

1

Будем пользоваться библиотекой OpenCV. Так как она в конфликте с Jupyter пишем соответствующую команду, чтобы смотреть на картинки.

С таким исходным изображением будем работать:

RGB



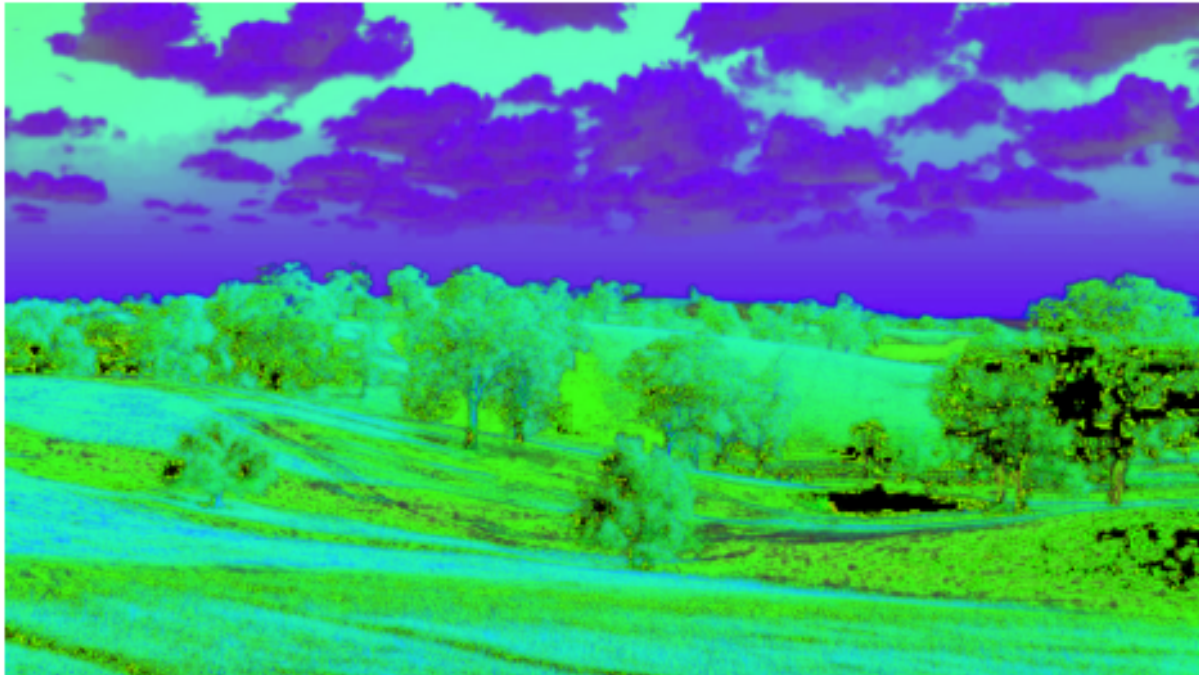
2

Преобразовали исходник в grayscale & HSV.

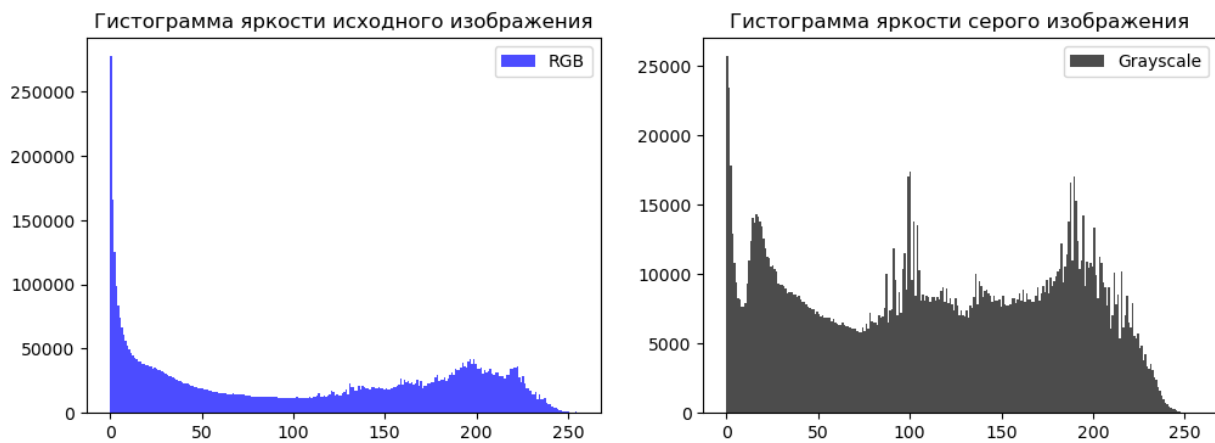
Grayscale



HSV



Гистограммы яркости для обоих выглядят след. образом:



Видим, что в RGB гистограмма более разнообразна, т.к. каждый цветовой канал вносит свой вклад в общую яркость. В Grayscale гистограмма выглядит более симметричной, т.к. Здесь распределение только одного канала яркости.

