**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра ВТ**

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**ПО ДИСЦИПЛИНЕ**

**«Обработка изображений и компьютерное зрение (OpenCV)»**

**ТЕМА: «СГЛАЖИВАНИЕ ИЗОБРАЖЕНИЙ, АФИННЫЕ ПРЕОБРАЗОВАНИЯ, МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ»**

**Бригада №1**

| Студенты гр. 8307: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Лукьянов А.В. |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Крылов А. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Якушев Г.С. |
| Преподаватель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Костичев С.В. |
|  |  |  |

Санкт-Петербург

2023

**Задание на лабораторную работу:**

1. Фильтрация (функция filter2D**)**

Разработать программу, обеспечивающую загрузку изображения image0.jpg и применение линейного фильтра с вещественным ядром 3х3, заданным константой

kernel = {-0.1, 0.2, -0.1,

0.2, 3.0, 0.2,

-0.1, 0.2, -0.1};

Выполнить отображение исходного и результирующего изображений

1. Сглаживание изображений (функции blur, boxFilter, GaussianBlur и medianBlur)

Разработать программу, обеспечивающую загрузку изображения image0.jpg и применение размытия изображения функциями blur, boxFilter, GaussianBlur, с квадратным ядром со стороной в 5 пикселей и функцией medianBlur с линейным размером апертуры 15.

Выполнить отображения исходного и результирующих изображений.

1. Аффинные преобразования
   1. Разработать программу, обеспечивающую загрузку изображения lena.jpg и применение поворота изображения на 45 градусов функцией warpAffine.

Выполнить отображение исходного и результирующего изображений

* 1. Уменьшить исходное изображение lena.jpg в 2 раза по горизонтали (функция resize). Выполнить отображение исходного и результирующего изображений.

1. Морфологические операции
   1. Разработать программу, обеспечивающую загрузку черно-белого изображения image1.jpg и применение

- эрозии к исходному изображению (функция erode)

- дилатации к исходному изображению (функции dilate)

- эрозии к исходному изображению, а затем дилатации к результату эрозии (функции erode и dilate).

В качестве структурного элемента использовать прямоугольный структурный элемент размером 3х3.

Выполнить отображения исходного и результирующих изображений.

* 1. Разработать программу, обеспечивающую загрузку черно-белого изображения image1.jpg и применения к нему морфологических операций размыкания, замыкания, морфологический градиент, «цилиндр», «черная шляпа».

В качестве структурного элемента использовать прямоугольный структурный элемент размером 3х3.

Выполнить отображения исходного и результирующих изображений после применения морфологических операций.

**Программное и аппаратное окружение при выполнении работы**

Поскольку члены нашей группы пользуются различными ОС, мы не смогли сделать выбор в пользу какой-то конкретной ОС, потому мы использовали машины под управлением ОС Ubuntu и ОС Windows 11, а также среду разработки CLion для языков C/C++ с поддержкой кроссплатформенной системой сборки CMake.

Аппаратное обеспечение варьируется довольно сильно - в нашем распоряжении оказались процессоры Intel Core i5-6600(4C|4T, 32 gb ram), i5-12600k(10c|16T, 32gb ram + 40 gb swap), а также AMD A10-9620P(4C|4T, 6gb ram + 20gb swap).

При работе мы использовали CLion версии 2023.1.1., CMake 3.24, libopencv 4.7.0-dev.

