Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра информационных технологий и систем

**Программа обработки изображений**

Лабораторная работа по дисциплине:

«Распознавание образов и обработка изображений»

Выполнил студент группы 8091:

\_\_\_\_\_\_\_\_\_\_ / Кузин И.А.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Принял преподаватель:

\_\_\_\_\_\_\_\_\_\_ / Назаров A. Г.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**Великий Новгород**

**2022**

1. **Цель работы**

Изучение основ обработки изображений и базовых алгоритмов фильтрации изображений.

Разработать и реализовать программу обработки изображений.

Реализовать плагин медианной фильтрации изображения.

1. **Описание среды окружения**

Для решения поставленной задачи использовалось следующее окружение:

* ОС: Windows 10
* Среда программирования: Qt версии 6.2.2
* Библиотека OpenCV 4.5.5
* Компилятор: Qt 6.2.3 MinGW 64-bit

1. **Описание установки OpenCV**

Для выполнения данной лабораторной работы использовалась библиотека OpenCV. Последовательность установки библиотеки OpenCV:

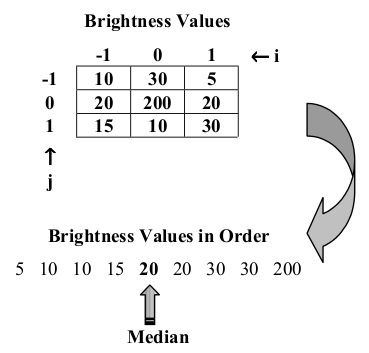
1. Загрузить OpenCV последней версии с официального сайта.
2. Выполнить установку Cmake на свой компьютер.
3. Запустить Сmake и собрать конфигурацию скаченной библиотеки OpenCV для используемого компилятора MinGW.
4. Находясь в директории с собранной конфигурации, через командную строку запустить команду *mingw32-make* и после успешного ее выполнения, запустить установку, с помощью команды *mingw32-install.*
5. Создать файл проекта заголовка (opencv\_library\_path.pri), где прописать пути ко всем необходимым dll файлам.
6. Подключить заголовок в .pri файл *include(../ImageEditor/opencv\_library\_path.pri).*
7. **Использование медианного фильтра**

Необходимость применять фильтрацию вызвана шумами, присутствующими в кадре. Эти шумы имеют разную природу: одни вносит сама камера, другие вносят алгоритмы преобразования, третьи вносит окружающая среда.

Медианный фильтр, на мой взгляд, является наиболее значимым для подавления помех, с которыми мне пришлось столкнуться при разработке этого проекта. Он, как раз, подходит для устранения различного рода мелких рассредоточенных вкраплений, однако он не является идеальным инструментом т.к. пропускает чуть более крупные области сосредоточенных помех.

**Описание работы алгоритма:**

Медианный фильтр представляет собой скользящее окно, в нашем случае, размерностью 3x3 пикселя. На вход он принимает 9 значений (пикселей), а на выход выдаёт одно. Работает медианный фильтр так: сортирует входные данные (пиксели) в порядке возрастания и выдаёт серединный результат (медиану).



