Image Processing Tool

PIL→ PILLOW

By

Malepati Nagendra Deepak

Image Processing Tool

Image processing tool is a software tool that is designed to manipulate digital images Or crop an image to remove unwanted background clutter. Its purpose is to improve the contents and quality of an image.

USED IN: Field of Science and Technology include-

- Computer vision
- Remote sensing
- Feature extraction
- Face detection
- Forecasting
- Optical character recognition
- Finger-print detection
- Microscope imaging

PIL (Python Imaging Library)→PILLOW

- Its a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

- However, its development has stagnated, with its last release in 2009. Fortunately, there is **Pillow**, an actively developed fork of PIL, that is easier to install, runs on all major operating systems, and supports Python 3.

- The library contains basic image processing functionality, including point operations, filtering with a set of built-in convolution kernels, and color-space conversions.

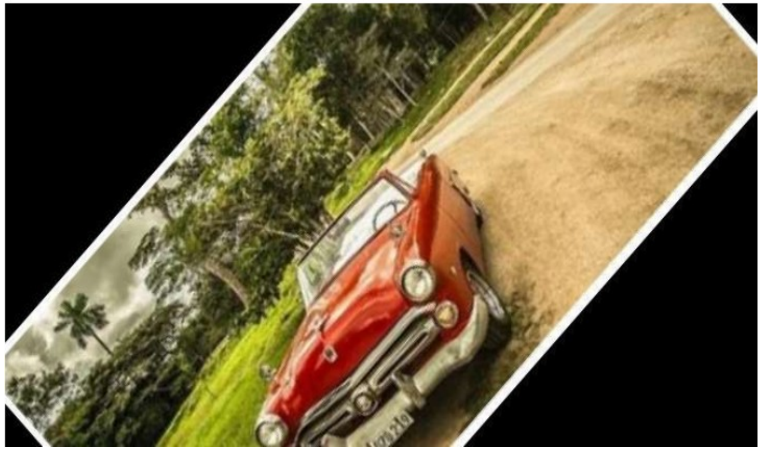
Python Pillow - Using Image Module

Opening, rotating and displaying an image

To load the image, we simply import the image module from the pillow and call the **Image.open()**, passing the image filename.

to load the image by calling the **Image.open() function**, which returns a value of the Image object data type. Any modification we make to the image object can be saved to an image file with the **save()** method.

The image object we received using **Image.open()**, later can be used to resize, crop, draw or other image manipulation method calls on this Image object.



Attributes of Image Module

The instance of the Image class has some attributes. Let's try to understand few of them by example −

Image.filename

This function is used to get the file name or the path of the image.

Image.format

This function returns file format of the image file like 'JPEG', 'BMP', 'PNG', etc.

Image.size
It returns the tuple consist of height & weight of the image.

Image.mode

It is used to get the pixel format used by the image. Typical values are "1", "L", "RGB" or "CMYK".

Image.info

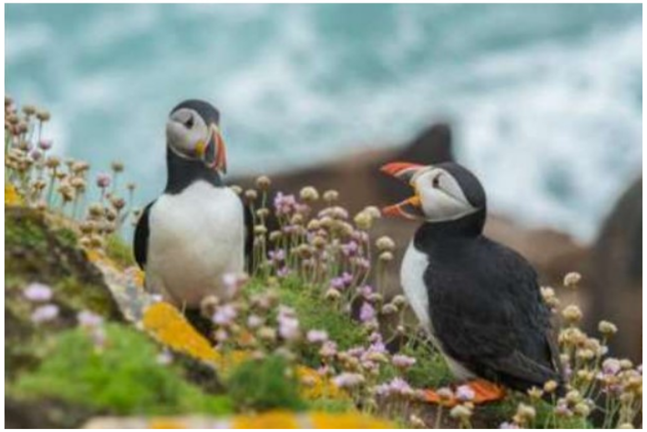
It returns a dictionary holding data associated with the image.