**Пример запуска программы**

**Задание 1**

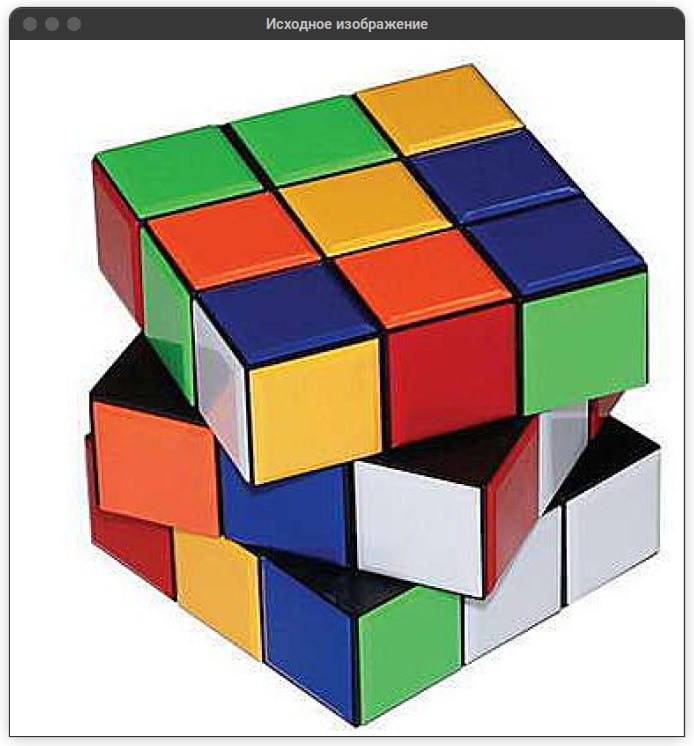
****

Рисунок 1 - Исходное изображение

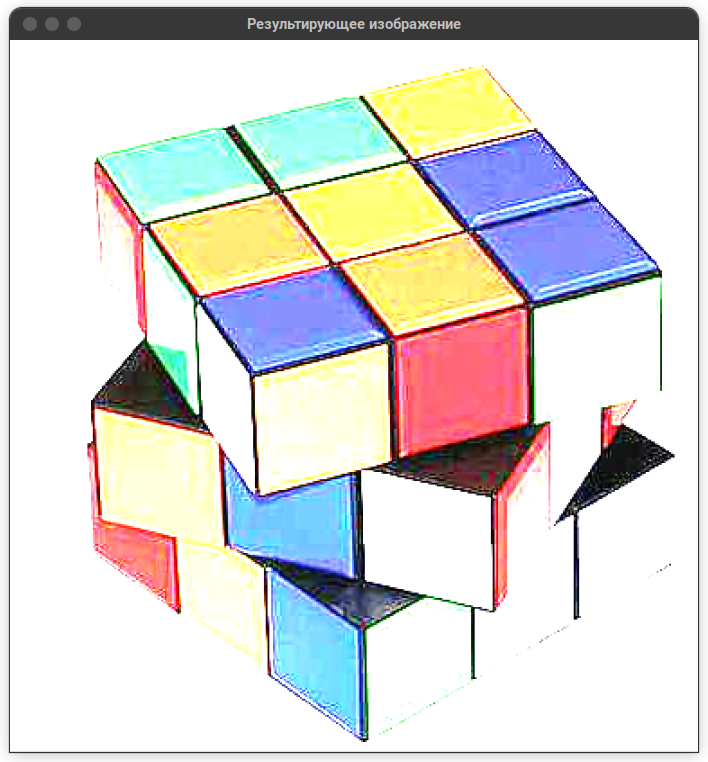


Рисунок 2 - Отфильтрованное изображение