1. **Пример работы реализованного алгоритма:**



Рисунок 1 Изображение до применения Медианного фильтра



Рисунок 2 Изображение после применения Медианного фильтра

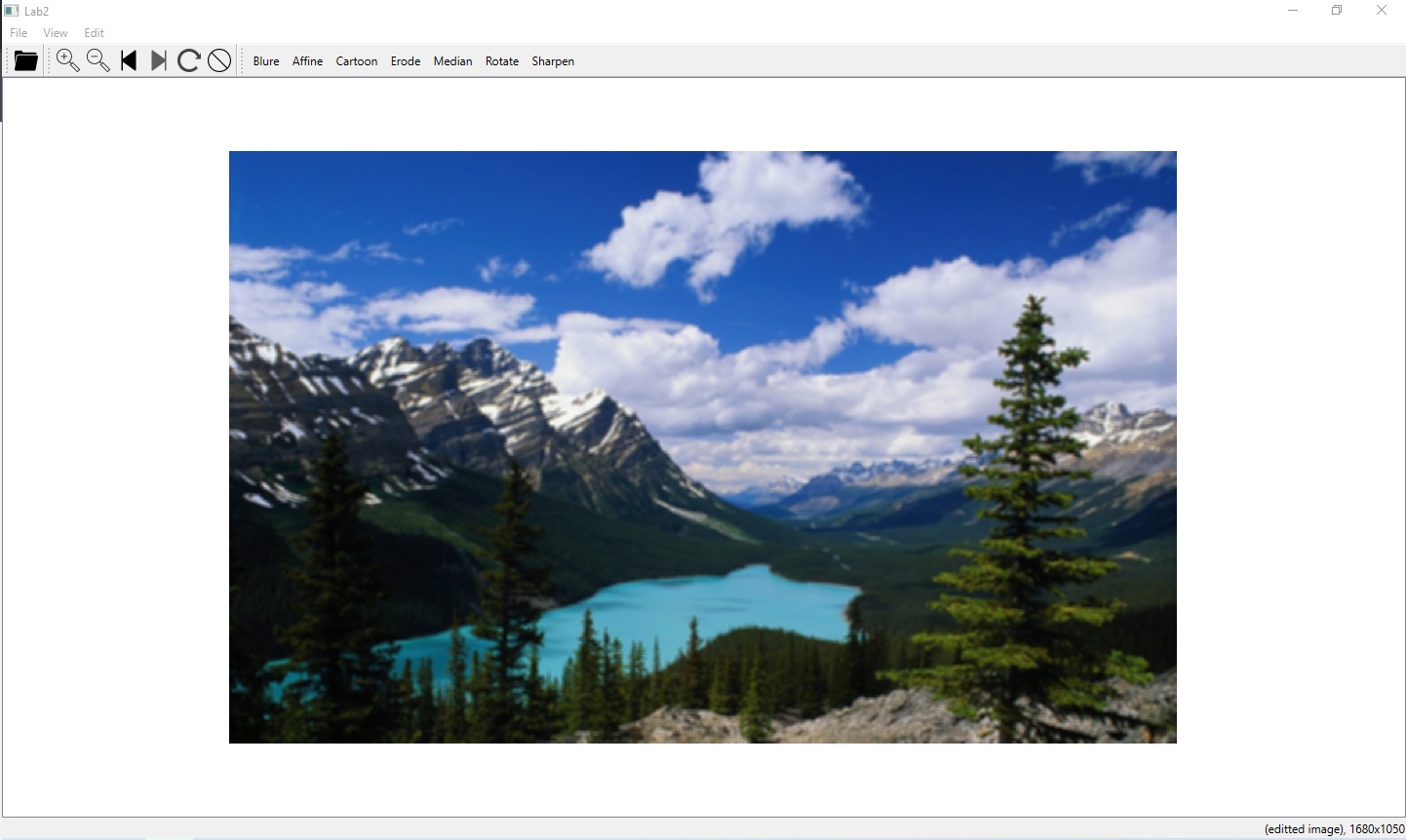


Рисунок 3 Изображение после применения функции blure

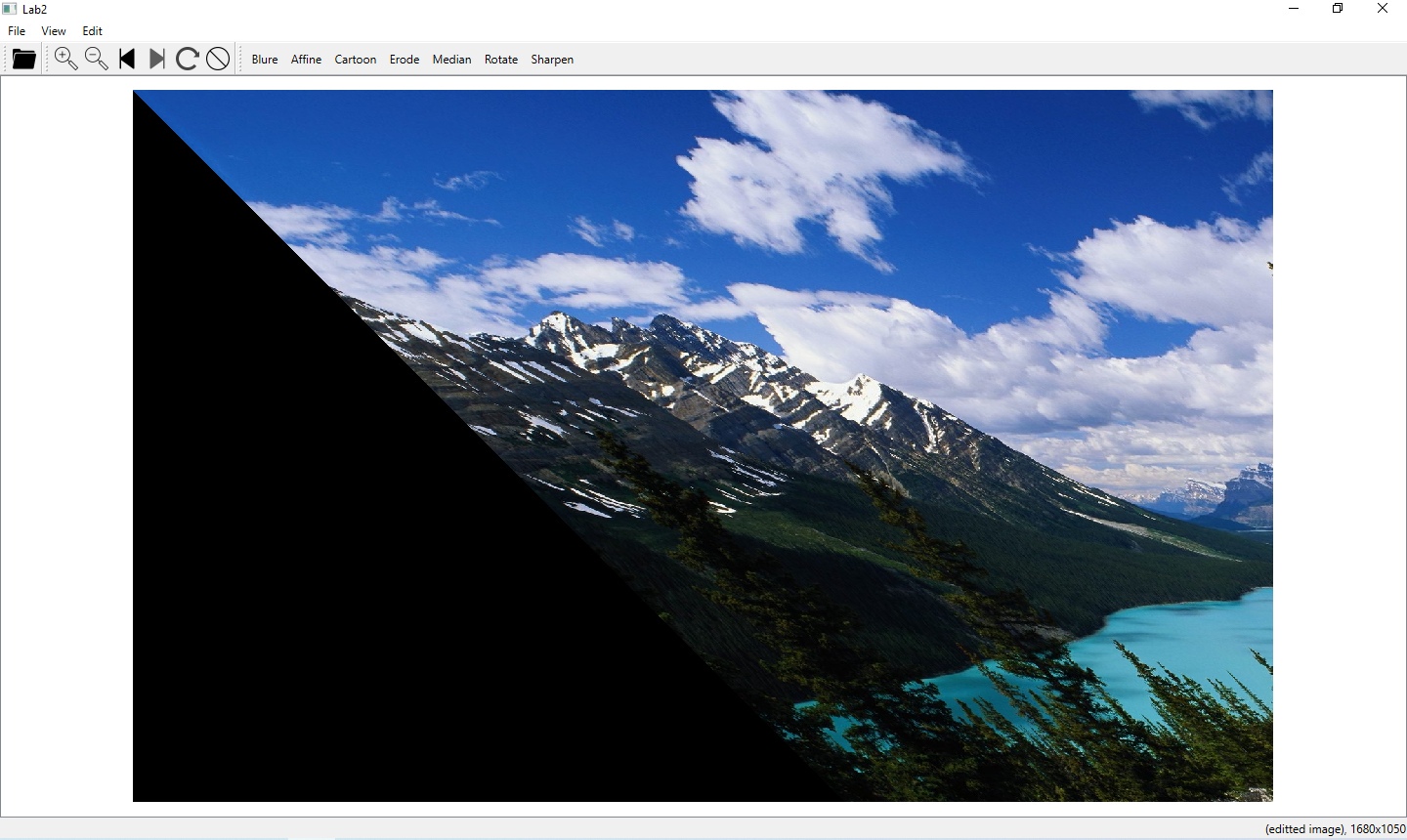


Рисунок 4 Изображение после применения функции affine

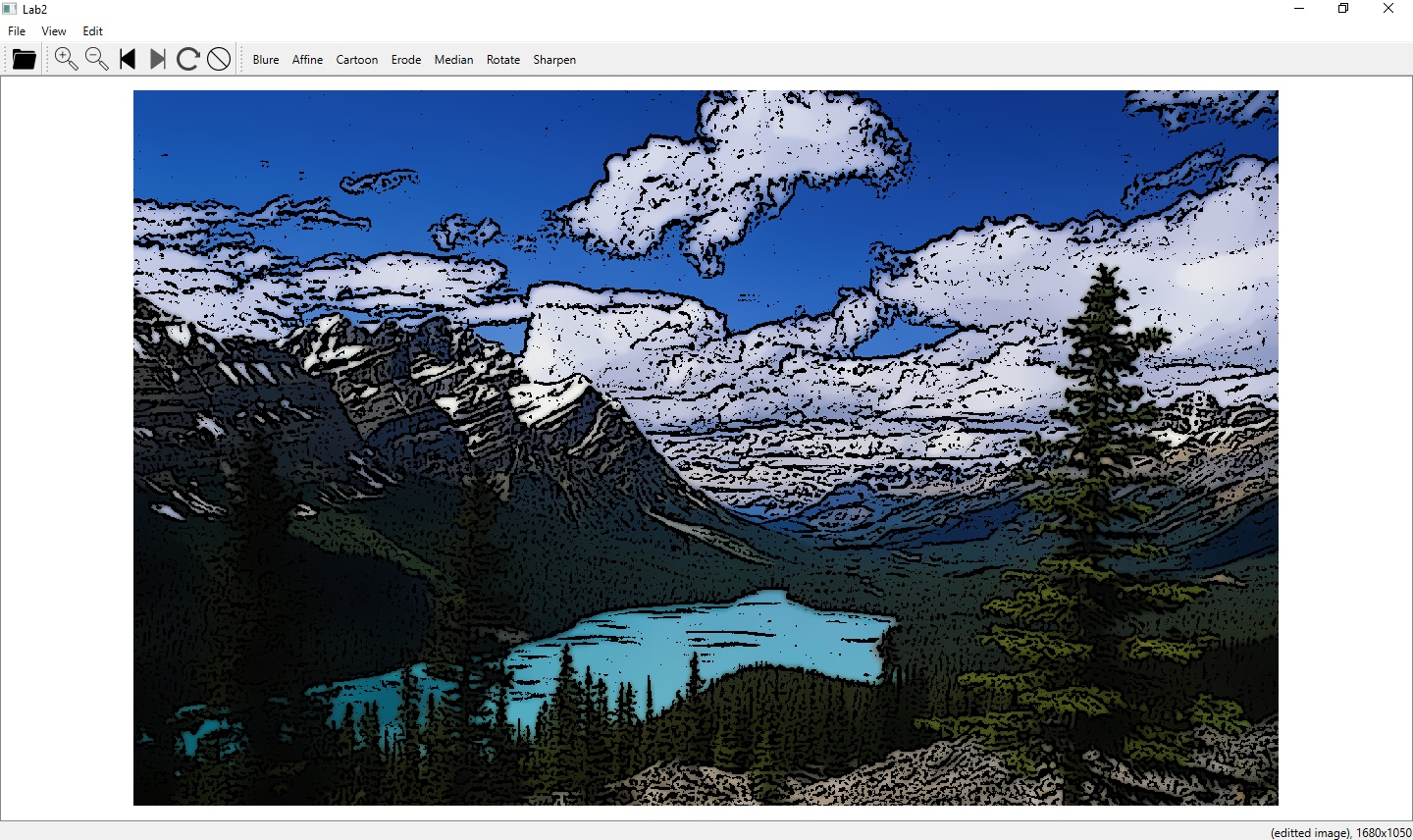