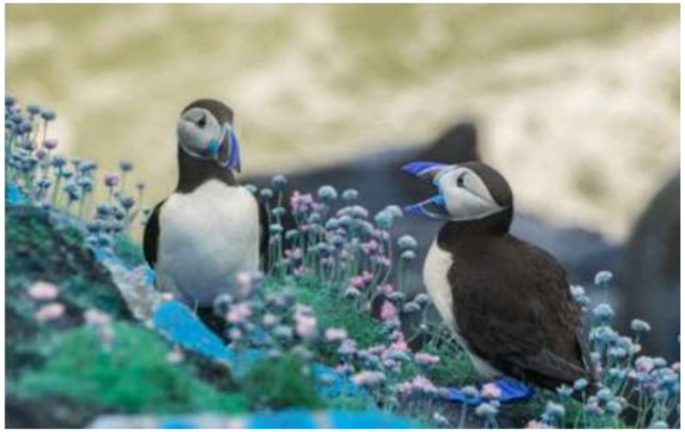
Python Pillow - Merging Images

Pillow package allows you to paste an image onto another one. The merge() function accepts a mode and a tuple of images as parameters, and combines them into a single image.

Merge the RGB bands:

- Using the merge() function, you can merge the RGB bands of an image
- On executing the above piece of code, you can see the original image and the image with merge the RGB bands as shown below

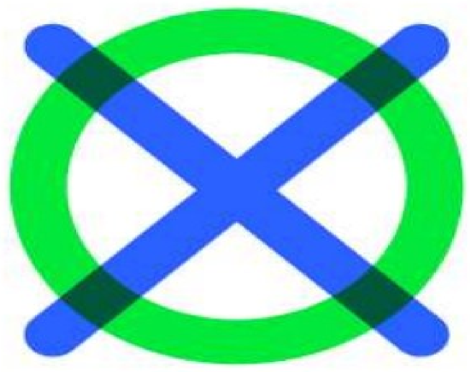




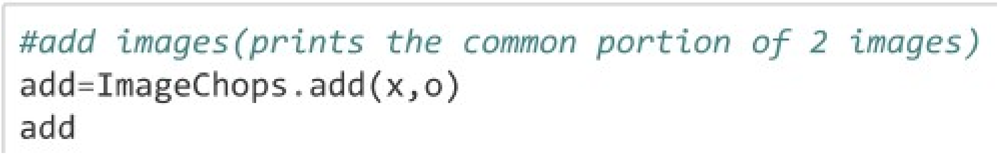
Merging Images

Merging two images:

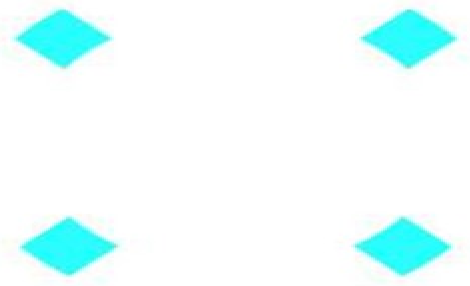
- In the same way, to merge two different images, you need to −
- Create image object for the required images using the open() function.
- While merging two images, you need to make sure that both images are of same size. Therefore, get each sizes of both images and if required, resize them accordingly.



Add Two Images

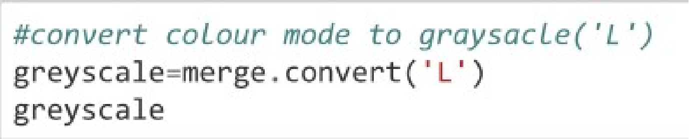


```
#add images(prints the common portion of 2 images)
add=ImageChops.add(x,o)
add
```

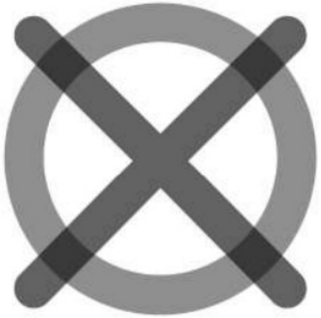


Grayscale and Binary Scale

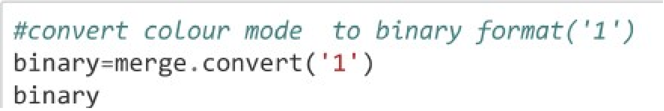
Conversion To GreyScale



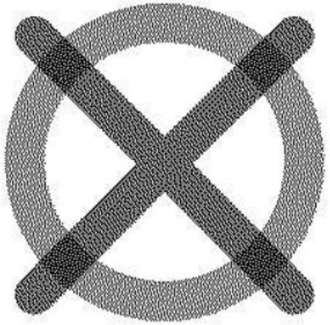
```
#convert colour mode to graysacle('L')
greyscale=merge.convert('L')
greyscale
```