**Задание 2**

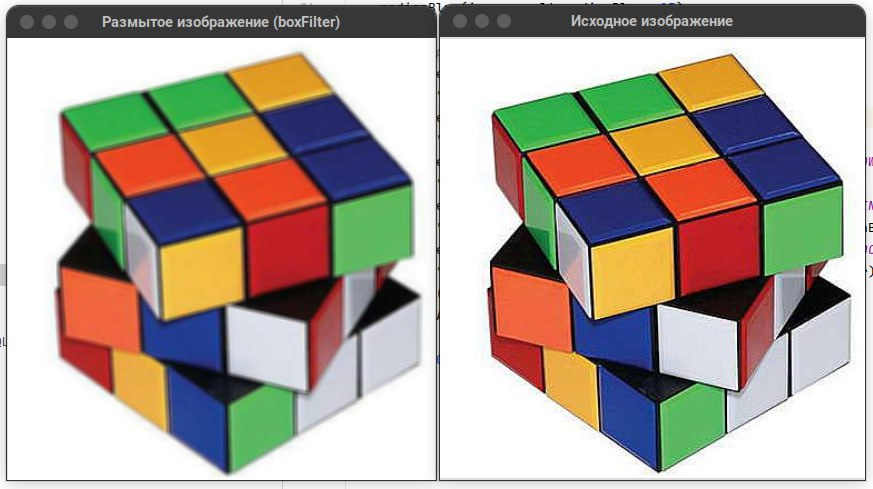


Рисунок 3 - Размытое изображение(boxFilter) и исходное изображение

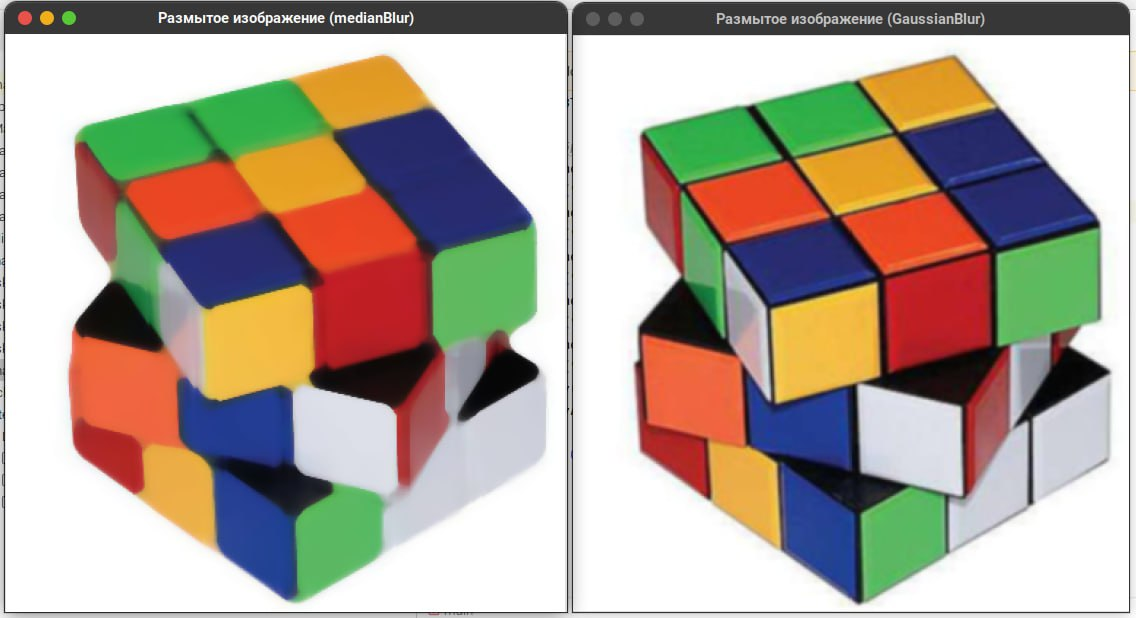


Рисунок 4 - Размытое изображение(medianBlur) и размытое изображение(GaussianBlur)

