

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

ОТЧЕТ
По лабораторной работе №5
Библиотека OpenCV

Выполнил:
Студент 4 курса
группы АС-50
Ольховик И.Ю.
Проверил:
Давидюк Ю.И.

Брест 2020

Цель работы:

Изучить и применить на практике доступные функции openCV.

Постановка задачи:

	ФИО	Варианты		
17	Ольховик Илья Юрьевич	2	8	29

Варианты:

2	Фильтрация изображения. Гауссовский фильтр
8	Выделение краев. Фильтр Собела
29	Повышение яркости с помощью свертки.

Задание 1:

Фильтрация изображения. Гауссовский фильтр

Код программы:

```
import org.opencv.core.Core;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.Size;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Введите путь: "); //Ducky.bmp
        String path = input.next();

        try {
            System.loadLibrary( Core.NATIVE_LIBRARY_NAME );

            Mat source = Imgcodecs.imread(path,Imgcodecs.IMREAD_COLOR);
            System.out.println("Введите степень фильтра: "); //15
            int probability = input.nextInt();

            Mat destination = new Mat(source.rows(),source.cols(),source.type());
            Mat noise = new Mat(source.size(), source.type());
            MatOfDouble mean = new MatOfDouble ();
            MatOfDouble dev = new MatOfDouble();

            Core.meanStdDev(source,mean,dev);
```

```

        Core.randn(noise, mean.get(0,0)[0], dev.get(0,0)[0]);
        Core.add(source, noise, source);
        Imgcodecs.imwrite("noise.jpg", source);

        Imgproc.GaussianBlur(source, destination, new
        Size(probability, probability), 0);

        Imgcodecs.imwrite("Gaussian"+ probability + ".jpg", destination);
        System.out.println("Применение фильтра Гауса со степенью " + probability
        + " завершено успешно!");

        } catch (Exception e) {
            System.out.println("Ошибка применения фильтра Гауса:" +
            e.getMessage());
        }
    }
}

```

Запуск программы:

Введите путь:

