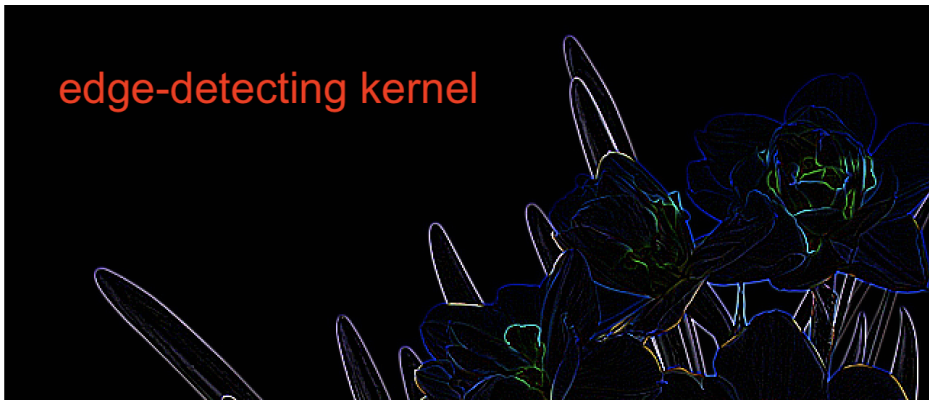
```
import java.awt.*;
import java.awt.image.BufferedImage;
import com.programwithjava.basic.DrawingKit;

public class ConvolveImage {

	public BufferedImage convolve(float[][] kernel, BufferedImage picture){

		int width = picture.getWidth();
		int height = picture.getHeight();
		// create a new BufferedImage called newPicture
		BufferedImage newPicture = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);

		float[][]arrayRed = new float [3][3];
		float[][]arrayGreen = new float [3][3];
		float[][]arrayBlue = new float [3][3];

		// modify the pixels in the picture
		for (int x = 1; x < picture.getWidth()-1; x++) {
			for (int y = 1; y < picture.getHeight()-1; y++) {

				// get the red, green and blue components of pixel at (x, y)
				int colorValue = picture.getRGB(x, y);
				Color pixelColor = new Color(colorValue);
				arrayRed[1][1] = pixelColor.getRed();
				arrayGreen[1][1] = pixelColor.getGreen();
				arrayBlue[1][1] = pixelColor.getBlue();

				colorValue = picture.getRGB(x-1,y-1);
				pixelColor = new Color(colorValue);
				arrayRed[0][0] = pixelColor.getRed();
				arrayGreen[0][0] = pixelColor.getGreen();
				arrayBlue[0][0] = pixelColor.getBlue();

				colorValue = picture.getRGB(x,y-1);
				pixelColor = new Color(colorValue);
				arrayRed[0][1] = pixelColor.getRed();
				arrayGreen[0][1] = pixelColor.getGreen();
				arrayBlue[0][1] = pixelColor.getBlue();

				colorValue = picture.getRGB(x+1,y-1);
				pixelColor = new Color(colorValue);
				arrayRed[0][2] = pixelColor.getRed();
				arrayGreen[0][2] = pixelColor.getGreen();
				arrayBlue[0][2] = pixelColor.getBlue();

				colorValue = picture.getRGB(x+1,y);
				pixelColor = new Color(colorValue);
				arrayRed[1][2] = pixelColor.getRed();
				arrayGreen[1][2] = pixelColor.getGreen();
				arrayBlue[1][2] = pixelColor.getBlue();

				colorValue = picture.getRGB(x+1,y+1);
				pixelColor = new Color(colorValue);
				arrayRed[2][2] = pixelColor.getRed();
				arrayGreen[2][2] = pixelColor.getGreen();
				arrayBlue[2][2] = pixelColor.getBlue();

				colorValue = picture.getRGB(x,y+1);
				pixelColor = new Color(colorValue);
				arrayRed[2][1] = pixelColor.getRed();
				arrayGreen[2][1] = pixelColor.getGreen();
				arrayBlue[2][1] = pixelColor.getBlue();

				colorValue = picture.getRGB(x-1,y+1);
				pixelColor = new Color(colorValue);
				arrayRed[2][0] = pixelColor.getRed();
				arrayGreen[2][0] = pixelColor.getGreen();
				arrayBlue[2][0] = pixelColor.getBlue();

				colorValue = picture.getRGB(x-1,y);
				pixelColor = new Color(colorValue);
				arrayRed[1][0] = pixelColor.getRed();
				arrayGreen[1][0] = pixelColor.getGreen();
