвертый, третий на четвертый и так далее.

Функция должна вернуть обработанное изображение.

Выполнение работы

Мой программный код написан на языке Python и использует библиотеку Pillow для обработки изображений. Программа включает три функции, каждая из которых выполняет свою задачу.

Функция **pentagram**:

- Функция рисует пентаграмму на изображении с использованием библиотеки PIL.
- Для рисования пентаграммы используется математическая формула для вычисления координат точек.
- Линии и окружность рисуются с помощью методов изображения ImageDraw.
- Результат возвращается в виде обработанного изображения. Функция

Функция **invert**:

Функция инвертирует каждую N-ую линию изображения в зависимости от значения параметра `vertical`.

- Используется метод `crop` для выделения области изображения, которая затем инвертируется с помощью метода `ImageOps.invert`.
- Результат возвращается в виде обработанного изображения.

Функция **mix(img, rules)**:

- Функция меняет местами 9 частей изображения в соответствии с заданными правилами.
- Используется библиотека `numpy` для работы с массивами изображения.
- Изображение разбивается на 9 частей, которые затем переставляются в соответствии с правилами.
- Результат возвращается в виде обработанного изображения.

Данный код демонстрирует использование библиотеки Pillow для работы с изображениями и реализации различных преобразований. Эти функции могут быть полезны при обработке изображений с целью создания графических эффектов или изменения их структуры в соответствии с заданными правилами.

Разработанный программный код см. в приложении А.

Тестирование см. в приложении Б.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from PIL import Image, ImageDraw, ImageOps
import math
import numpy as np

def pentagram(img, x0, y0, x1, y1, thickness, color):
    r = (x1 - x0) // 2
    center_x = x0 + (x1 - x0) // 2
    center_y = y0 + (y1 - y0) // 2
    coordinates = []
    color = tuple(color)
    for i in range(1, 6):
        coordinate_x = int(center_x + r * math.cos((math.pi / 5) * (2 * i
+ 3 / 2)))
        coordinate_y = int(center_y + r * math.sin((math.pi / 5) * (2 * i
+ 3 / 2)))
        coord = (coordinate_x, coordinate_y)
        coordinates.append(coord)
    drawing = ImageDraw.Draw(img)
    drawing.line(((coordinates * 2)[::2] + [coordinates[0]]), color,
thickness)
    drawing.ellipse((x0, y0, x1, y1), None, color, thickness)
    return img

def invert(img, N, vertical):
    columns = img.size[0]
    strs = img.size[1]
    strips = math.ceil((columns if vertical else strs) / N)
    for i in range(1, strips, 2):
        borders = (N * i, 0, min(N * (i + 1), columns), strs) if vertical
else (0, N * i, columns, min(N * (i + 1), strs))
        invert_strips = ImageOps.invert(img.crop(borders))
        img.paste(invert_strips, (N * i, 0) if vertical else (0, N * i))
    return img

def mix(img, rules):
    pix = np.array(img)
    w, h = img.size
    x, y = w // 3, h // 3
    box_l = [[] for i in range(9)]
    box = []
    for i in range(3):
        for j in range(3):
            box.append(pix[x * i:x * (i + 1), y * j:y * (j + 1)])
    for i in rules:
        box_l[i] = box[rules[i]]
    image = np.vstack([np.hstack((box_l[i], box_l[i + 1], box_l[i + 2]))
for i in range(0, 7, 3)])
    img = Image.fromarray(image)
    return img
```

ПРИЛОЖЕНИЕ Б

ТЕСТИРОВАНИЕ

Исходные изображения:



Рисунок 1 – Изображение для функции invert (krab1.jpeg)



Рисунок 2 – Изображение для функции mix (krab1.jpeg)

Результат:

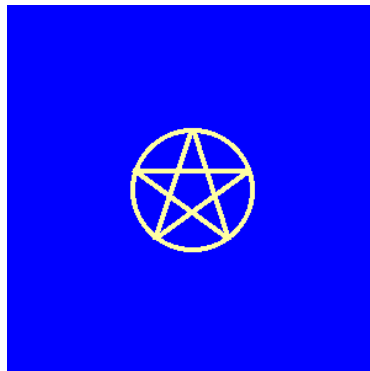


Рисунок 1 – Результат функции pentagram



Рисунок 2 – Результат функции invert



Рисунок 3 – Результат функции mix

Выводы

Были изучены различные способы преобразования изображения, написаны функции с использованием библиотеки Pillow.