JAVA DIP - BASIC THRESHOLDING

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

Thresholding enables to achieve image segmentation in the easiest way. Image segmentation means dividing the complete image into a set of pixels in such a way that the pixels in each set have some common characteristics. Image segmentation is highly useful in defining objects and their boundaries.

In this chapter we perform some basic thresholding operations on images.

We use **OpenCV** function **threshold**. It can be found under **Imgproc** package. Its syntax is given below:

```
Imgproc.threshold(source, destination, thresh , maxval , type);
```

The parameters are described below:

Sr.No.	Parameters
1	**source** It is source image.
2	**destination** It is destination image.
3	**thresh** It is threshold value.
4	**maxval** It is the maximum value to be used with the THRESH_BINARY and THRESH_BINARY_INV threshold types.
5	**type** The possible types are THRESH_BINARY, THRESH_BINARY_INV, THRESH_TRUNC, and THRESH_TOZERO.

Apart from these thresholding methods, 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

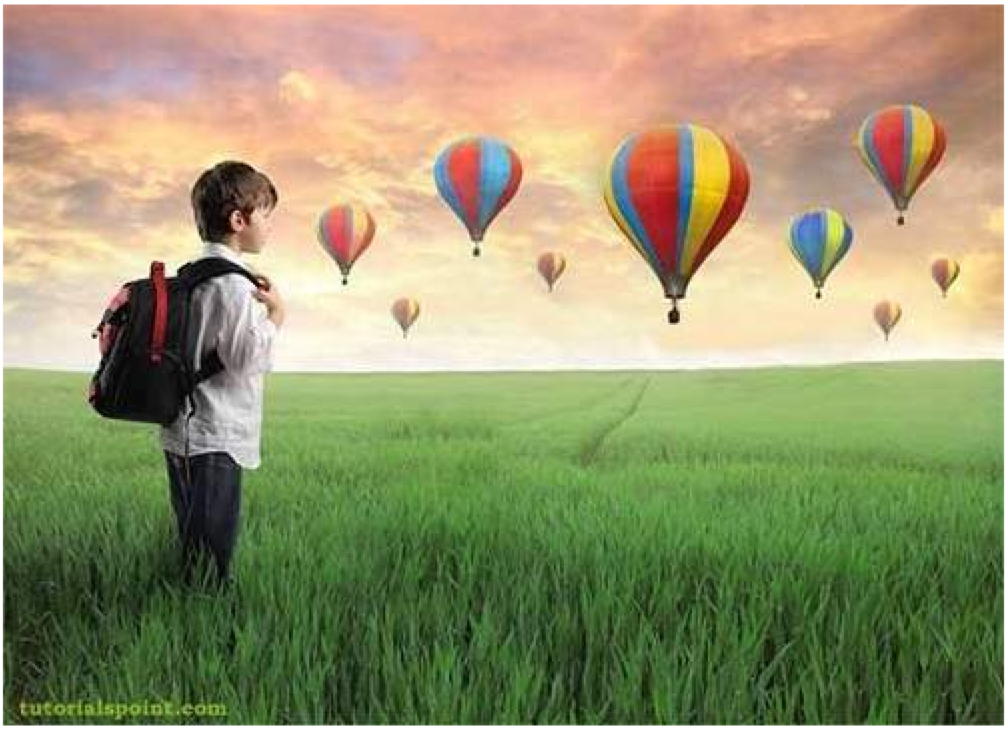
The following example demonstrates the use of Imgproc class to perform thresholding operations to an image:

```
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 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 destination = new Mat(source.rows(),source.cols(),source.type());

         destination = source;
         Imgproc.threshold(source,destination,127,255,Imgproc.THRESH_TOZERO);
         Highgui.imwrite("ThreshZero.jpg", destination);

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

Output

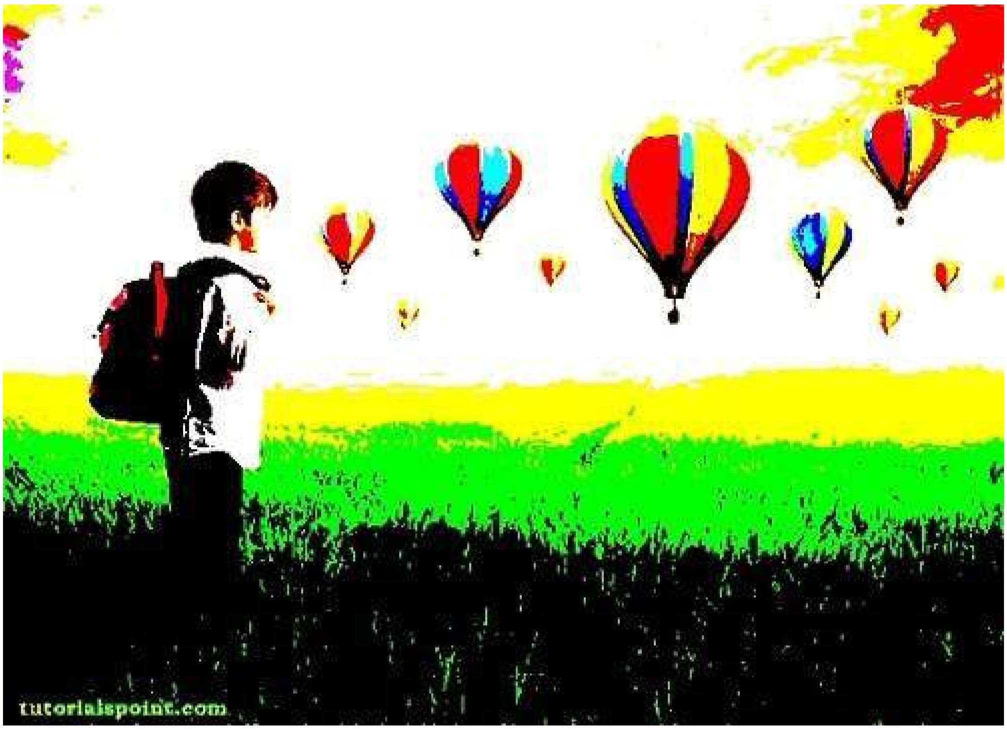
When you execute the given code, the following output is seen:

Original Image

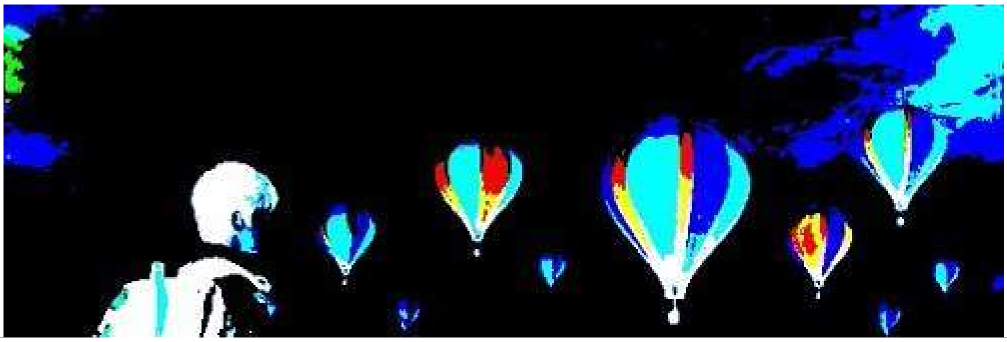


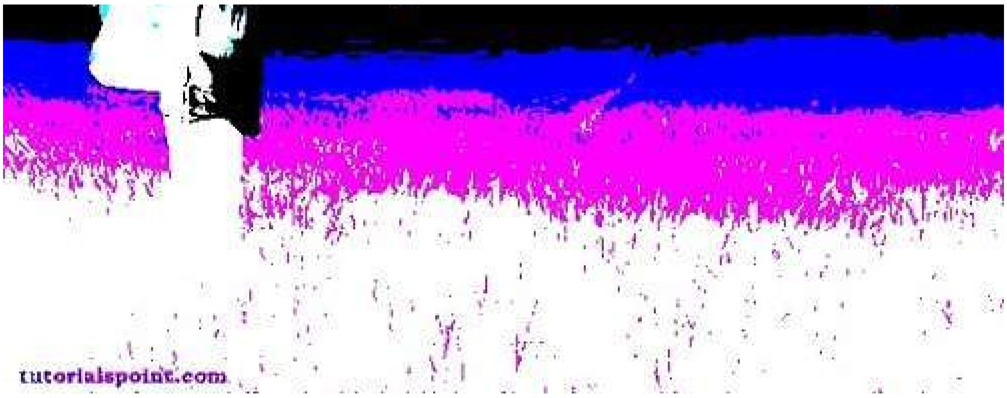
On the above original image, some thresholding operations is performed which is shown in the output below:

Thresh Binary



Thresh Binary Invert





Thresh Zero