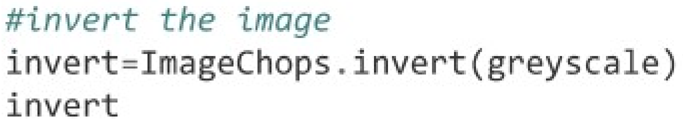
Conversion To BinaryFormat



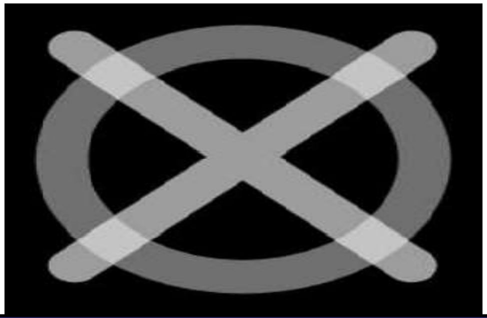
```
#convert colour mode  to binary format('1')
binary=merge.convert('1')
binary
```



Invert the image



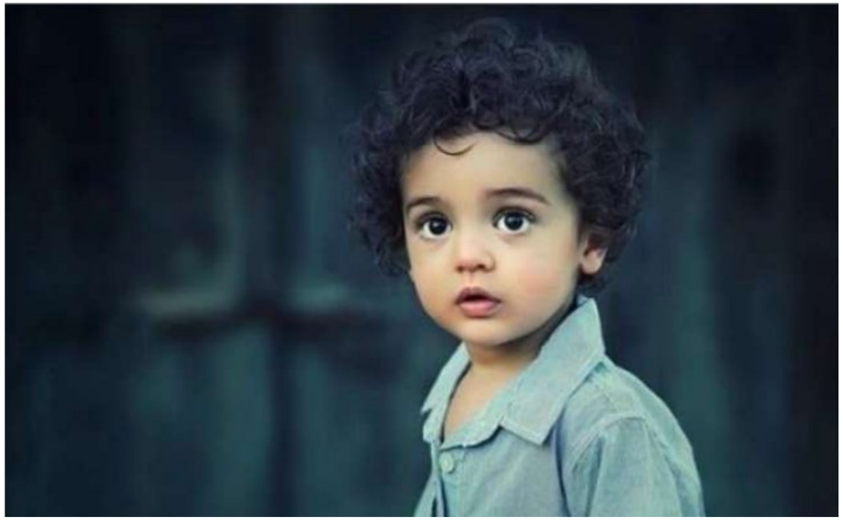
```
#invert the image
invert=ImageChops.invert(greyscale)
invert
```



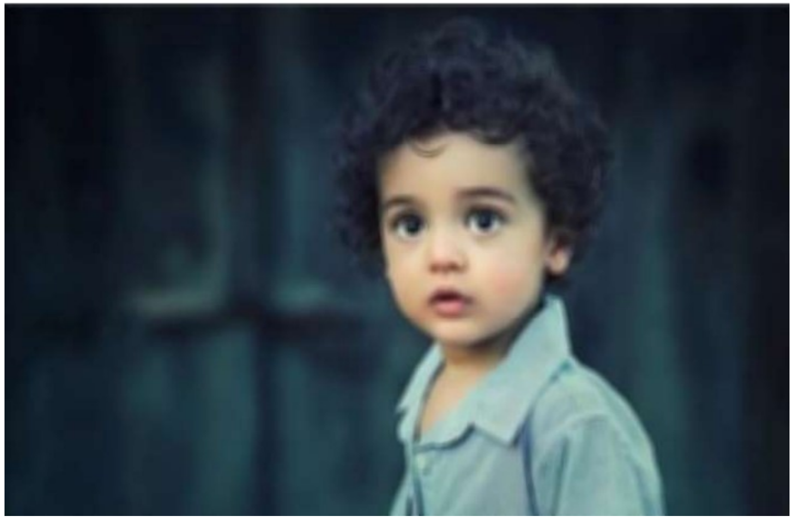
Blur an Image

Simple blur:

It applies a blurring effect on to the image as specified through a specific kernel or a convolution matrix



Original Image



Simple Blur

Blur an Image

Box blur

In this filter, we use ‘radius’ as parameter. Radius is directly proportional to the blur value

Gaussian blur

This filter also uses parameter radius and does the same work as box blur with some algorithmic changes. In short, changing the radius value, will generate different intensity of ‘Gaussian blur’ images.

