

JAVA DIP - IMAGE PYRAMIDS

http://www.tutorialspoint.com/java_dip/image_pyramids.htm

Copyright © tutorialspoint.com

Image pyramid is nothing but a method to display a multi-resolution image. The lowermost layer is a highest-resolution version of image and the topmost layer is a lowest-resolution version of the image. Image pyramids are used to handle image at different scales.

In this chapter we perform some down sampling and up sampling on images.

We use **OpenCV** functions **pyrUp** and **pyrDown**. They can be found under **Imgproc** package. Its syntax is given below:

```
Imgproc.pyrUp(source, destination, destinationSize);  
Imgproc.pyrDown(source, destination, destinationSize);
```

The parameters are described below:

Sr.No.	Parameters
1	source It is the source image.
2	destination It is the destination image.
3	destinationSize It is the size of the output image. By default, it is computed as <code>Size(src.cols * 2, src.rows * 2)</code> .

Apart from the pyrUp and pyrDown methods, there are other methods provided by the Imgproc class. They are described briefly:

Sr.No.	Methods
1	cvtColorMatsrc, Matdst, intcode, intdstCn It converts an image from one color space to another.
2	dilateMatsrc, Matdst, Matkernel It dilates an image by using a specific structuring element.
3	equalizeHistMatsrc, Matdst It equalizes the histogram of a grayscale image.
4	

filter2D*Matsrc, Matdst, intdepth, Matkernel, Pointanchor, doubledelta*

It convolves an image with the kernel.

5

GaussianBlur*Matsrc, Matdst, Sizeksize, doublesigmaX*

It blurs an image using a Gaussian filter.

6

integral*Matsrc, Matsum*

It calculates the integral of an image.

Example

The following example demonstrates the use of `Imgproc` class to perform up sampling and down sampling on an image.

```
import org.opencv.core.Core;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.Size;

import org.opencv.highgui.Highgui;
import org.opencv.imgproc.Imgproc;

public class main {
    public static void main( String[] args ) {

        try{

            System.loadLibrary( Core.NATIVE_LIBRARY_NAME );
            Mat source = Highgui.imread("digital_image_processing.jpg",
            Highgui.CV_LOAD_IMAGE_COLOR);

            Mat destination1 = new Mat(source.rows()*2, source.cols()*2, source.type());
            destination1 = source;

            Imgproc.pyrUp(source, destination1, new Size(source.cols()*2
source.rows()*2));
            Highgui.imwrite("pyrUp.jpg", destination1);

            source = Highgui.imread("digital_image_processing.jpg",
            Highgui.CV_LOAD_IMAGE_COLOR);

            Mat destination = new Mat(source.rows()/2, source.cols()/2, source.type());
            destination = source;
            Imgproc.pyrDown(source, destination, new Size(source.cols()/2,
source.rows()/2));
            Highgui.imwrite("pyrDown.jpg", destination);

        }catch (Exception e){
            System.out.println("error: " + e.getMessage());
        }
    }
}
```

Output

When you execute the given code, the following output is seen:

Original Image



On the original image, `pyrUpUPSampling` and `pyrDownDownSampling` are performed. The output after sampling is as shown below:

PyrUP Image



pyrDown Image





Loading [MathJax]/jax/output/HTML-CSS/jax.js