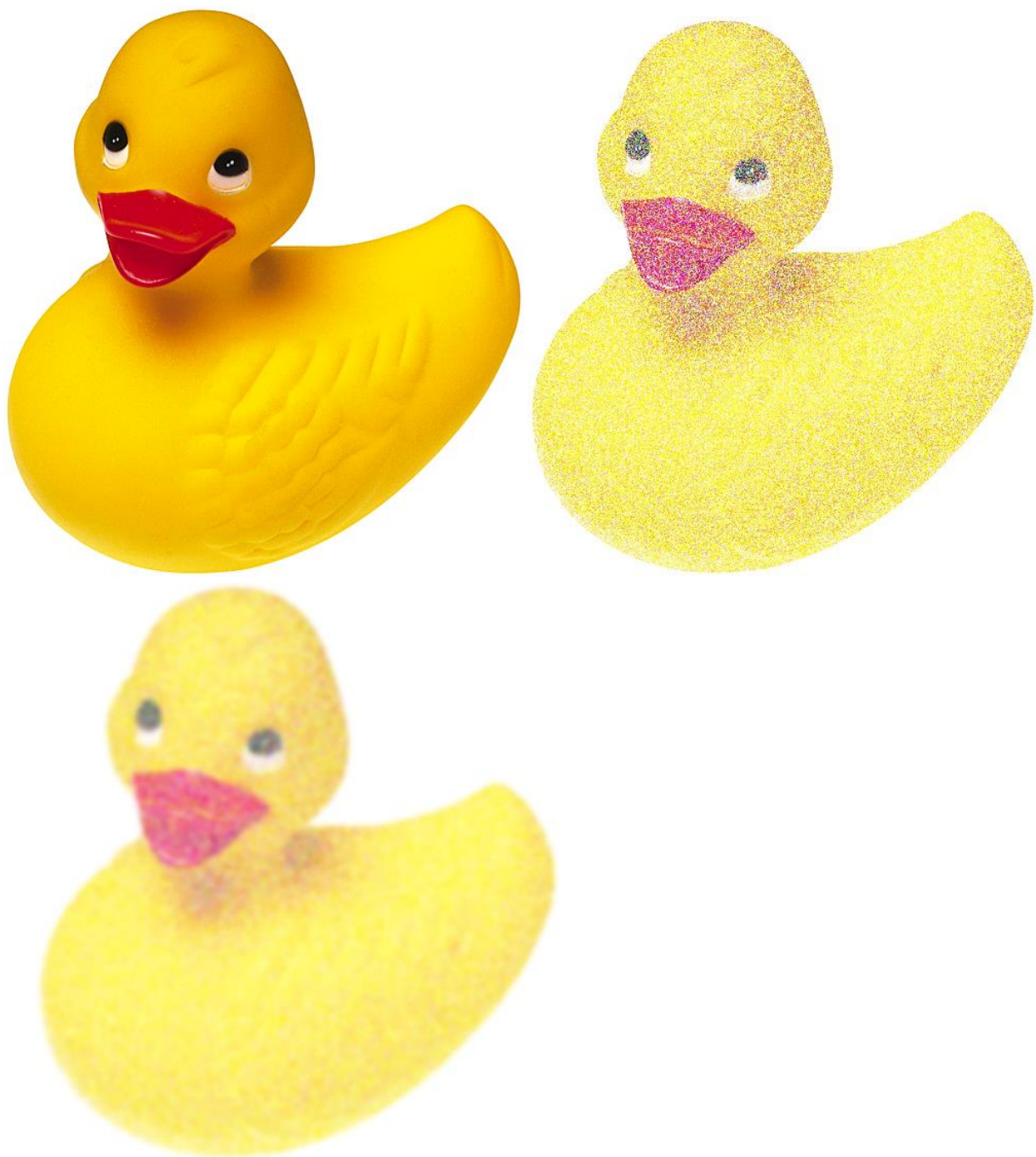
Ducky.bmp

Введите степень фильтра:

15

Применение фильтра Гауса со степенью 15 завершено успешно!

Исходное изображение, зашумленное изображение и итоговое изображение:



Задание 2:

Выделение краев. Фильтр Собела (использовалась маска для фильтрации по оси Y).

Код программы:

```
import org.opencv.core.Core;  
import org.opencv.core.CvType;  
import org.opencv.core.Mat;  
import org.opencv.core.Size;  
import org.opencv.imgcodecs.Imgcodecs;  
import org.opencv.imgproc.Imgproc;
```

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Введите путь: "); //Ducky.bmp
        String path = input.next();

    try{
        int kernelSize = 9;
        Mat source = Imgcodecs.imread(path,Imgcodecs.IMREAD_GRAYSCALE);
        Mat destination = new Mat(source.rows(),source.cols(),source.type());

        Mat kernel = new Mat(kernelSize,kernelSize, CvType.CV_32FC1) { //Вводим
матрицу фильтра Собеля по y
            {
                put(0,0,1);
                put(0,1,2);
                put(0,2,1);

                put(1,0,0);
                put(1,1,0);
                put(1,2,0);

                put(2,0,-1);
                put(2,1,-2);
                put(2,2,-1);
            }
        };

        Imgproc.filter2D(source, destination, -1, kernel);
        Imgcodecs.imwrite("Filtred.jpg", destination);
        System.out.println("Применение фильтра Собеля завершено успешно!");
    } catch (Exception e) {
        System.out.println("Ошибка применения фильтра Собеля:" + e.getMessage());
    }
}
}

```

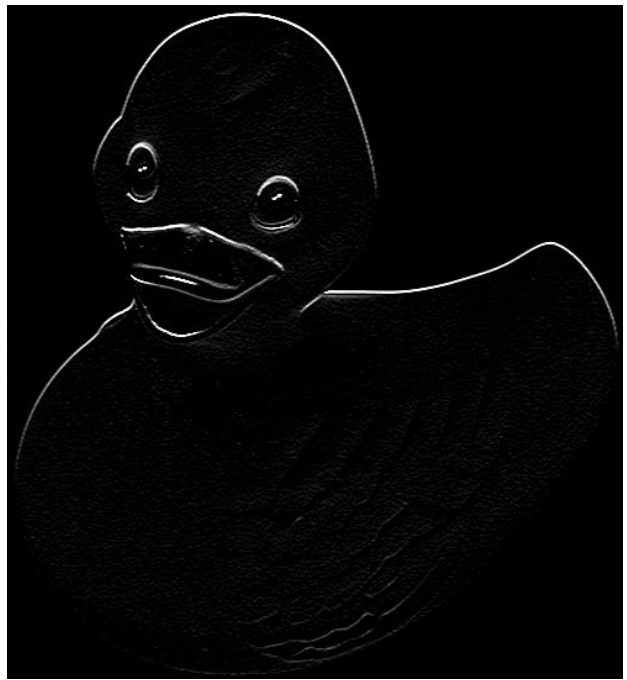
Запуск программы:

Введите путь:

Ducky.bmp

Применение фильтра Собеля завершено успешно!

Исходное изображение и итоговое изображение:



Задание 3:

Повышение яркости с помощью свертки.

Код программы:

```
import org.opencv.core.Core;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.Size;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Введите путь: "); //Ducky.bmp
        String path = input.next();

        try {
            int kernelSize1 = 9;
            Mat source = Imgcodecs.imread(path, Imgcodecs.IMREAD_COLOR);
            Mat destination = new
            Mat(source.rows(), source.cols(), source.type());
            Mat kernel1 = new Mat(kernelSize1, kernelSize1, CvType.CV_32FC1) {
                //Вводим матрицу для повышения яркости
                {
                    put(0, 0, -0.1);
                    put(0, 1, 0.2);
                    put(0, 2, -0.1);

                    put(1, 0, 0.2);
```

```

        put(1,1,3);
        put(1,2,0.2);

        put(2,0,-0.1);
        put(2,1,0.2);
        put(2,2,-0.1);
    }
};
Imgproc.filter2D(source, destination, -1, kernell1);
Imgcodecs.imwrite("Brightness.jpg", destination);
System.out.println("Повышение яркости завершено успешно!");
} catch (Exception e) {
    System.out.println("Ошибка повышения яркости:" + e.getMessage());
}
}
}

```

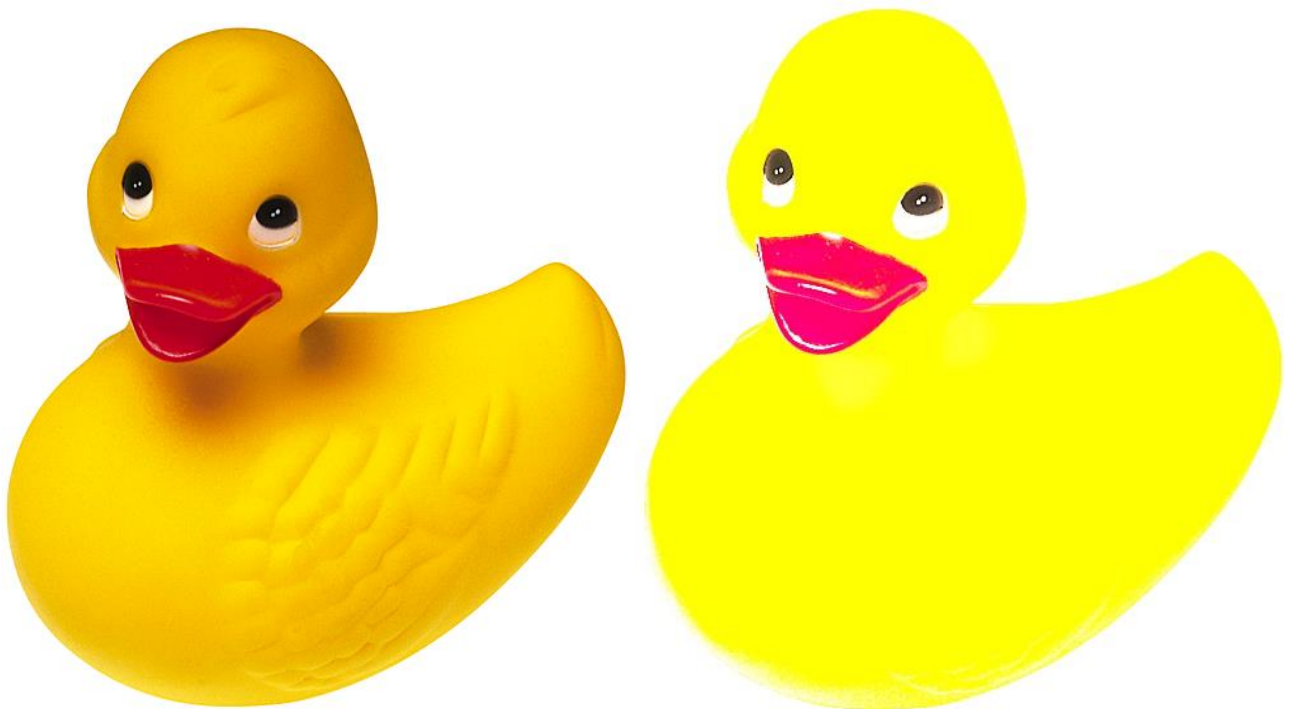
Запуск программы:

Введите путь:

Ducky.bmp

Повышение яркости завершено успешно!

Исходное изображение и итоговое изображение:



Вывод: Изучил и применил на практике доступные функции openCV.