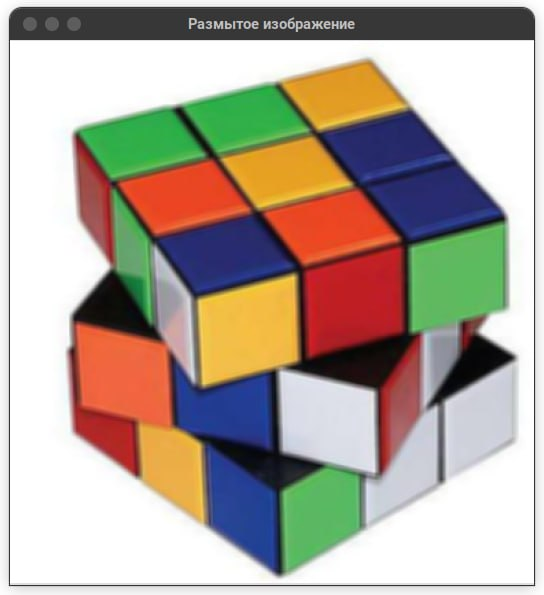


Рисунок 5 - Размытое изображение

**Задание 3**

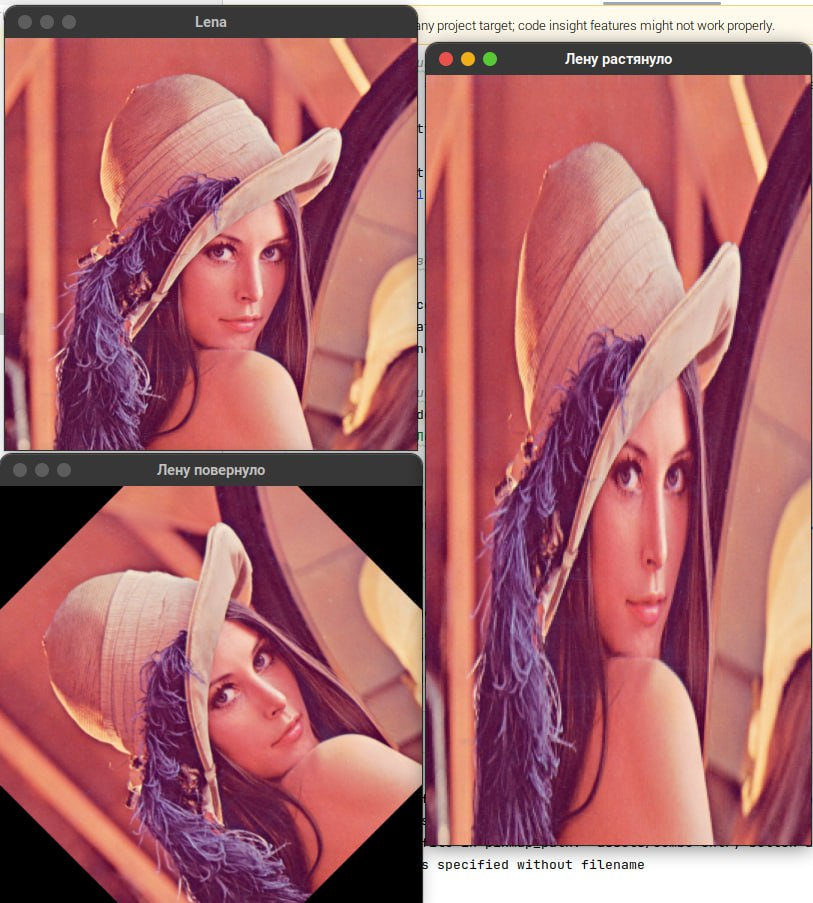


Рисунок 6 - Операции над Леной

**Задание 4**

**Часть 1**

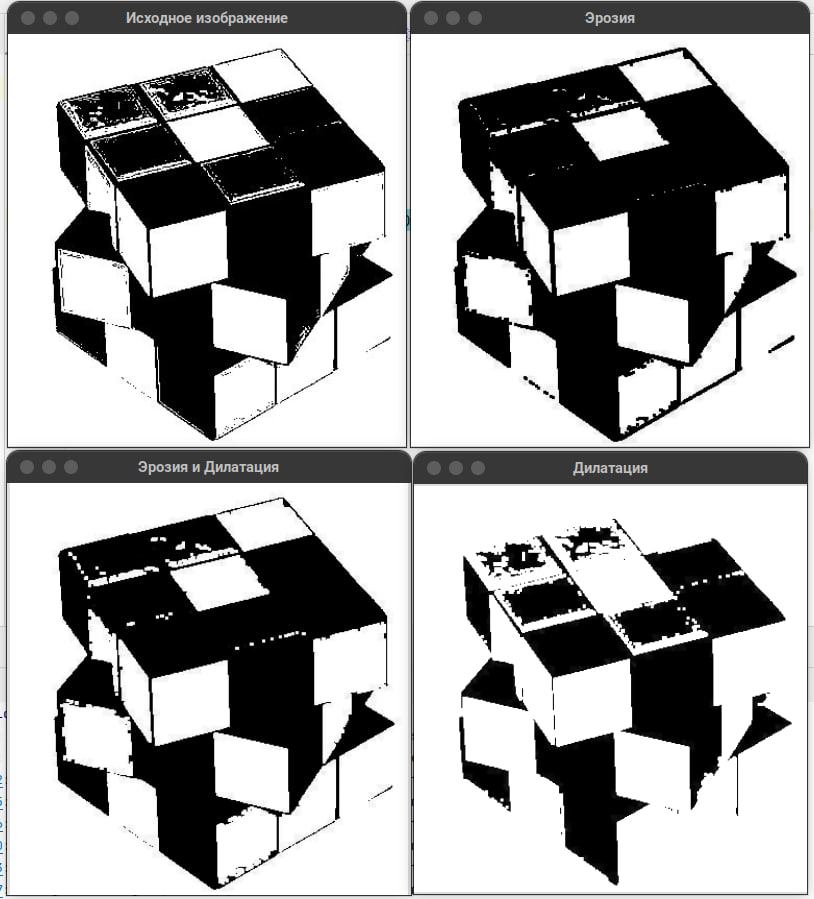
****

Рисунок 7 - Морфологические операции часть 1

