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 _____ Щербак М.С.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Освоение обработки изображений 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'>

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.

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).

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).

При необходимости можно писать дополнительные функции.

Выполнение работы

Функция triangle рисует треугольник на изображении img с помощью модуля PIL. Она принимает следующие параметры:

- x0, y0, x1, y1, x2, y2 - координаты вершин треугольника,

- thickness - толщина линии треугольника,

- color - цвет линии треугольника в формате RGB,

- fill_color - цвет заливки треугольника в формате RGB или None, если заливка не требуется.

Функция создает объект draw с помощью метода ImageDraw.Draw, который позволяет рисовать на изображении. Затем она определяет координаты треугольника в виде кортежа cord. Если указан цвет заливки (fill_color не равен None), то функция вызывает метод draw.polygon для рисования треугольника с заданными параметрами (цвет заливки, цвет линии и толщина). Если заливка не требуется, вызывается аналогичный метод без параметра заливки. В конце функция возвращает изображение.

Функция change_color изменяет цвет пикселей изображения img на заданный цвет color. Она создает копию изображения img с помощью метода img.copy(). Затем функция проходит по каждому пикселю изображения с помощью двух вложенных циклов for. Внутри циклов каждая точка проверяется на равенство заданному цвету color_name, который определяется как наиболее часто встречаемый цвет пикселя в исходном изображении img. Если цвет точки равен color_name, то он заменяется на новый цвет color с помощью метода img_copy.putpixel. В конце функция возвращает измененную копию изображения.

Функция collage создает коллаж изображений путем повторного встраивания исходного изображения img в новое изображение. Она принимает два параметра: N - количество строк коллажа и M - количество столбцов

коллажа. Сначала функция получает размеры исходного изображения width и height. Затем создается новое изображение new_img с помощью метода Image.new, указывая размеры width * M и height * N и режим "RGB". Затем исходное изображение img последовательно встраивается в новое изображение new_img с помощью метода new_img.paste, указывая координаты вставки (width * i, height * j) для каждого столбца и строки. В конце функция возвращает новое изображение-коллаж.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	triangle(Image.new("RGB", (300, 300)), 67, 82, 139, 154, 4, [197, 114, 130])	img	-
2.	change_color(Image.open("image.jpg"), [255, 0, 0])	img	-
3.	collage(Image.new("RGB", (600, 400)), 2, 3)	img	-

Выводы

Была освоена обработка изображений на языке Python. Были получены базовые навыки работы с пакетом *Pillow*. Были освоены функции рисования фигур и линий.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.py

```
from PIL import Image, ImageDraw
# Задача 1
def triangle(img, x0, y0, x1, y1, x2, y2, thickness, color, fill_color):
    draw = ImageDraw.Draw(img)
    cord = ((x0, y0), (x1, y1), (x2, y2))
    if fill_color is not None:
         draw.polygon(cord, fill=tuple(fill_color), outline=tuple(color),
width=thickness)
    else:
        draw.polygon(cord, outline=tuple(color), width=thickness)
    return img

# Задача 2
def change_color(img, color):
    img_copy = img.copy()
    x_size, y_size = img.size
    cveta = {}

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

    mas = max(cveta.values())
    color_name = ''

    for i in cveta:
        if cveta[i] == mas:
            color_name = i

    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

# Задача 3
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
```