BAKI ALİ NEFT MƏKTƏBİ
BAKU HIGHER OIL SCHOOL

Image Processing

Image Processing in Python using Pillow

Instructor : PhD, Associate Professor Leyla Muradkhanli

What is Pillow?

Pillow is a powerful library for processing images in Python. Pillow supports a range of image file formats, such as .PNG, .JPEG, .PPM, .GIF, .TIFF, and .BMP. You can perform various operations on images such as cropping, resizing, adding text, rotating, grayscaling, and so much more using this library.

Installation and project setup

You can install Pillow using pip, a package manager for Python packages:

```
pip install pillow
```

or

```
pip install PIL
```

Image Processing

To get started, first import the Image object to the Python file.

```
from PIL import Image
```

Next, load the image by calling the Image.open() function, which returns a value of the Image object data type.

```
image = Image.open('Flower.jpg')
```

Open and Display an Image

```
from PIL import Image
# Open an image
img = Image.open("example.jpg")

# Show image
img.show()
```

Properties of the Image object

There are several properties of the image we can access to get more data from the image:

image.width returns the width of the image

image.height returns the height of the image

image.format returns the file format of the image (e.g., .JPEG, .BMP, .PNG, etc.)

image.size returns the tuple height and weight of the image

image.palette returns the color palette table, if one exists

image.mode returns the pixel format of the image (e.g., L, RGB, CMYK)

Basic Operations

Resize

```
resized = img.resize((200, 200))
resized.show()
```

Basic Operations

Crop

```
box = (100, 100, 300, 300)   # (left, upper, right, lower)
cropped = img.crop(box)
cropped.show()
```

Cropping images

The **crop()** function in Pillow requires the portion to be cropped as a rectangle. The method takes a box tuple that defines the position and size of the cropped region and returns an Image object representing the cropped image. The region is defined by a 4-tuple, where coordinates are (left, upper, right, lower).

```
image = Image.open('sample.jpg')
image.crop((200, 50, 450, 300))
image.save('sample_cropped.jpg')
```

In the example above, the first two values represent the starting position from the upper-left; the third and fourth values represent the distance in pixels from the starting position toward the right and bottom direction. The full size of the cropped image can be calculated as 250×250 pixels.

Color Conversions

```
gray = img.convert("L")   # Grayscale
bw = img.convert("1")     # Black & White
gray.show()
bw.show()
```

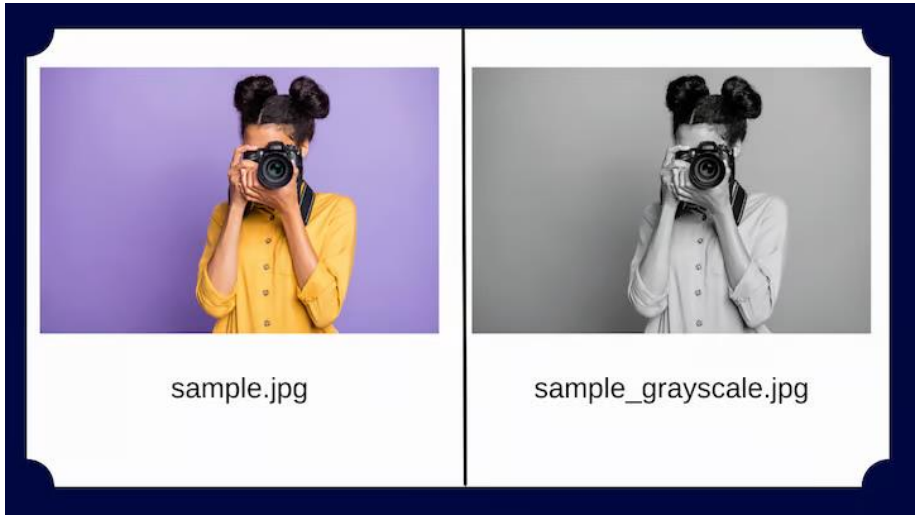
Color transformation

There are various forms of pixel representations, including L (luminance), RGB, and CMYK.

Pillow allows you to convert images between different pixel representations using the **convert()** method. The library supports transformations between each supported mode as well as the “L” and “RGB” modes. To convert between other modes, you may have to use an “RGB” image.

```
image = Image.open('sample.jpg')
grayscale_image = image.convert('L')
grayscale_image.save('sample_grayscale.jpg')
```

Using the convert function, the sample image is converted from RGB to L (luminance) mode, which will result in a grayscale image.



Basic image operations

Any changes made to the Image object can be saved to an image file with the **save()** method. All the rotations, resizing, cropping, drawing, and other image manipulations are done through via calls on this Image object.

Save Image

```
img.save("output.png")
```

Changing image formats

Pillow supports a wide variety of images formats. An image can be converted from one format to another as follows:

```
image = Image.open('sample.jpg')
image.save('sample_formatted.png')
```

First, the image is loaded. Then, Pillow sees the file extension specified as PNG, so it converts the image to .PNG before saving it to file.

Basic Operations

Rotate & Flip

```
rotated = img.rotate(45)      # Rotate 45 degrees
flipped = img.transpose(Image.FLIP_LEFT_RIGHT)
rotated.show()
flipped.show()
```

Flipping and rotating images

If you need the image to face a different direction, Pillow enables you to flip it. This is done using the transpose function, which takes any of the following parameters:

Image.FLIP_LEFT_RIGHT, which flips the image horizontally

Image.FLIP_TOP_BOTTOM, which flips the image vertically

Image.ROTATE_90, which rotates the image to a certain degree, depending on the angle

```
image = Image.open('sample.jpg')
Image_flip=image.transpose(Image.FLIP_TOP_BOTTOM)
Image_flip.save('sample_flip.jpg')
```

Flipping and rotating images

The resulting image is flipped vertically.

Flipping and rotating images

Alternatively, you can rotate images using the **rotate()** method. This takes an integer or float argument representing the degrees of rotation and returns a new Image object of the rotated image. The rotation is counterclockwise.

```
image = Image.open('sample.jpg')
Image_rotate=image.rotate(90)
Image_rotate.save('image_rotate90.jpg')
```

The image is rotated by an angle of 90 degrees.