**Часть 2**

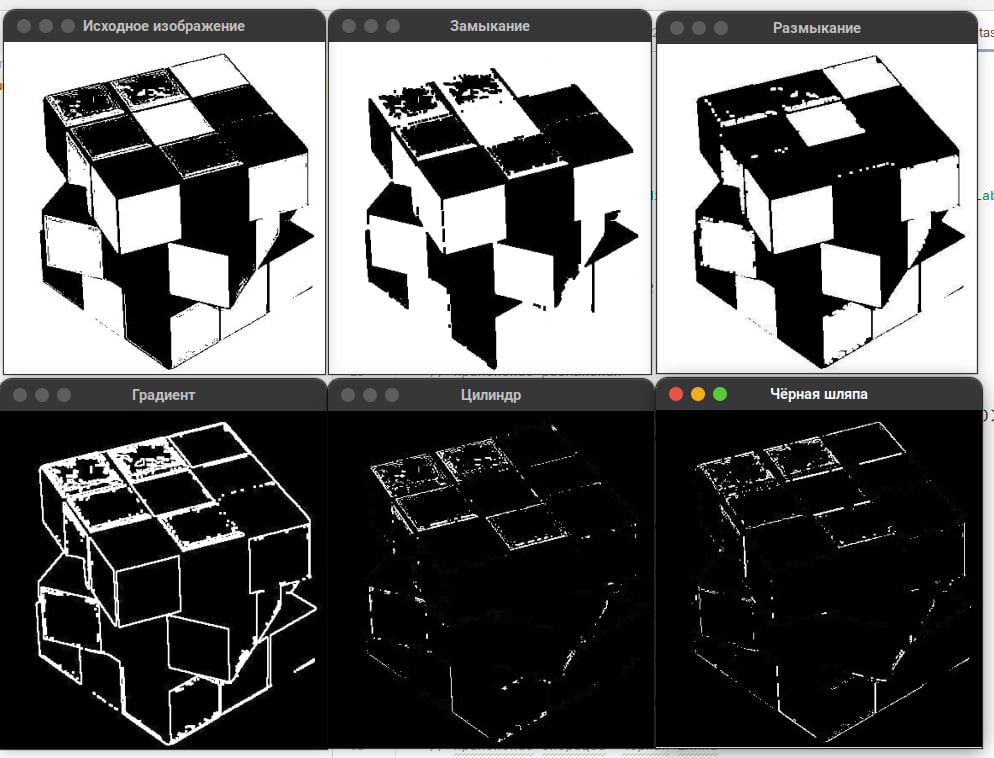


Рисунок 8 - Морфологические операции часть 2

**Заключение**

В ходе лабораторной работы было изучено фильтрация, сглаживание изображений, аффинные преобразования, морфологические операции над изображениями используя возможности библиотеки OpenCV.

