

Лабораторная работа «Преобразование изображения»

Цель: Освоить основные способы трансформации изображения с использованием языка программирования Python и библиотеки PIL.

Используйте лекционный материал

<https://portal.edu.asu.ru/mod/resource/view.php?id=320189>.

Задание

Возьмите полученную в 1-й лабораторной работе картинку, а также выберите на портале <https://pixabay.com/> со свободно распространяемыми графическими файлами не менее 3х изображений разной тематики и цветовой гаммы (можно использовать любой другой подобный ресурс со ссылкой на него, либо изображения собственного авторства, хотя бы одна картинка должна быть пейзажем).

Вариант 1. Серое изображение

1. Получите условно серое изображение: выделив все 3 RGB-канала, получите результат их усреднения, а затем запишите его вместо каждого из изначальных цветовых каналов. Выполните это для всех своих изображений.
2. Получите Grayscale-изображения (в градациях серого) с использованием формулы $Y = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B$, часто применяемой в различных графических пакетах для такого преобразования. Здесь R , G и B соответствуют красному, зеленому и синему каналам.
3. Получите Grayscale-изображения (в градациях серого) с использованием метода `grayscale` из модуля `ImageOps` библиотеки PIL.
4. Проанализируйте полученные изображения. Есть ли разница в картинках, полученных с помощью различных применённых преобразований?
5. Используя метод `transpose`, отразите свои изображения слева на право

Вариант 2. Черно-белое изображение с порогом. Сепия

1. Получите черно-белое изображения по вводимому пользователем с клавиатуры порогу. Выполните это для всех своих изображений. Учтите, что итоговое значение для каждого канала должно быть в интервале от 0 до 255.

2. Получите сепия-изображения вводимой пользователем с клавиатуры глубины.
3. Проанализируйте полученные изображения. Как влияют на результат вводимые пользователем числа?
4. Повысьте контраст исходных и полученных изображений с использованием метода `autocontrast` из модуля `ImageOps` библиотеки `PIL`. Поэкспериментируйте с разными входными параметрами, прокомментируйте результаты.
5. Используя метод `transpose`, поверните свои изображения на 270 градусов против часовой стрелки.

Вариант 3. Инверсия цветов (негатив). Колоризация серого изображения на основе пары опорных цветов

1. Поменяйте цвета своих изображений на противоположные в RGB используя прямые вычисления значений для каждого из каналов.
2. Получите инвертированные по цвету изображения с использованием метода `invert` из модуля `ImageOps` библиотеки `PIL`.
3. Получите Grayscale-изображения (в градациях серого) с использованием метода `grayscale` из модуля `ImageOps` библиотеки `PIL`, а затем раскрасьте полученные изображения методом `colorize` из того же модуля, используя в качестве пары базовых цветов не белый и черный, какой-то другой набор. Затем поменяйте эти цвета местами, проанализируйте полученные результаты, сделайте выводы. Попробуйте после `colorize` использовать метод `invert` из модуля `ImageOps`, что получится в этом случае?
4. Используя метод `transpose`, поверните свои изображения на 90 градусов против часовой стрелки.
5. Создайте bmp-изображения с 8, 12, 16 и 24 цветами из ранее полученных картинок.

Вариант 4. Добавление шумов. Размытие с помощью фильтров

1. Используя датчик случайных чисел и заданный пользователем с клавиатуры уровень шума, добавьте в каждый канал своих изображений равномерно распределенный шум. Учтите, что итоговое значение для каждого канала должно быть в интервале от 0 до 255.
2. Примените метод `filter` из модуля `ImageFilter` библиотеки `PIL` для размытия изображений. Сравните работу фильтров `BLUR`, `SMOOTH`, `SMOOTH_MORE`, `GaussianBlur` (поварьируйте радиус размытия).
3. К исходным изображениям примените фильтры `CONTOUR`, `FIND_EDGES` и `EMBOSS`. Сравните работу этих фильтров.
4. Наложите одно из исходных изображений на другое, управляя их прозрачностью. Помните, что накладываемое изображение не должно иметь размеры больше картинки-подложки.
5. Проведите кадрирование исходных и полученных изображений с помощью метода `crop` с разными входными параметрами, запрошенными у пользователя.

Вариант 5. Управление яркостью изображения. Фильтры резкости и контраста

1. Запросите у пользователя параметр регулирования яркости (целое не обязательно положительное число) для всех своих изображений. Затем в каждый цветовой канал добавьте это значение. Если оно > 0 , то картинка станет ярче, иначе – темнее. Учтите, что итоговое значение для каждого канала должно быть в интервале от 0 до 255.
2. Примените метод `filter` из модуля `ImageFilter` библиотеки `PIL`, проанализируйте и сравните работу фильтров для повышения резкости `SHARPEN`, `DETAIL`, `EDGE_ENHANCE`, `EDGE_ENHANCE_MORE`.
3. Создайте из каждого полученного изображения с помощью метода `Contrast` из модуля `ImageEnhance` библиотеки `PIL` объект, для которого повысьте контраст применив `enhance` с разными входными параметрами.
4. Используя метод `transpose`, отразите свои изображения сверху вниз.

5. Поменяйте размеры исходных и полученных изображений с помощью метода `resize` с разными входными параметрами. Запросите у пользователя высоту целевого изображения (в пикселях), а ширину вычислите пропорционально введённой высоте.