

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. 3344

Анахин Е.Д.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Написание функций для работы с изображениями и использование библиотеки PIL на языке Python.

Задание

Вариант 1. 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'>

Функция 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.

Функция 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).

Функция 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 NxM раз. (N раз по высоте, M раз по ширине) и вернуть его (новое изображение).

Выполнение работы

Для выполнения первой функции `triangle` в ней создается объект `ImageDraw` с названием `draw` для рисования внутри изображения `image`. Далее создается список из кортежей координат. Каждая координата является вершиной треугольника, который нужно нарисовать в `img`. Далее, если переданный аргумент `fill_color` не равен `None`, то рисуется треугольник с вершинами в координатах `coords`, с толщиной линии `thickness` и цветом линии `color` и цветом фона `fill_color`. Если `fill_color` равен `None`, то рисуется треугольник без фона. `fill_color` и `color` преобразуются к кортежу, так как эти аргументы могут быть только типа `tuple` или `int`.

Для выполнения второй функции `change_color` делается копия изображения, переданного в аргументах функции. `X_size` и `y_size` хранят в себе размеры изображения. Далее создается словарь для подсчета количества каждого цвета. Далее, с помощью цикла `for` и еще одного, вложенного в него подсчитывается количество каждого цвета. Переменная `max_color_count` хранит в себе это значение. Далее цикл проходится по словарю `colors` и, если количество цветов равно максимальному, то сохраняется название цвета. Далее циклами `for` проходится по всем цветам и, если цвет равен самому часто, то он меняется на цвет `color`.

Для выполнения третьей функции `collage` находятся ширина и высота картинки, которую нужно будет добавить на новое изображение, создается новое пустое изображение размерами `ширина * количество картинок по x`, `высота * количество картинок по y`. Далее, с помощью циклов `for` на эту картинку добавляются изображения `img`.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	triangle(Image.new('RGB', (500,500),"black"), 250, 400, 100, 100, 400, 100, 7, (0,255,0), (250,0,0))	img	-
2.	change_color(Image.new('RGB', (400,400),"white"), (0,255,0))	img	-
3.	collage(Image.new('RGB', (300,300),"blue"),5,6)	collage	-

Выводы

Было написано 3 функции обработки изображений на языке Python, а также была изучена работа с библиотекой PIL, использующейся для обработки изображений.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Anakhin_Egor_lb2.py

```
from PIL import Image, ImageDraw

def triangle(img, x0, y0, x1, y1, x2, y2, thickness, color, fill_color):

    draw = ImageDraw.Draw(img)
    coords = [(x0, y0), (x1, y1), (x2, y2)]
    if fill_color:
        draw.polygon(coords, width=thickness, outline=tuple(color),
fill=tuple(fill_color))
    else:
        draw.polygon(coords, width=thickness, outline=tuple(color))
    return img

def change_color(img, color):

    img_copy = img.copy()
    x_size, y_size = img.size
    colors = {}

    for x in range(x_size):
        for y in range(y_size):
            pixel_color = img.getpixel((x, y))
            if pixel_color not in colors:
                colors[pixel_color] = 0
            colors[pixel_color] += 1

    max_color_count = max(colors.values())
    color_name = ''

    for colorm in colors:
        if colors[colorm] == max_color_count:
            color_name = colorm

    for x in range(x_size):
        for y in range(y_size):
            if img.getpixel((x, y)) == color_name:
                img_copy.putpixel((x, y), tuple(color))

    return img_copy

def collage(img, N, M):

    width, height = img.size
    new_img = Image.new("RGB", (width*M, height*N))

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

    return new_img
```