**ПРИЛОЖЕНИЕ А.**

**Текст программы задания 1:**

#include <opencv2/opencv.hpp>

#include <iostream>

int main()

{

*// Загрузка изображения*

cv::Mat image = cv::imread("/media/artyom/Новая папка/ETU/10\_sem/OpenCV/Lab\_3/image0.jpg");

*// Проверка на успешную загрузку изображения*

if (image.empty())

{

std::cout << "Не удалось загрузить изображение!" << std::endl;

return -1;

}

*// Линейный фильтр с ядром kernel*

cv::Mat filtered\_image;

float kernel\_data[9] = {-0.1, 0.2, -0.1, 0.2, 3.0, 0.2, -0.1, 0.2, -0.1};

cv::Mat kernel(3, 3, CV\_32F, kernel\_data);

cv::filter2D(image, filtered\_image, -1, kernel);

*// Отображение исходного и результирующего изображений*

cv::namedWindow("Исходное изображение", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Результирующее изображение", cv::*WINDOW\_NORMAL*);

cv::imshow("Исходное изображение", image);

cv::imshow("Результирующее изображение", filtered\_image);

cv::waitKey(0);

return 0;

}

**ПРИЛОЖЕНИЕ Б.**

**Текст программы задания 2:**

#include <opencv2/opencv.hpp>

#include <iostream>

int main()

{

*// Загрузка изображения*

cv::Mat img = cv::imread("/media/artyom/Новая папка/ETU/10\_sem/OpenCV/Lab\_3/image0.jpg");

if (img.empty())

{

std::cout << "Не удалось загрузить изображение!" << std::endl;

return -1;

}

*// Применение размытия функцией blur*

cv::Mat result\_blur;

cv::blur(img, result\_blur, cv::Size(5, 5));

*// Применение размытия функцией boxFilter*

cv::Mat result\_boxFilter;

cv::boxFilter(img, result\_boxFilter, -1, cv::Size(5, 5));

*// Применение размытия функцией GaussianBlur*

cv::Mat result\_GaussianBlur;

cv::GaussianBlur(img, result\_GaussianBlur, cv::Size(5, 5), 0);

*// Применение медианного размытия*

cv::Mat result\_medianBlur;

cv::medianBlur(img, result\_medianBlur, 15);

*// Отображение исходного и результирующих изображений*

cv::namedWindow("Исходное изображение", cv::*WINDOW\_NORMAL*);

cv::imshow("Исходное изображение", img);

cv::namedWindow("Размытое изображение", cv::*WINDOW\_NORMAL*);https:*//habr.com/ru/articles/326986/*

cv::imshow("Размытое изображение", result\_blur);

cv::namedWindow("Размытое изображение (boxFilter)", cv::*WINDOW\_NORMAL*);

cv::imshow("Размытое изображение (boxFilter)", result\_boxFilter);

cv::namedWindow("Размытое изображение (GaussianBlur)", cv::*WINDOW\_NORMAL*);

cv::imshow("Размытое изображение (GaussianBlur)", result\_GaussianBlur);

cv::namedWindow("Размытое изображение (medianBlur)", cv::*WINDOW\_NORMAL*);

cv::imshow("Размытое изображение (medianBlur)", result\_medianBlur);

cv::waitKey(0);

cv::destroyAllWindows();

return 0;

}

**ПРИЛОЖЕНИЕ В.**

**Текст программы задания 3**

#include <opencv2/opencv.hpp>

#include <iostream>

int main()

{

*// загрузка изображения*

cv::Mat lena = cv::imread("/media/artyom/Новая папка/ETU/10\_sem/OpenCV/Lab\_3/lena.jpg");

if (lena.empty())

{

std::cout << " Помогите! Вы потеряли лену!" << std::endl;

return -1;

}

*// поворот изображения на 45 градусов*

cv::Mat rot;

cv::Point2f center(lena.cols/2, lena.rows/2);

cv::Mat rotMat = cv::getRotationMatrix2D(center, 45, 1.0);

cv::warpAffine(lena, rot, rotMat, lena.size());

*// отображение исходного и результирующего изображений*

cv::namedWindow("Лену повернуло", cv::*WINDOW\_NORMAL*);

cv::imshow("Лену повернуло", rot);