Cropping an Image

- ❑ Cropping is one of the important operations of the image processing to remove unwanted portions of an image as well as to add required features to an image. It is widely used process in web applications, for uploading an image.

- ❑ The crop() function of the image class in Pillow requires the portion to be cropped as rectangle.

Flip and Rotate Images

Image module of the pillow library allows us to flip an image very easily. We are going to use the transpose (method) function from the Image module for flipping the images.

- **Image.FLIP_LEFT_RIGHT** − For flipping the image horizontally
- **Image.FLIP_TOP_BOTTOM** − For flipping the image vertically
- **Image.ROTATE_90** − For rotating the image by specifying degree

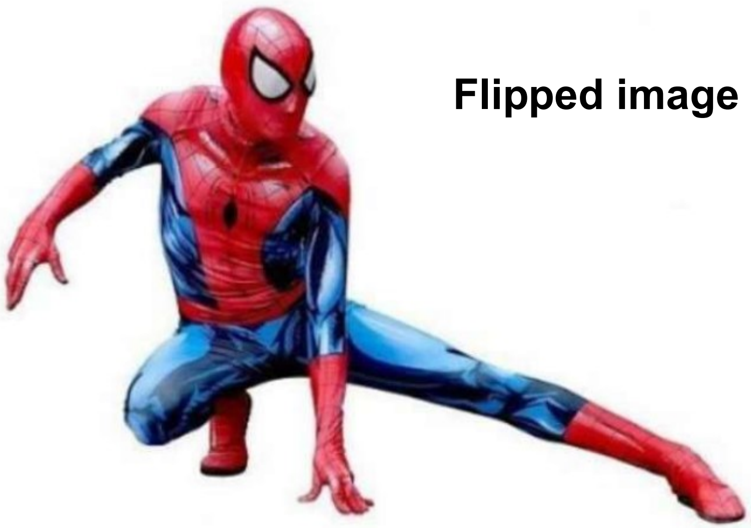
Horizontally flipped Image

Following Python example reads an image, flips it horizontally, and displays the original and flipped image using standard PNG display utility

Original image

→

Flipped image



Flip and Rotate Images

Vertically Flipped Image

Following Python example reads an image, flips it vertically, and displays the original and flipped image using standard PNG display utility

Rotate Image to a specific degree

Following Python example reads an image, rotates to a specified degree, and displays the original and rotated image using standard PNG display utility

Resizing an Image

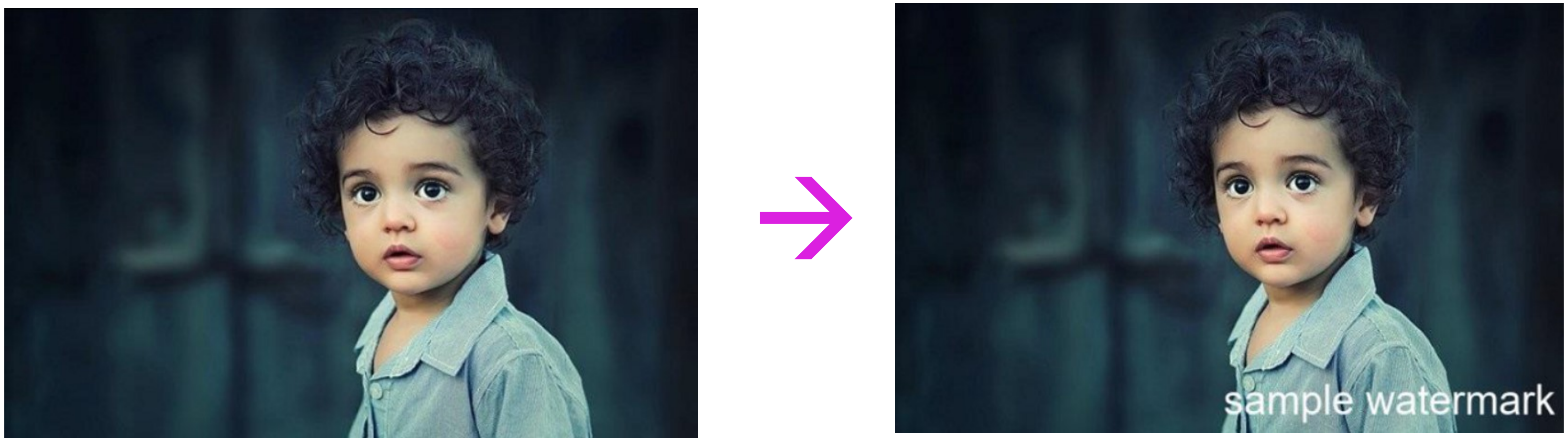
Most of the digital image is a two-dimensional plane of pixels and it has a width and height. The Image module from pillow library has an attribute size. This tuple consists of width and height of the image as its elements. To resize an image, you call the resize() method of pillow's image class by giving width and height.