				arrayBlue[1][0] = pixelColor.getBlue();
```

```
int newRed = (int) (kernel[1][1] * arrayRed[1][1] + kernel[0][0] * arrayRed[0][0] + kernel[0][1] * arrayRed[0][1]
+ kernel[0][2] * arrayRed[0][2] + kernel[1][2] * arrayRed[1][2] + kernel[2][2] * arrayRed[2][2] + kernel[2][1] *
arrayRed[2][1]+ kernel[2][0] * arrayRed[2][0] + kernel[1][0] * arrayRed[1][0]);

int newGreen = (int) (kernel[1][1] * arrayGreen[1][1] + kernel[0][0] * arrayGreen[0][0] + kernel[0][1] *
arrayGreen[0][1] + kernel[0][2] * arrayGreen[0][2] + kernel[1][2] * arrayGreen[1][2] + kernel[2][2] *
arrayGreen[2][2] + kernel[2][1] * arrayGreen[2][1] + kernel[2][0] * arrayGreen[2][0] + kernel[1][0] *
arrayGreen[1][0]);

int newBlue = (int) (kernel[1][1] * arrayBlue[1][1] + kernel[0][0] * arrayBlue[0][0] + kernel[0][1] *
arrayBlue[0][1] + kernel[0][2] * arrayBlue[0][2] + kernel[1][2] * arrayBlue[1][2] + kernel[2][2] * arrayBlue[2]
[2] + kernel[2][1] * arrayBlue[2][1] + kernel[2][0] * arrayBlue[2][0] + kernel[1][0] * arrayBlue[1][0]);


if(newRed>255){ newRed=255; } else if(newRed<=0){ newRed=1; }
if(newGreen>255){ newGreen=255; } else if(newGreen<=0){ newGreen=1; }
if(newBlue>255){ newBlue=255; } else if(newBlue<=0){ newBlue=1; }

// update the pixel color in picture
Color newPixelColor = new Color(newRed, newGreen, newBlue);
int newRgbvalue = newPixelColor.getRGB();
newPicture.setRGB(x, y, newRgbvalue);

                } // end inner for loop
        } // end outer for loop
                // returns a reference to the new image
return newPicture;
}


public static void main(String[] args) {

                ConvolveImage convolveImage = new ConvolveImage();
                DrawingKit dk = new DrawingKit("Convolve colors", 1680, 700);
                BufferedImage picture = dk.loadPicture("image/daffodils.jpg");

                // sharpening kernel
                float[][] kernel = {{-1.0f, -1.0f, -1.0f},{-1.0f, 9.0f, -1.0f},{-1.0f, -1.0f, -1.0f}};
                // blurring kernel
//              float[][] kernel = {{1.0f/9,1.0f/9, 1.0f/9},{1.0f/9,1.0f/9, 1.0f/9},{1.0f/9,1.0f/9, 1.0f/9}};
                // edge-detecting kernel
//              float[][] kernel = {{-1.0f, -1.0f, -1.0f},{-1.0f, 8.0f, -1.0f},{-1.0f, -1.0f, -1.0f}};
                // embossed kernel
//              float[][] kernel = {{-2.0f, -2.0f, -2.0f},{-1.0f, 1.0f, 1.0f},{2.0f, 2.0f, 2.0f}};

                // draw the original picture
                dk.drawPicture(picture, 0, 50);

                BufferedImage newPicture = convolveImage.convolve(kernel, picture);
                dk.drawPicture(newPicture, 800, 50);
        }
}
```


edge-detecting kernel


embossed kernel


blurring kernel