**Лабораторная работа 2**

**Цель работы:**

* Изучить поэлементные операции, линейное контрастирование и алгоритмы глобальной пороговой обработки.
* Разработать класс реализации вышеперечисленных операций
* Разработать приложение с графическим интерфейсом, в котором пользователь сможет загружать изображения, применять вышеперечисленные операции и сохранять изображение.

**Использованные средства разработки:**

* язык Java и библиотека openCV

**Ход работы:**

1. Были разработаны классы View, Model, Controller. View отвечает за простейший графический интерфейс, Model отвечает за реализацию алгоритмов. Controller обеспечивает связь между этими двумя классами.

2. Изображение хранится в классе Model, объекты которого принадлежат классу Tab, который отвечает за вкладки на окне изображения. Таким образом в программу можно загружать несколько изображений.

3. Класс Preview позволяет увидеть результат применения выбранной операции, не изменяя исходное изображение.

**Вывод:**

В ходе выполнения работы я:

* закрепил знания по следующим темам: “Поэлементные операции”, “глобальная пороговая обработка”;
* создал приложение, позволяющее редактировать изображения;
* получил опыт в работе с библиотекой openCV
* получил дополнительный опыт в разработке приложений на Java;

Исходное изображение:



После применение операции «Add scalar» со значением 100 и -100 соответственно:



Как видно из изображений, данный метод повышает или уменьшает яркость всех пикселей изображения, причем изменение происходит равномерно.

Исходное изображение:



После применения операции “Negative”:



Данный метод просто инвертирует пиксели исходного изображения.

Исходное изображение:



После применения операции “Multiply”:



Как видно из изображений, умножение на скаляр повышает яркость, но делает это неравномерно, в отличии от прибавления скаляра. Таким образом, изначально яркие участки будут становится еще ярче, в то время как темные участки будут либо светлеть незначительно, либо и вовсе не изменят значения (если пиксель черного цвета, т.е. его значение 0).

Исходное изображение:



После применения операции «Power» со значениями 0.3 и 3 соответственно:



Операция «Возведение в степень» меняет резкость изображения. Как видно из изображений, после возведения в степень 0.3 изображение стало мнее резким, стало сложнее отличить отдельные элементы, например, оттенки шерсти на морде кота. После возведения в 3 степень светлые участки шерсти на морде кота стали более различимы.

Исходное изображение:



После применения операции «Log»:



Операция “Log” улучшает значения нижних пикселей. На этом изображении львица не является единственным фокусирующим объектом. На ровне с ней куст справа. Также трава на заднем плане мешает фокусировке. После применения операции «Log» львица становится фокусирующим объектом на изображении

Исходное изображение:



После применения операции «Linear Contrast»:

