Работа с изображениями

01 Библиотека Pillow

В Python для работы с изображениями используется библиотека Pillow. Вот некоторые основные операции, которые вы можете выполнять с помощью Pillow:

01 Установка библиотеки Pillow

```
pip install pillow
```

02 Открытие изображения

```
from PIL import Image

# Открываем изображение
image = Image.open('image.jpg')
```

03 Показ изображения

```
image.show()
```

04 Изменение размера изображения

```
# Меняем размер изображения на 300x200 пикселей
resized_image = image.resize((300, 200))
```

05 Извлечение информации об изображении

```
width, height = image.size
format = image.format
mode = image.mode
```

06 Конвертация изображения в другой формат

```
# Сохраняем изображение в формате PNG
image.save('new_image.png', 'PNG')
```

07 Применение фильтров к изображению

```
from PIL import ImageFilter

# Применяем размытие к изображению
blurred_image = image.filter(ImageFilter.BLUR)
```

08 Изменение яркости, контраста и цветового баланса

```
from PIL import ImageEnhance

# Изменяем яркость изображения на 1.5
enhancer = ImageEnhance.Brightness(image)
bright_image = enhancer.enhance(1.5)
```

09 Обрезка изображения

```
# Обрезаем изображение с координатами верхнего левого и нижнего правого углов
cropped_image = image.crop((100, 100, 400, 400))
```

10 Поворот изображения

```
from PIL import Image

# Открываем изображение
image = Image.open('image.jpg')

# Поворачиваем изображение на 45 градусов по часовой стрелке
rotated_image = image.rotate(45)

# Сохраняем повернутое изображение
rotated_image.save('rotated_image.jpg')
```

В этом примере мы открываем изображение с помощью Image.open, затем используем метод rotate для поворота изображения на указанный угол. Повернутое изображение сохраняется с помощью метода save. Вы можете указывать положительные значения угла для поворота по часовой стрелке или отрицательные значения угла для поворота против часовой стрелки. Также возможно указать дополнительные параметры, например, expand=True, чтобы увеличить размер холста и сохранить всю повернутую картинку

11 Изменение цвета изображения

Для изменения цвета изображения в Python с использованием библиотеки Pillow, вы можете воспользоваться методом convert() или ImageMath из этой библиотеки. Вот примеры:

Изменение на черно-белый

```
from PIL import Image

# Открываем изображение
image = Image.open('image.jpg')

# Преобразуем в черно-белый цвет
black_white_image = image.convert('L')

# Сохраняем новое изображение
black_white_image.save('black_white_image.jpg')
```

Изменение на оттенки серого

```
from PIL import ImageOps

# Открываем изображение
image = Image.open('image.jpg')

# Преобразуем в оттенки серого
gray_image = ImageOps.grayscale(image)

# Сохраняем новое изображение
gray_image.save('gray_image.jpg')
```

Изменение на сепию

```
from PIL import ImageOps

# Открываем изображение
image = Image.open('image.jpg')

# Преобразуем в сепию
sepia_image = ImageOps.colorize(image.convert('L'), '#704214', '#C0A080')

# Сохраняем новое изображение
sepia_image.save('sepia_image.jpg')
```

Каждый из этих примеров демонстрирует различные способы изменения цветовой палитры изображения с использованием библиотеки Pillow. Метод convert() предоставляет различные режимы изменения цвета, такие как "L" для черно-белого, "RGB" для оттенков серого и другие

ImageMath предоставляет более гибкий способ манипулирования пикселями изображения. Обе эти функции позволяют сохранять измененные изображения с помощью метода save()

Это лишь некоторые примеры операций, которые можно выполнять с изображениями с помощью библиотеки Pillow. Pillow предлагает еще множество других методов и возможностей для работы с изображениями, включая манипуляции, фильтрацию, добавление текста и многое другое. Более подробную информацию и примеры вы можете найти в документации библиотеки Pillow

02 Библиотека Matplotlib

Библиотека Matplotlib также предоставляет возможности для работы с изображениями. Вот некоторые операции, которые можно выполнить при работе с изображениями с помощью Matplotlib:

01 Открытие и отображение изображения:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

# Открываем и загружаем изображение
image = mpimg.imread('image.jpg')

# Отображаем изображение
plt.imshow(image)
plt.show()
```

02 Изменение размера изображения

```
resized_image = mpimg.imresize(image, (300, 200))
plt.imshow(resized_image)
plt.show()
```

03 Применение фильтра к изображению

```
from skimage import filters

edges = filters.sobel(image)
plt.imshow(edges, cmap='gray')
plt.show()
```

04 Сохранение изображения

```
mpimg.imsave('filtered_image.jpg', edges, cmap='gray')
```

05 Отображение изображения с несколькими подобными графиками

```
fig, axes = plt.subplots(2, 2)
axes[0, 0].imshow(image)
axes[0, 1].imshow(resized_image)
axes[1, 0].imshow(edges, cmap='gray')
axes[1, 1].imshow(resized_image)
plt.show()
```

06 Добавление текста или меток на изображение

```
plt.imshow(image)
plt.text(50, 50, 'Text', fontsize=12, color='red')
plt.xlabel('X Label')
plt.ylabel('Y Label')
plt.title('Title')
plt.show()
```

Это лишь несколько примеров операций работы с изображениями при использовании библиотеки Matplotlib. Matplotlib также предоставляет множество других функций и параметров для настройки визуализации, включая цветовую карту, прозрачность, размер и положение элементов графика, подписи осей и другие. Более подробную информацию и примеры использования вы можете найти в документации Matplotlib

03 Библиотека Requests

Библиотека requests не предназначена непосредственно для работы с изображениями, но она может использоваться для загрузки изображений из внешних источников. Вот пример использования библиотеки requests для загрузки изображения

01 Установка библиотеки requests

```
pip install requests
```

02 Загрузка и сохранение изображения

```
import requests

url = "https://example.com/image.jpg"
response = requests.get(url)

with open("image.jpg", "wb") as file:
    file.write(response.content)
```

В этом примере мы определяем URL изображения, затем используем метод get из requests для получения содержимого изображения. После этого мы открываем файл в режиме записи бинарного файла ("wb") и записываем содержимое ответа в файл с помощью метода write

Обратите внимание, что при использовании библиотеки requests для загрузки изображений, ваш код должен быть готов обрабатывать ошибки и обеспечивать правильную обработку ответов сервера, проверку доступности сети и другие аспекты, связанные с сетевыми операциями

Итог

Библиотека Pillow является одной из наиболее популярных и простых в использовании библиотек для работы с изображениями. С ее помощью вы можете открывать и сохранять изображения, изменять размеры, применять фильтры, изменять цветовую палитру, обрезать изображения, добавлять текст и рисовать на изображениях