Original Image

Resized Image

Creating a Watermark

- You have noticed that, some of the online photos are watermarked. Watermark is definitely one of the better ways to protect your images from misuse. Also, it is recommended to add watermark to your creative photos, before sharing them on social media to prevent it from being misused.
- Pillow package allows us to add watermarks to your images. For adding watermark to our image, we need **“Image”**, **“ImageDraw”** and **“ImageFont”** modules from pillow package.
- The ‘ImageDraw’ module adds functionality to draw 2D graphics onto new or existing images. The ‘ImageFont’ module is employed for loading bitmap, TrueType and OpenType font files.



Adding Filters to an Image

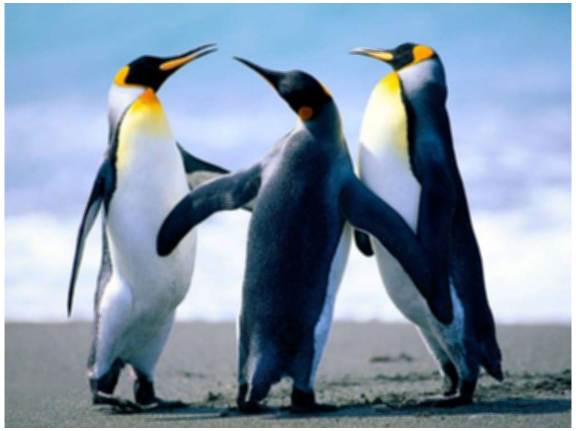
BLUR

Following python example applies the blur filter on an image saves it and, displays it using standard PNG display utility

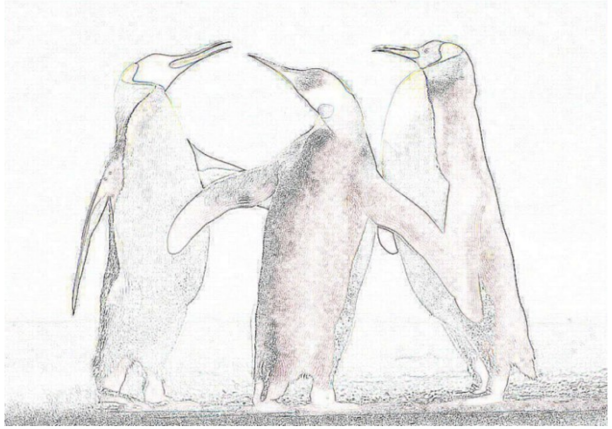
CONTOUR

If you save the above program and execute, it shows the original image and, the CONTOUR filtered image using standard PNG display utility, as follows

