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 Клюкин А.В.

Преподаватель Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Изучить работу библиотеки Pillow.

Задание

Вариант 4

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 отрезка. Отрезок определяется:

координатами начала

координатами конца

цветом

толщиной.

Необходимо реализовать функцию user_func(), рисующую на картинке отрезок

Функция user_func() принимает на вход:

изображение;

координаты начала (x0, y0);

координаты конца (x1, y1);

цвет;

толщину.

Функция должна вернуть обработанное изображение.

2) Преобразовать в Ч/Б изображение (любым простым способом).

Функционал определяется:

Координатами левого верхнего угла области;

Координатами правого нижнего угла области;

Алгоритмом, если реализовано несколько алгоритмов преобразования изображения (по желанию студента).

Нужно реализовать 2 функции:

check_coords(image, x0, y0, x1, y1) - проверяет координаты области (x0, y0, x1, y1) на корректность (они должны быть неотрицательными, не

превышать размеров изображения, поскольку x0, y0 - координаты левого верхнего угла, x1, y1 - координаты правого нижнего угла, то x1 должен быть больше x0, а y1 должен быть больше y0);

set_black_white(image, x0, y0, x1, y1) - преобразовывает заданную область изображения в черно-белый (используйте для конвертации параметр '1'). В этой функции должна вызываться функция проверки, и, если область некорректна, то должно быть возвращено исходное изображение без изменений. Примечание: поскольку черно-белый формат изображения (greyscale) является самостоятельным форматом, а не вариацией RGB-формата, для его получения необходимо использовать метод Image.convert.

3) Найти самый большой прямоугольник заданного цвета и перекрасить его в другой цвет. Функционал определяется:

Цветом, прямоугольник которого надо найти

Цветом, в который надо его перекрасить.

Написать функцию find_rect_and_recolor(image, old_color, new_color), принимающую на вход изображение и кортежи rgb-компонент старого и нового цветов. Она выполняет задачу и возвращает изображение. При необходимости можно писать дополнительные функции.

Выполнение работы

Было написано 4 функции

user_func, которая в зависимости от параметров строила отрезок с помощью метода line

check_coords, которая проверяла корректность координат и возвращала логическое выражение

set_black_white, которая на основе предыдущей функции либо преобразовывает изображение в ЧБ с помощью .convert, либо возвращает начальное изображение

find_rect_and_recolor, которая получает на вход изображение и два цвета. В ней сначала создаются основные переменные и массив. picture = image.load() - получает массив, содержащий данные о пикселях. Затем записываются размеры полученного изображения. Создаются два кортежа, которые будут содержать координаты крайних точек нужного прямоугольника. Начинается цикл в котором перебираются все пиксели. В случае, если пиксель подходит под заданный изменяемый цвет, то создаются две новые переменные координат и приравниваются их к текущим, чтобы не потерять начало фигуры. Далее циклом идет просчет длины и высоты фигуры. Изменяя только одну координату, просматривается условие на выход за пределы координаты и удовлетворению цветового условия. После того, как длина и высота посчитана - можно посчитать площадь, как произведение длины на высоту. Сразу идет сравнение на максимальную площадь. Если это самая большая на момент проверки, то максимум обновляется и записываются координаты начала фигуры и конца в кортежи. После прохождения по всему изображению - перекрашивается прямоугольник по полученным координатам и возвращается изображение.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	user_func(image, 1,1,100,100,”green”,5)	image	OK
2.	set_black_white(image, 10, 10, 100, 100)	image	OK
3.	find_rect_and_recolor(image, “red”, “green”)	image	OK

Выводы

Была написана программа, использующая основные функции для работы с изображениями с использованием библиотекой Pillow.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Klyukin_Aleksandr_lb2.py

```
import PIL
import numpy as np
import math
from PIL import Image, ImageDraw
def user_func(image, x0, y0, x1, y1, fill, width):
    draw = ImageDraw.Draw(image)
    draw.line(((x0, y0), (x1, y1)), fill, width)
    return image

def check_coords(image, x0, y0, x1, y1):
    width, height = image.size
     if (x0 < x1 and y0 < y1) and (x0 >= 0 and y0 >= 0) and (x1 <=
width and y1 <= height):
        return True
    else:
        return False

def set_black_white(image, x0, y0, x1, y1):
    check = check_coords(image, x0, y0, x1, y1)
    if check is True:
        img = image.crop((x0, y0, x1, y1))
        img = img.convert('1')
        image.paste(img, (x0, y0))
        return image
    else:
        return image

def find_rect_and_recolor(image, old_color, new_color):
    picture = image.load()
    width, height = image.size
    coords_start = (0, 0)
    coords_end = (0, 0)
    max_area = 0
    for x in range(width):
        for y in range(height):
            if picture[x, y] == old_color:
                x_new = x
                y_new = y
                     while x_new < width and picture[x_new, y] ==
old_color:
                    x_new += 1
                     while y_new < height and picture[x, y_new] ==
old_color:
                    y_new += 1
                if (x_new - x) * (y_new - y) > max_area:
                    max_area = (x_new - x) * (y_new - y)
                    coords_start = (x, y)
                    coords_end = (x_new-1, y_new-1)
    draw = ImageDraw.Draw(image)
    draw.rectangle((coords_start, coords_end), new_color)
    return image
```