Рисунок 5 Изображение после применения функции cartoon

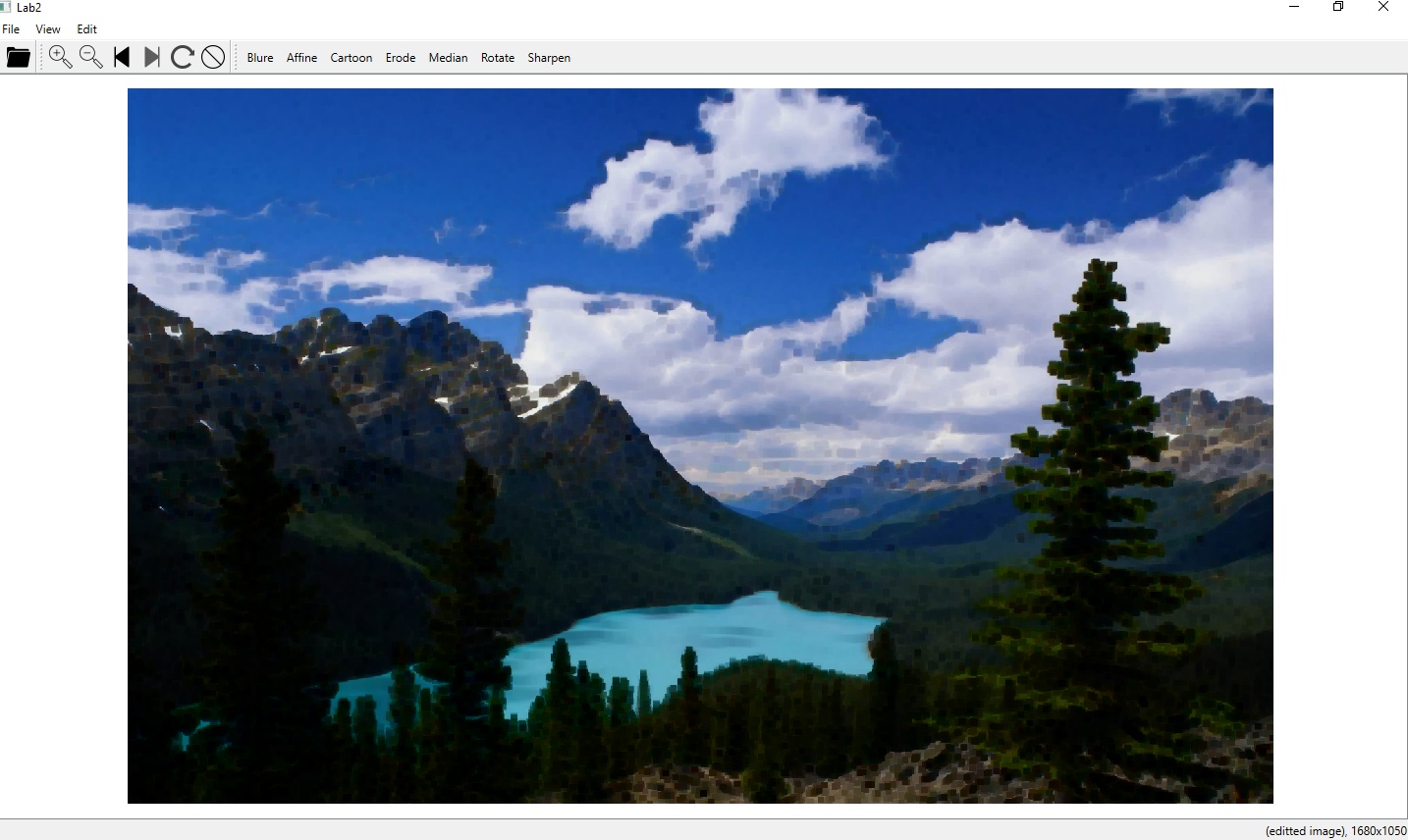


Рисунок 6 Изображение после применения функции errode

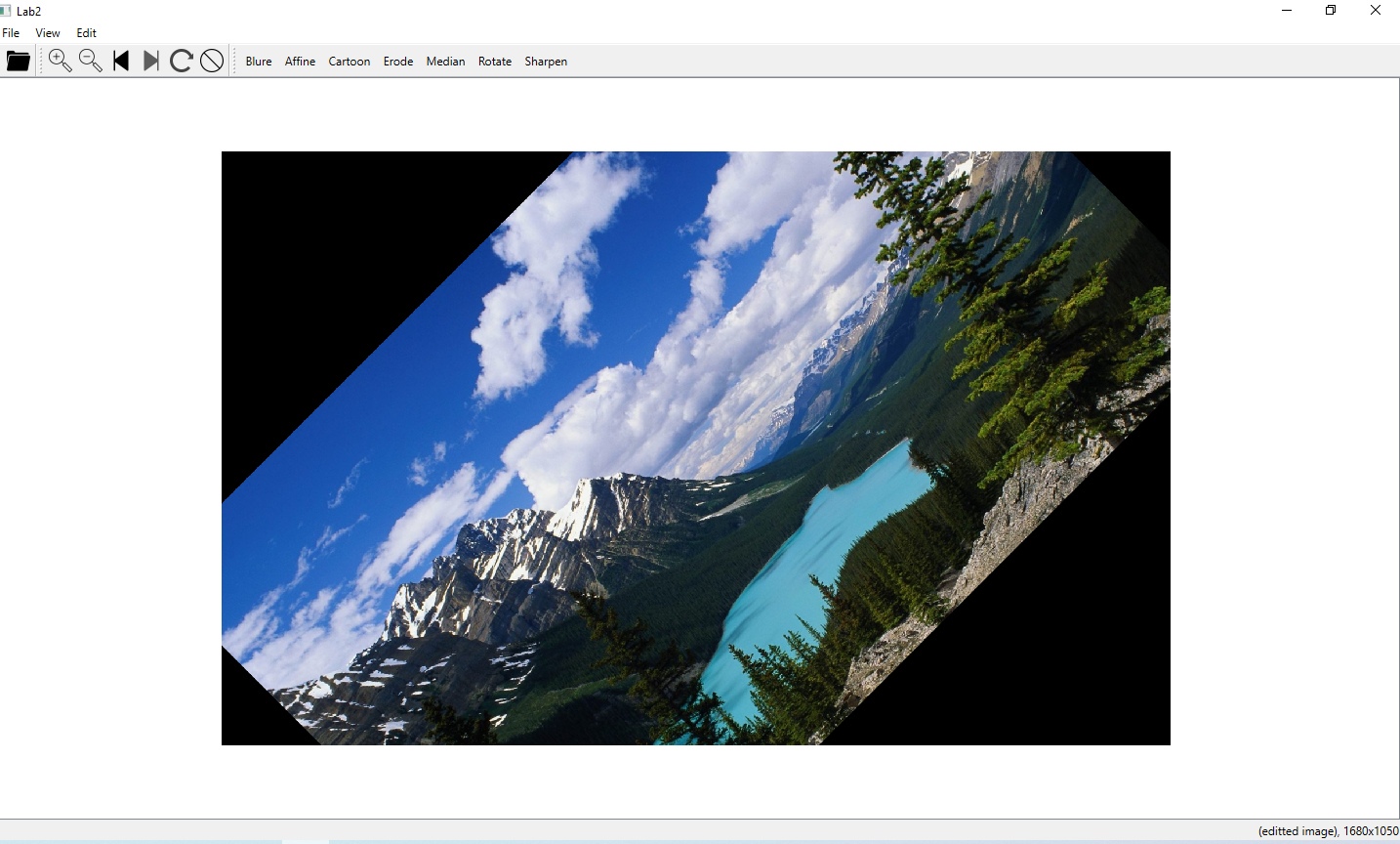


Рисунок 7 Изображение после применения функции rotate

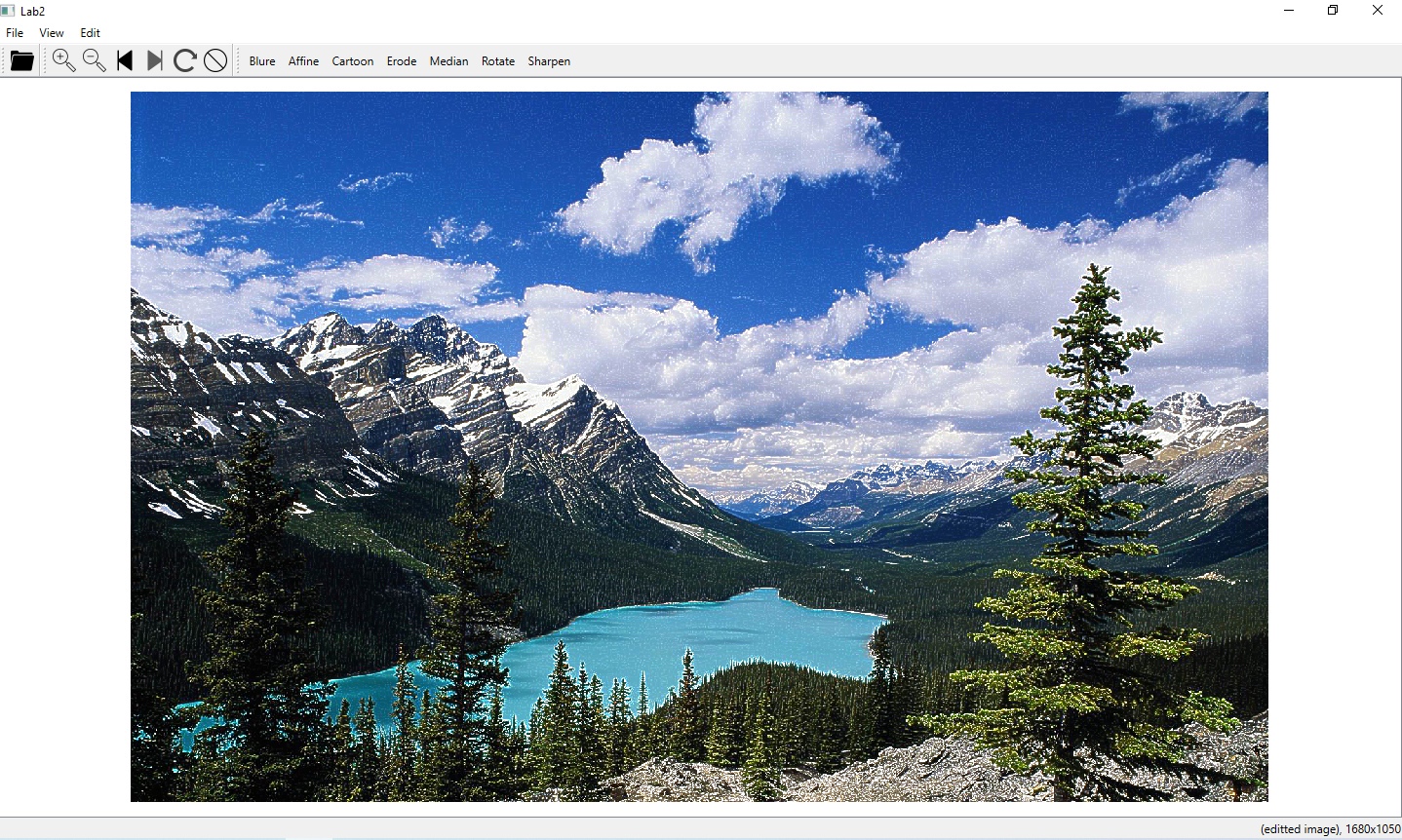


Рисунок 8 Изображение после применения функции sharpen

1. **Ответы на вопросы**
2. Если нам необходимо узнать, поддерживает ли функция OpenCV работу на