Original Image



BLUR

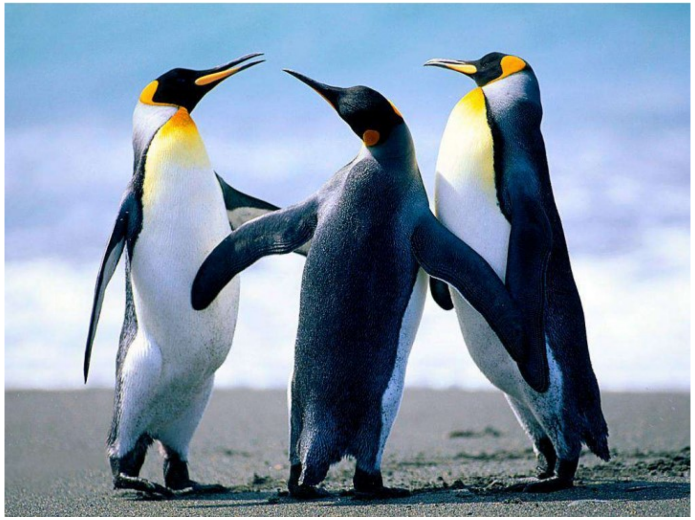


CONTOUR

Adding Filters to an Image

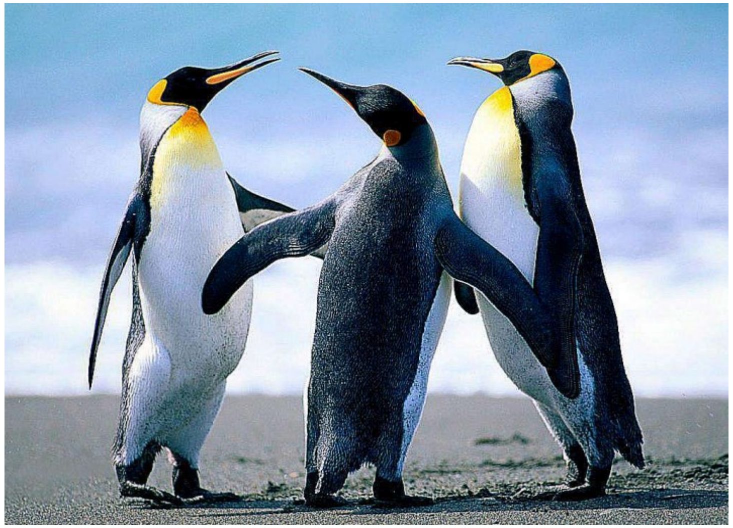
DETAIL

If you save the above program and execute, it shows the original image, and the DETAIL filtered image using standard PNG display utility, as follows



EDGE_ENHANCE

If you save the above program and execute, it shows the original image, and the EDGE_ENHANCE filtered image using standard PNG display utility, as follows



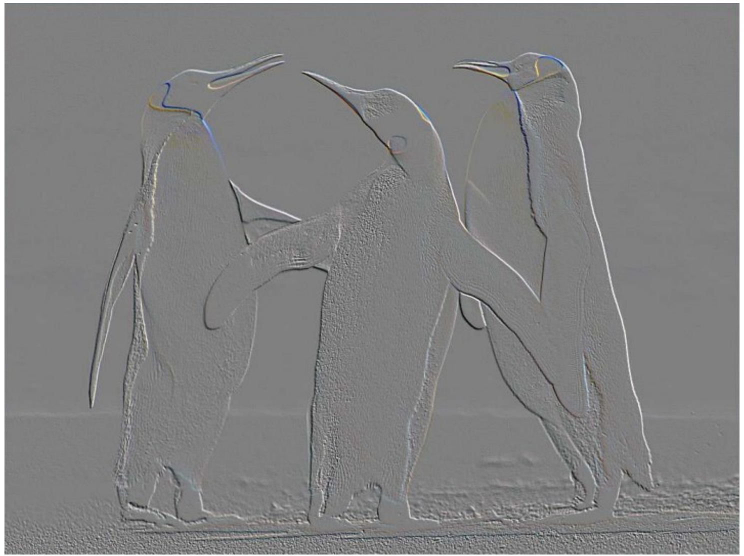
Adding Filters to an Image

EDGE_ENHANCE_MORE

If you save the above program and execute, it shows the original image, and the EDGE_ENHANCE _MORE filtered image using standard PNG display utility, as follows

EMBOSS

If you save the above program and execute, it shows the original image, and the EMBOSS filtered image using standard PNG display utility, as follows



Adding Filters to an Image

SHARPEN

If you save the above program and execute, it shows the original image, and the SHARPEN filtered image using standard PNG display utility, as follows

SMOOTH

If you save the above program and execute, it shows the original image, and the SMOOTH filtered image using standard PNG display utility, as follows

Adding Filters to an Image

SMOOTH_MORE

If you save the above program and execute, it shows the original image, and the SMOOTH_MORE filtered image using standard PNG display utility, as follows

Image Draw Module

The **‘ImageDraw’** module provides simple 2D graphics support for Image Object. Generally, we use this module to create new images, annotate or retouch existing images and to generate graphics on the fly for web use.
The graphics commands support the drawing of shapes and annotation of text.

