**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 |  | Гребнев Е.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*x*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.

1. 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.
2. 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 |
|  | 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. |
|  | 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. |
|  | 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 ImageDraw, Image

import numpy as np

from collections import Counter

from itertools import product

def triangle(img: Image, x0: int, y0: int, x1: int, y1: int, x2: int, y2:int , thickness: int, color: list, fill\_color=None) -> Image:

draw = ImageDraw.Draw(img)

points = [(x0, y0), (x1, y1), (x2, y2)]

draw.polygon(points, width=thickness, outline=tuple(color), fill=tuple(fill\_color) if fill\_color else None)

return img

def change\_color(img: Image, color: list) -> Image:

data = np.array(img)

common\_color = Counter(map(tuple, data.reshape(-1, 3))).most\_common(1)[0][0]

data[(data == common\_color).all(axis=-1)] = color

return Image.fromarray(data, mode='RGB')

def collage(img: Image, N: int, M: int) -> Image:

new\_bg = Image.new("RGB", (img.width \* M, img.height \* N), (0, 0, 0))

for n, m in product(range(N), range(M)):

new\_bg.paste(img, (img.width \* m, img.height \* n))

return new\_bg