месте, нужно зайти на официальный сайт OpenCV, найти нужную функцию и в описании к функции должно быть написано, есть ли работа не месте. Если такой информации в описании нету, можно считать, что функция не поддерживает работу на месте.
3. Для добавления горячей клавиши для действия плагина, необходимо проверить условием, действие какого плагина было инициализировано, чтобы потом задать горячую клавишу для этого действия.

Пример реализации:

*if(plugin\_ptr->name()=="Erode")*

*action->setShortcut(Qt::Key\_E);*

*if(plugin\_ptr->name()=="Median")*

*action->setShortcut(Qt::Key\_M);*

1. Для создания возможности отмены примененных модификаций у изображения, необходимо создать новый Action, который будет обращаться к слоту cancelImg(), где необходимо вызвать функцию *showImage(currentImagePath);*
2. Для создания возможности изменения размера изображения необходимо иcпользовать функцию

*OpenCV void cv::resize(inputArray,OutputArray,Size,fx,fy,flag)*, где

*inputArray* – входное изображение

*OutputArray* – выходное изображение

*Size* – размер выходного изображения

*Fx* – масштабный коэффициент по x

*Fy* – масштабный коэффициент по y

*Flag* – интерполяционный метод

1. **Разработанное решение**

Исходники данного проекта доступны на GitHub по ссылке:  
https://github.com/ivan-kuzin/Pattern-recognition/tree/main/lab\_2

Запустить разработанную программу возможно из папки *release* (дополнительно приложена в виде архива).

Исходный код плагина доступен в папке plugins\_code (Дополнительно приложена в виде архива).

1. **Вывод**

В ходе выполнения лабораторной работы я изучил принципы работы с базовыми алгоритмами обработки изображений и разработал программу для обработки изображений.