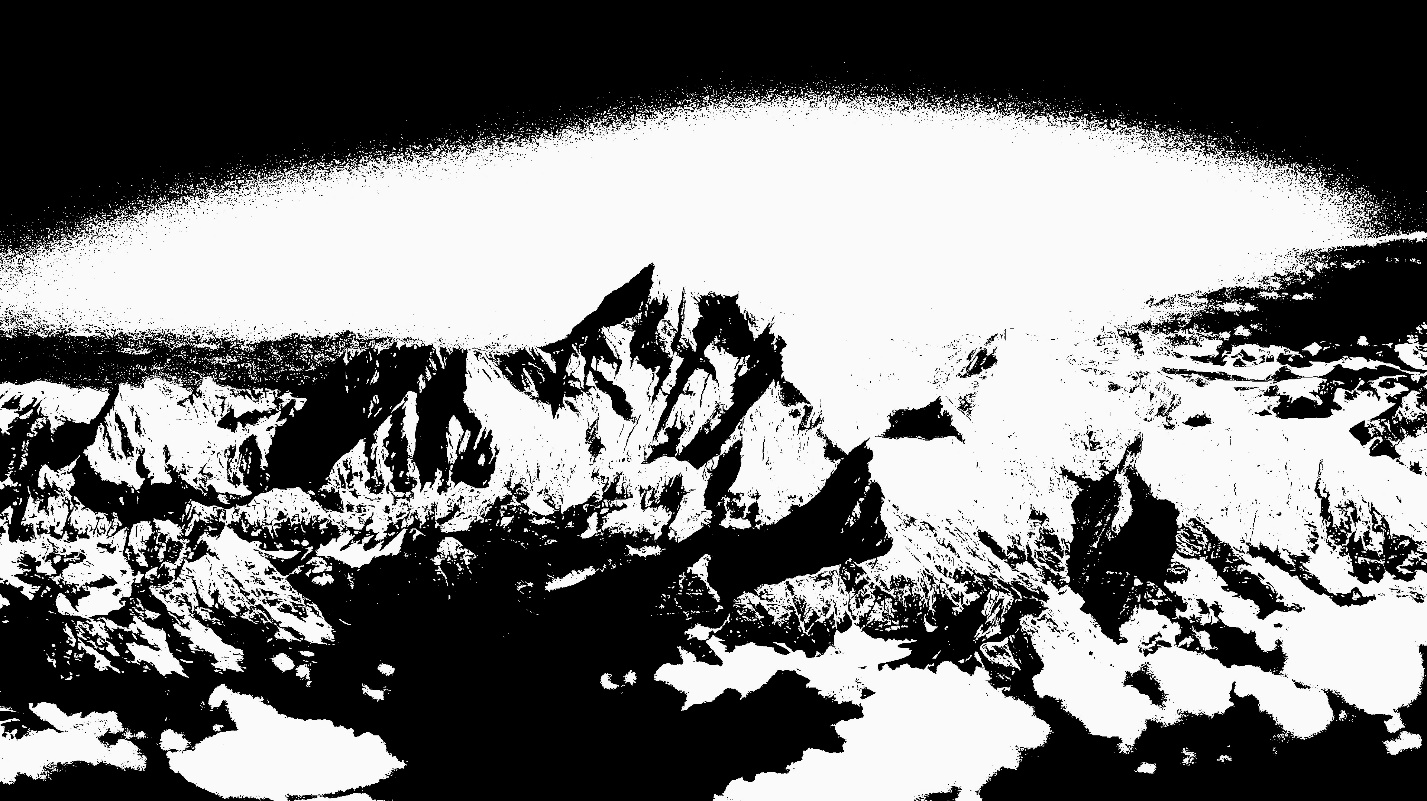


На исходном изображении щенок сливается с фоном. После применения линейного контрастирования щенок четко виден на изображении.

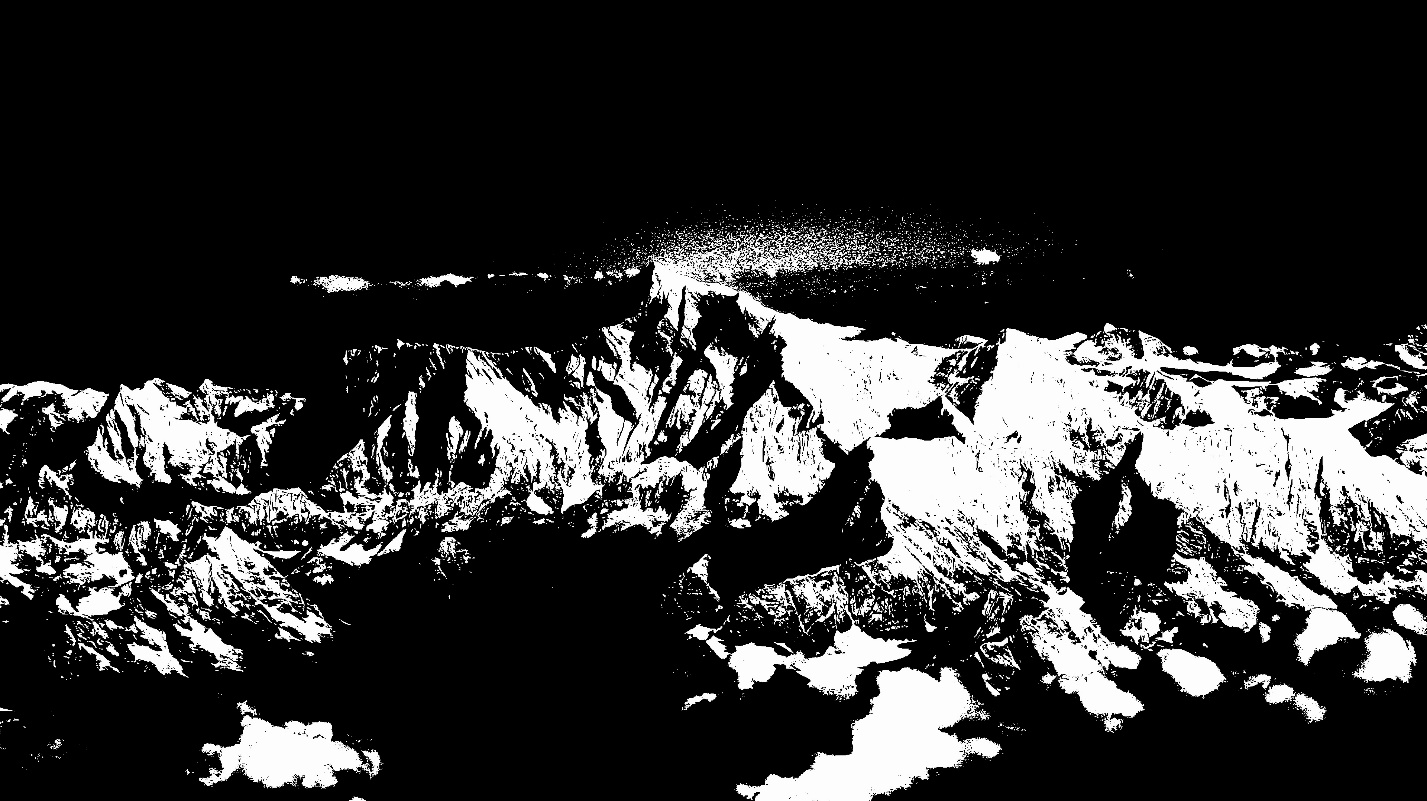
Исходное изображение:



После применения Глобальной пороговой обработки с поиском порога по гистограмме:



После применения глобальной пороговой обработки с поиском порога по методу Отсу:



При бинаризации с поиском порога по гистограмме были потеряны облака, однако четко виден контур гор. В алгоритме поиска порога по гистограмме начальное значение порога выбирается, как среднее между минимумом и максимумом, а затем пиксели разбиваются на группы по отношению к порогу. Затем по этим группам значение порога корректируется. И так до тех пор, пока разность между текущим и предыдущим порогом не будет меньше какого-то заданного эпсилон (в моем случае это 10). Из-за этого после обработки не различимы облака, ибо значения их пикселей не сильно больше значения пикселей неба. Алгоритм поиска порога Отсу работает на основе градиента, из-за этого на изображении после обработки можно отличить облака, а также ряби на границе между черным и белым. Контур гор все также четко прослеживается.