

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

Наумкин А.Д.

Преподаватель

Иванов Д.В

Санкт-Петербург

2023

Цель работы

Целью данной работы было создание трех функций, использующих библиотеки *Pillow (PIL)* и *numpy* для обработки изображений. Первая функция должна была рисовать треугольник на изображении, принимая координаты вершин, толщину и цвет линий, а также цвет, которым треугольник должен быть залит. Вторая функция должна была заменять наиболее часто встречаемый цвет на переданный цвет в изображении. Третья функция должна была создать коллаж изображений, повторяя переданное изображение заданное количество раз по вертикали и горизонтали.

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'>`

Задача 1. Рисование треугольника

Необходимо написать функцию `triangle()`, которая рисует на изображении треугольник

Функция `triangle()` принимает на вход:

- Изображение (`img`)
- Координаты вершин (`x0,y0,x1,y1,x2,y2`)
- Толщину линий (`thickness`)
- Цвет линий (`color`) - представляет собой список (`list`) из 3-х целых чисел
- Цвет, которым залит (`fill_color` - если значение `None`, значит треугольник не залит) - представляет собой список (`list`) из 3-х целых чисел

Функция должна вернуть исходное обработанное изображение.

Задача 2. Замена наиболее часто встречаемого цвета

Необходимо написать функцию `change_color()`, которая заменяет наиболее часто встречаемый цвет на переданный.

Функция `change_color()` принимает на вход:

- Изображение (`img`)
- Цвет (`color` - представляет собой **список** из трех целых чисел)

Функция должна найти в изображении самый частый цвет и заменить его на переданный, затем вернуть новое изображение (исходное изображение не должно меняться).

Задача 3. Коллаж

Необходимо написать функцию *collage()*.

Функция *collage()* принимает на вход:

- *Изображение (img)*
- *Количество изображений по "оси" Y (N - натуральное)*
- *Количество изображений по "оси" X (M - натуральное)*

Функция должна создать коллаж изображений (это же изображение, повторяющееся $N \times M$ раз. (N раз по высоте, M раз по ширине) и вернуть его (новое изображение).

При необходимости можно писать дополнительные функции.

Выполнение работы

1. Функция *triangle(img, x0, y0, x1, y1, x2, y2, thickness, color, fill_color=None)*.

Принимает изображение *img*, координаты вершин треугольника $(x0, y0)$, $(x1, y1)$, $(x2, y2)$, толщину линий *thickness*, цвет линий *color* и опциональный цвет заливки *fill_color*. Создает объект *ImageDraw.Draw*, рисует треугольник на изображении с использованием переданных параметров и возвращает измененное изображение.

2. Функция *change_color(img, color)*. Принимает изображение *img* и цвет *color*. Преобразует изображение в массив *numpy* и находит наиболее часто встречаемый цвет. Заменяет все пиксели этого цвета на переданный цвет и возвращает новое изображение.

3. Функция *collage(img, N, M)*. Принимает изображение *img*, количество изображений по вертикали *N* и количество изображений по горизонтали *M*.

Создает новый объект изображения, заполняет его повторяющимися изображениями по указанным координатам и возвращает новое изображение.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Изображение (шириной 100, высотой 100), (10, 10, 50, 90, 90, 10), 2, [255, 0, 0], [0, 255, 0]	Изображение с нарисованным красным треугольником, заливка зеленого цвета	Функция успешно рисует треугольник с заданными параметрами и правильно заполняет его указанным цветом.
2.	Изображение (шириной 200, высотой 200)	Изображение с наиболее часто встречающимся цветом, замененным на [0, 0, 255]	Функция корректно определяет наиболее частый цвет в изображении и заменяет его на заданный синий цвет.
3.	Изображение (шириной 50, высотой 50), 3, 2	Коллаж изображения, повторенного 6 раз (3x2)	Функция успешно создает коллаж изображений, повторяя заданное изображение 3 раза по вертикали и 2 раза по горизонтали.

Выводы

В результате работы были созданы три функции для обработки изображений с использованием библиотек *Pillow* (PIL) и *numpy*. Эти функции могут быть полезны в различных задачах, связанных с обработкой и манипуляциями изображений, таких как создание графики, замена цветов и создание коллажей. Они позволяют легко и эффективно выполнять указанные задачи, обрабатывая изображения с минимальным использованием ресурсов.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from PIL import Image
from collections import Counter
import numpy as np

def triangle(img, x0, y0, x1, y1, x2, y2, thickness, color,
fill_color=None):
    draw = ImageDraw.Draw(img)
    draw.polygon(((x0, y0), (x1, y1), (x2, y2)), width=thickness,
outline=tuple(color), fill=tuple(fill_color) if fill_color else
None)

    return img

def change_color(img, color):
    img_array = np.array(img)
    colors_count = Counter(tuple(pixel) for i in img_array for
pixel in i)
    most_common_color = colors_count.most_common(1)[0][0]
    img_array[(img_array == most_common_color).all(axis=2)] =
color
    new_img = Image.fromarray(img_array)

    return new_img

def collage(img, N, M):
    width, height = img.size
    collage_img = Image.new("RGB", (M * width, N * height))

    for i in range(N):
        for j in range(M):
            collage_img.paste(img, (j * width, i * height))

    return collage_img
```