- An image can be well-thought-out to be a two-dimensional array of pixels (picture elements). A pixel is the smallest dot of color being supported.
- The origin of the two-dimensional co-ordinate system used by ImageDraw, is in the **upper left corner** of the image.
- The pillow color schemes we use is RGB. The color RGB representation and support is provided by the module **ImageColor**.
- bitmap, OpenType or TrueType are the acceptable fonts for text annotations.
- Most of the drawing commands may require a bounding box parameter that specifies the area on the image to which the command is to be applied.
- A sequence of co-ordinates can be represented as [(x0, y0), (x1, y1),…(xn, yn)].
- For some drawing commands, we require angle values.

Image Draw Module

Drawing a line across the image:

If you save the above program as Example.py and execute, it draws a line across the image and displays it using standard PNG display utility, as follows

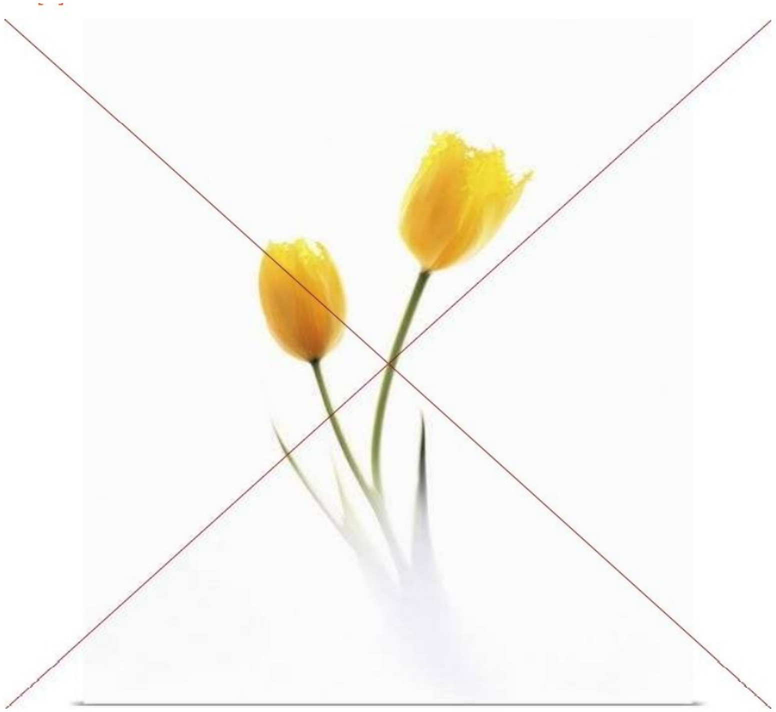


Image Draw Module

Drawing Shapes using ‘ImageDraw’ module(Rectangle, Ellipse)

Eclipse:

The **ellipse()** method draws the ellipse surrounded by bounding box xy on draw. The shape is filled using color fill and the perimeter in color outline. Default values of None are for the parameters fill and width which are optional.

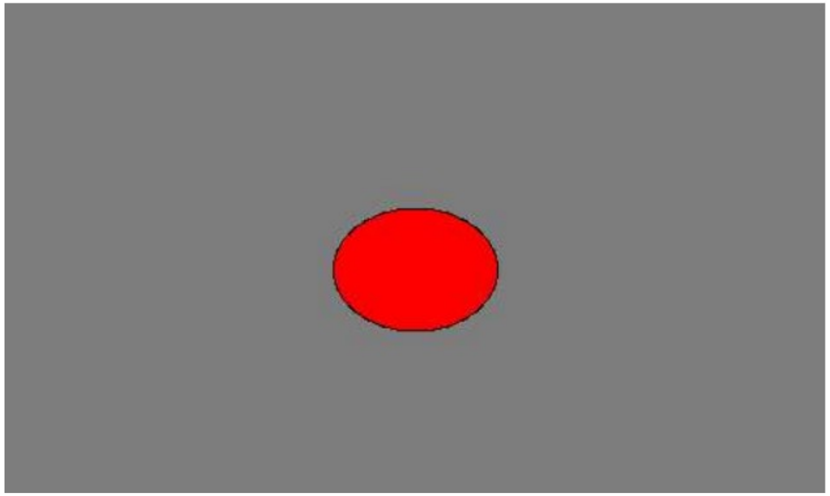


Image Draw Module

Drawing Shapes using ‘ImageDraw’ module(Rectangle, Ellipse)

Rectangle

The **rectangle()** method draws the rectangle given bounding box xy on draw. The shape is filled using color fill and the perimeter in color outline. Default values of None are for the parameters fill and width which are optional.

