JAVA DIP - APPLYING BOX FILTER

We apply Box filter that blurs an image. A Box filter could be of dimensions 3x3, 5x5, 9x9 etc.

We use **OpenCV** function **filter2D** to apply Box filter to images. It can be found under **Imgproc** package. Its syntax is given below:

```
filter2D(src, dst, ddepth , kernel, anchor, delta, BORDER_DEFAULT );
```

The function arguments are described below:

Sr.No.	Arguments
1	**src** It is source image.
2	**dst** It is destination image.
3	**ddepth** It is the depth of dst. A negative value $suchas-1$ indicates that the depth is the same as the source.
4	**kernel** It is the kernel to be scanned through the image.
5	**anchor** It is the position of the anchor relative to its kernel. The location Point $-1,\ -1$ indicates the center by default.
6	**delta** It is a value to be added to each pixel during the convolution. By default it is 0.
7	**BORDER_DEFAULT** We let this value by default.

Apart from the filter2D method, there are other methods provided by the Imgproc class. They are described briefly:

Sr.No.	Methods

1

cvtColor*Matsrc, Matdst, intcode, intdstCn*

It converts an image from one color space to another.

2

dilate*Matsrc, Matdst, Matkernel*

It dilates an image by using a specific structuring element.

3

equalizeHist*Matsrc, Matdst*

It equalizes the histogram of a grayscale image.

4

filter2D*Matsrc, Matdst, intddepth, 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 apply Box filter to an image of Grayscale.

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

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

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

      try {
         int kernelSize = 9;
         System.loadLibrary( Core.NATIVE_LIBRARY_NAME );

         Mat source = Highgui.imread("grayscale.jpg",  Highgui.CV_LOAD_IMAGE_GRAYSCALE);
         Mat destination = new Mat(source.rows(),source.cols(),source.type());
         Mat kernel = Mat.ones(kernelSize,kernelSize, CvType.CV_32F);

         for(int i=0; i<kernel.rows(); i++){
            for(int j=0; j<kernel.cols(); j++){

               double[] m = kernel.get(i, j);

               for(int k =0; k<m.length; k++){
                  m[k] = m[k]/(kernelSize * kernelSize);
               }
               kernel.put(i,j, m);
```

```
            }
         }

         Imgproc.filter2D(source, destination, -1, kernel);
         Highgui.imwrite("output.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

In this example we convolve our image with the following filter *kernel*. This filter results in blurring an image as its size increases.

This original image has been convolved with the box filter of size 5, which is given below:

Box filter of size 5

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

Convolved Image *withBoxFilterofSize*5

Convolved Image *withBoxFilterofSize*9