

JAVA DIP - OPENCV GRAYSCALE CONVERSION

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

Copyright © tutorialspoint.com

In order to convert a color image to Grayscale image using OpenCV, we read the image into **BufferedImage** and convert it into **Mat** Object. Its syntax is given below:

```
File input = new File("digital_image_processing.jpg");
BufferedImage image = ImageIO.read(input);
//convert BufferedImage to Mat.
```

Then you can transform the image from RGB to Grayscale format by using method **cvtColor** in the **Imgproc** class. Its syntax is given below:

```
Imgproc.cvtColor(source mat, destination mat1, Imgproc.COLOR_RGB2GRAY);
```

The method **cvtColor** takes three parameters which are the source image matrix, the destination image matrix, and the color conversion type.

Apart from the **cvtColor** method, there are other methods provided by the **Imgproc** class. They are listed below:

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

filter2DMatsrc, Matdst, intdepth, Matkernel, Pointanchor, doubledelta

It convolves an image with the kernel.

5

GaussianBlurMatsrc, Matdst, Sizeksize, doublesigmaX

It blurs an image using a Gaussian filter.

6

integralMatsrc, Matsum

It calculates the integral of an image.

Example

The following example demonstrates the use of **Imgproc** class to convert an image to Grayscale:

```

import java.awt.image.BufferedImage;
import java.awt.image.DataBufferByte;

import java.io.File;
import javax.imageio.ImageIO;

import org.opencv.core.Core;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.imgproc.Imgproc;

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

        try {
            System.loadLibrary( Core.NATIVE_LIBRARY_NAME );
            File input = new File("digital_image_processing.jpg");
            BufferedImage image = ImageIO.read(input);

            byte[] data = ((DataBufferByte) image.getRaster().getDataBuffer()).getData();
            Mat mat = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC3);
            mat.put(0, 0, data);

            Mat mat1 = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC1);
            Imgproc.cvtColor(mat, mat1, Imgproc.COLOR_RGB2GRAY);

            byte[] data1 = new byte[mat1.rows() * mat1.cols() * (int)(mat1.elemSize())];
            mat1.get(0, 0, data1);
            BufferedImage image1 = new BufferedImage(mat1.cols(), mat1.rows(),
BufferedImage.TYPE_BYTE_GRAY);
            image1.getRaster().setDataElements(0, 0, mat1.cols(), mat1.rows(), data1);

            File ouptut = new File("grayscale.jpg");
            ImageIO.write(image1, "jpg", ouptut);

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

```

Output

When you execute the given example, it converts an image name **digital_image_processing.jpg** to its equivalent Grayscale image and writes it on hard disk with name **grayscale.jpg**.

Original Image





Grayscale Image



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