

Работа с изображениями

Цель

Овладеть инструментами обработки и создания изображений с использованием соответствующих библиотек для дальнейшего создания приложений обработки изображений и анализа данных

Что нужно сделать

Решите следующие задачи на языке программирования Python

01 Магия фотофильтров

Вы решили стать волшебником фотофильтров и создать свой уникальный эффект для изображений. Для этого требуется написать функцию `apply_magic_filter(image, brightness, color, contrast)`, которая принимает на вход изображение и параметры для усиления яркости, цвета и контраста.

Пошаговый план решения задачи:

1 шаг

Загрузите модули `Image`, `ImageEnhance` из библиотеки PIL

2 шаг

Создайте функцию с требуемым заголовком

3 шаг

Внутри функции последовательно для яркости, цвета и контраста примените конструкцию:

```
image = ImageEnhance.<тип фильтра>(image).enhance(<степень  
усилений>)
```

`<тип фильтра>` - это `Brightness`, `Color` или `Contrast` соответственно

`<степень усиления>` - это положительное число типа `float`, которое вы передаете в качестве параметров функции

4 шаг

Верните новое измененное изображение как результат работы функции

5 шаг

Протестируйте работу функции на произвольном изображении с различными параметрами

Оригинальное изображение



Измененное изображение



03 Мозаика из пикселей

Вы решили создать свою собственную мозаичную картину, используя изображение и технику пиксельного искусства. Нужно разбить входное изображение на маленькие квадратные блоки, каждый из которых будет представлять определенный цвет и покрасить весь блок в требуемый цвет. Для этого требуется написать функцию `create_pixel_art(image, block_size)`, которая принимает на вход изображение и размер стороны квадрата пикселизации. Излишнюю часть изображения справа и снизу требуется откадрировать с помощью функции `crop` таким образом, чтобы ширина и высота были кратными `block_size`.

Пошаговый план решения задачи:

1 шаг

Загрузите изображение с помощью PIL и передайте в функцию

2 шаг

Установите размер блоков (например, 10x10 пикселей) для разбиения изображения

3 шаг

Обрежьте правую и нижнюю части входного изображения так, чтобы ширина и высота были кратными размеру блока

4 шаг

Пройдите по каждому блоку изображения и найдите средний цвет блока

5 шаг

Замените все пиксели внутри блока на этот средний цвет

6 шаг

Верните новое изображение в виде мозаичной картины

Примечание:

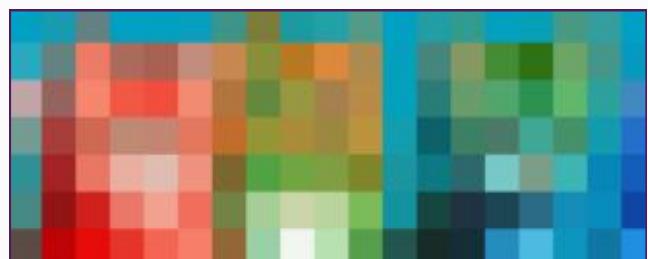
Используйте вложенные циклы для обхода блоков, функцию `Image.crop(box)` для обрезания очередного блока изображения, здесь `box = (x, y, x + block_size, y + block_size)` является прямоугольником, задающим границы блока.

Для расчета среднего цвета пикселя, требуется пройтись по всем пикселям блока и отдельно рассчитать средние арифметические значения для красного, зеленого и синего каналов внутри блока. Затем используйте функцию `Image.paste((r, g, b), box)` для вставки нужного цвета в блоке.

Оригинальное изображение



Измененное изображение



03 Геометрический абстракционизм

Вы решили создать геометрическую картину. Задайте размеры холста и превратите его в абстрактную композицию, используя разноцветные прямоугольники и эллипсы.

Пошаговый план решения задачи:

1 шаг

Загрузите модуль `ImageDraw` из библиотеки `PIL`, загрузите библиотеку `random`. Задайте случайное зерно `random.seed(777)` для воспроизводимости результатов

2 шаг

Создайте функцию `create_geometric_art(width, height)`, которая принимает на вход размеры нового изображения, а возвращает нарисованную композицию

3 шаг

Внутри функции создайте новое изображение заданного размера:
`blank_image = Image.new("RGB", (width, height), "white")`

4 шаг

Создайте объект для рисования на изображении
`draw = ImageDraw.Draw(blank_image)`

5 шаг

В цикле последовательно создайте 100 элементов композиции и нанесите их на холст с помощью функций `draw.rectangle` и `draw.ellipse`

6 шаг

Все параметры расположения фигур, а также их цвет должны быть сгенерированы случайным образом.

Для генерации левого верхнего угла фигуры используйте конструкцию `x, y = random.randint(0, width - 1), random.randint(0, height - 1)`

Для генерации ширины и высоты фигуры используйте `size_x, size_y = random.randint(20, 100), random.randint(20, 100)`

7 шаг

Сгенерируйте случайный цвет `color = (random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))`

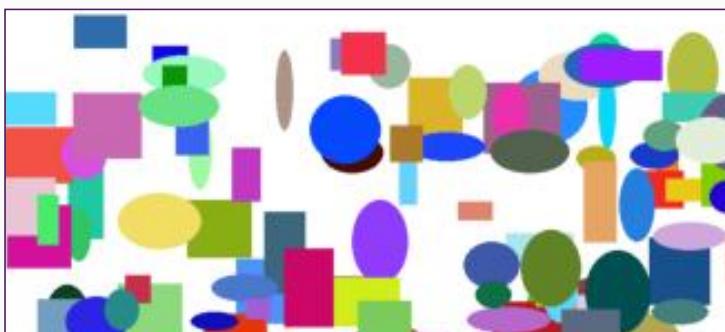
8 шаг

Сгенерируйте целое число: 0 или 1. Если выпал 0, рисуйте эллипс с координатами `(x, y, x + size_x, y + size_y)` и сгенерированным цветом (параметр `fill`). Если выпала 1, нарисуйте прямоугольник

9 шаг

Верните полученное изображение в качестве результата работы функции

Сгенерированное изображение (900, 400)



Результат

В ответе приложите файл с расширением ru для каждой задачи

Критерии оценивания

K1 Решена верно 1 задача с выполнением всех условий **2 балла**

K2 Решено верно 2 задачи с выполнением всех условий **3 балла**

K3 Решено верно 3 задачи с выполнением всех условий **4 балла**

Максимальное количество баллов **4 балла**

Минимальное количество баллов
чтобы преподаватель смог зачесть вашу работу **2 балла**