*// уменьшение изображения в 2 раза по горизонтали*

cv::Mat resized;

cv::resize(lena, resized, cv::Size(lena.cols/2, lena.rows), 0.5, 1);

*// отображение исходного и результирующего изображений*

cv::namedWindow("Lena", cv::*WINDOW\_NORMAL*);

cv::imshow("Lena", lena);

cv::namedWindow("Лену растянуло", cv::*WINDOW\_NORMAL*);

cv::imshow("Лену растянуло", resized);

cv::waitKey(0);

return 0;

}

**ПРИЛОЖЕНИЕ Г.**

**Текст программы задания 4, пункт 1.**

#include <opencv2/opencv.hpp>

int main()

{

*// загрузка изображения*

cv::Mat image = imread("/media/artyom/Новая папка/ETU/10\_sem/OpenCV/Lab\_3/image1.jpg", cv::*IMREAD\_GRAYSCALE*);

*// проверка на корректность пути и существования файла*

if (image.empty())

{

std::cout << "Не удалось загрузить изображение!" << std::endl;

return -1;

}

*// применение эрозии*

cv::Mat eroded;

cv::Mat element = getStructuringElement(cv::*MORPH\_RECT*, cv::Size(3, 3));

cv::erode(image, eroded, element);

*// применение дилатации*

cv::Mat dilated;

cv::dilate(image, dilated, element);

*// применение эрозии и дилатации*

cv::Mat eroded\_dilated;

cv::erode(image, eroded\_dilated, element);

cv::dilate(eroded\_dilated, eroded\_dilated, element);

*// отображение исходного и результирующих изображений*

cv::namedWindow("Исходное изображение", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Эрозия", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Дилатация", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Эрозия и Дилатация", cv::*WINDOW\_NORMAL*);

cv::imshow("Исходное изображение", image);

cv::imshow("Эрозия", eroded);

cv::imshow("Дилатация", dilated);

cv::imshow("Эрозия и Дилатация", eroded\_dilated);

cv::waitKey(0);

return 0;

}

**ПРИЛОЖЕНИЕ Д.**

**Текст программы задания 4, пункт 2.**

#include <opencv2/opencv.hpp>

#include <iostream>

int main()

{

*// загрузка изображения*

cv::Mat image = imread("/media/artyom/Новая папка/ETU/10\_sem/OpenCV/Lab\_3/image1.jpg", cv::*IMREAD\_GRAYSCALE*);

if (image.empty())

{

std::cout << "Не удалось загрузить изображение!" << std::endl;

return -1;

}

*// применение размыкания*

cv::Mat opened;

cv::Mat element = getStructuringElement(cv::*MORPH\_RECT*, cv::Size(3, 3));

cv::morphologyEx(image, opened, cv::*MORPH\_OPEN*, element);

*// применение замыкания*

cv::Mat closed;

cv::morphologyEx(image, closed, cv::*MORPH\_CLOSE*, element);

*// применение морфологического градиента*

cv::Mat gradient;

cv::morphologyEx(image, gradient, cv::*MORPH\_GRADIENT*, element);

*// применение операции "цилиндр"*

cv::Mat top\_hat;

cv::morphologyEx(image, top\_hat, cv::*MORPH\_TOPHAT*, element);

*// применение операции "черная шляпа"*

cv::Mat black\_hat;

cv::morphologyEx(image, black\_hat, cv::*MORPH\_BLACKHAT*, element);

*// отображение исходного и результирующих изображений*

cv::namedWindow("Исходное изображение", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Размыкание", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Замыкание", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Градиент", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Цилиндр", cv::*WINDOW\_NORMAL*);

cv::namedWindow("Чёрная шляпа", cv::*WINDOW\_NORMAL*);

cv::imshow("Исходное изображение", image);

cv::imshow("Размыкание", opened);

cv::imshow("Замыкание", closed);

cv::imshow("Градиент", gradient);

cv::imshow("Цилиндр", top\_hat);

cv::imshow("Чёрная шляпа", black\_hat);

cv::waitKey(0);

return 0;

}