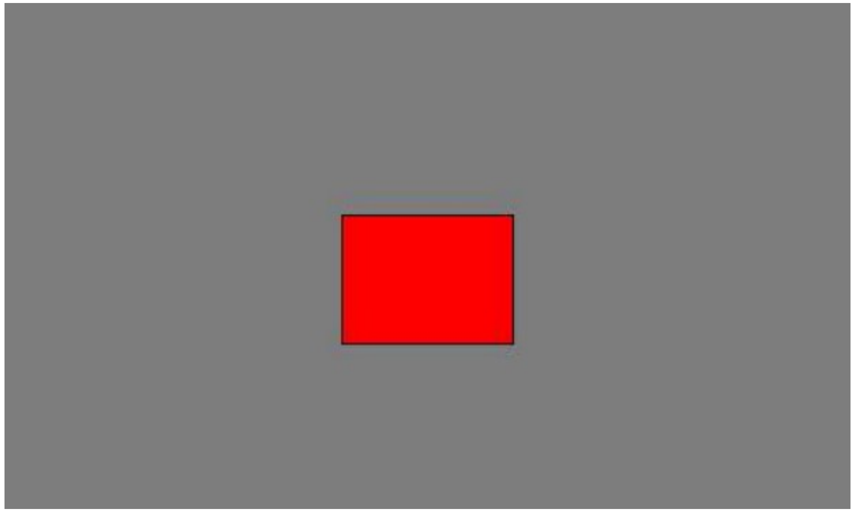


Image Draw Module

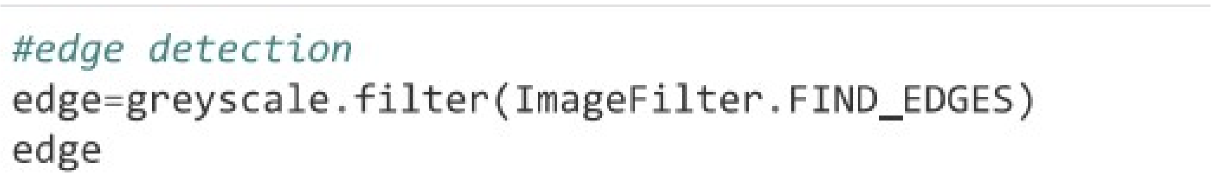
Writing Text on Image:

You can write text on images by passing the location of the text, the text itself and the color of the text. We can pass multiple other parameters to this method.

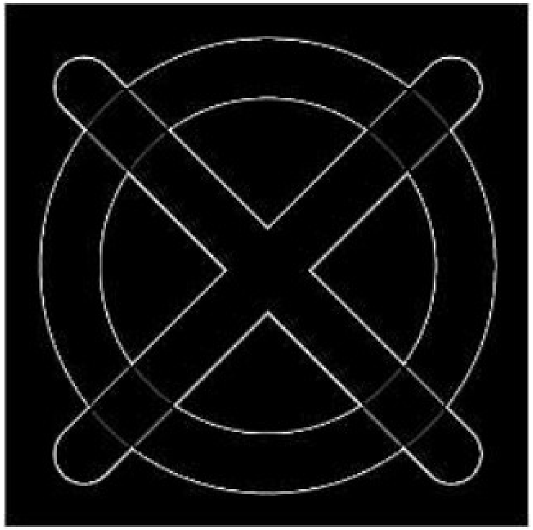
f you save the above program as .py and execute, it will add the given text on it, and displays it using the standard PNG display utility, as follows

Image Draw Module

Edge Detection



```
#edge detection
edge=greyscale.filter(ImageFilter.FIND_EDGES)
edge
```



Tool link and other reference materials

- ✓ https://www.youtube.com/watch?v=yiYqvRWykqE
- ✓ https://pypi.org/project/Pillow/2.2.1/
- ✓ https://readthedocs.org/projects/pillow/downloads/pdf/latest/

Thank you!