3

К серому изображению применили Гауссово сглаживание с разными значениями сигма:

Гауссово сглаживание с разными значениями σ



При увеличении значения сигма размытие становится сильнее.

1. Сигма = 1: минимальное сглаживание, детали изображения хорошо сохраняются
2. Сигма = 2: края становятся мягче
3. Сигма = 3: изображение теряет резкость, мелкие детали исчезают

Потом посмотрели на медианный фильтр с разными размерами ядра:

Медианный фильтр с разными размерами ядра



При увеличении размера ядра медианный фильтр удаляет больше шумов, но также размывает детали.

1. Kernel = 3: слабая фильтрация, сохраняет резкость
2. Kernel = 5: устраняет шум, но слегка сглаживает границы
3. Kernel = 7: более заметное размытие, мелкие текстуры теряются

В отличие от Гаусса, медианный фильтр лучше убирает соляно-перечный шум, но при больших ядрах искажает формы

В конце применили фильтр Лапласа для повышения резкости:

Фильтр Лапласа



Видим, что изображение действительно получилось довольно резким.

4

Здесь использовали оператор Собеля для определения горизонтальных и вертикальных градиентов, применили алгоритм Кэнни для обнаружения краев и использовали детектор углов Харриса для выявления угловых точек. В итоге наложили результаты трех операций на исходное изображение и получили такую картинку:

Наложенные на исходное изображение три операции



Оператор Собеля вычисляет градиенты, сильные градиенты окрашиваются в **красный** цвет. Алгоритм Кэнни обнаруживает границы, они окрашиваются в **зеленый** цвет. Детектор углов Harris находит углы, которые отображаются как **синие** точки.

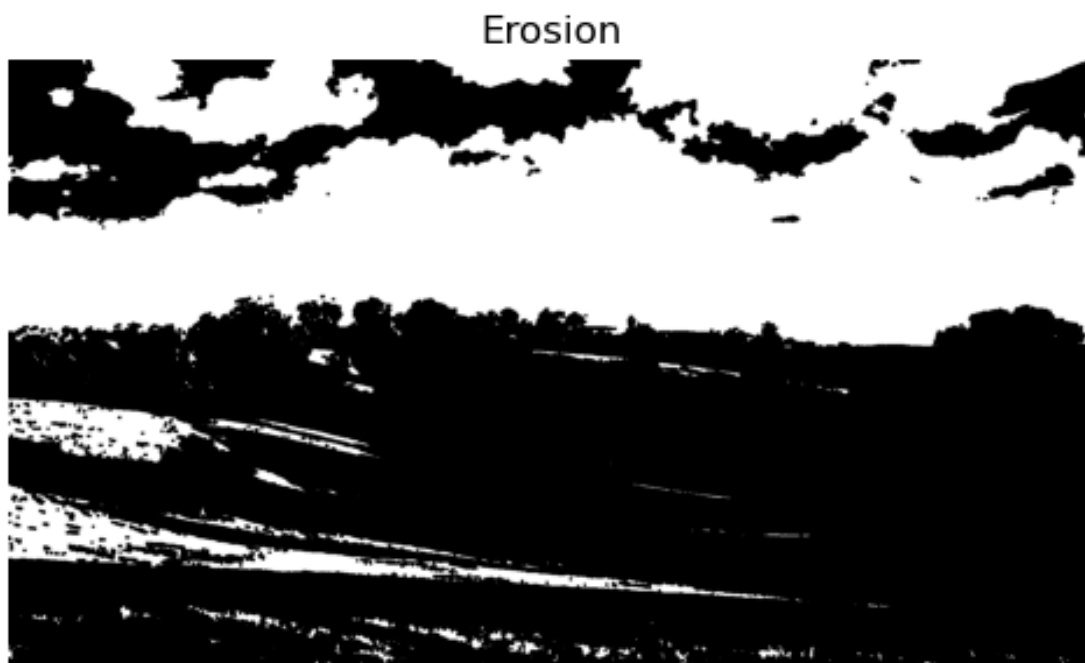
5

Бинаризовали серое изображение с помощью пороговой сегментации (127):



Бинаризация позволяет выделить объекты на фоне, чем выше порог, тем меньше остается деталей.

Затем применили операции эрозии и дилатации:



Эрозия удаляет мелкие детали и тонкие линии. Довольно хорошо разделяет объекты.

Dilation



Дилатация расширяет светлые области, заполняет разрывы. Восстанавливает контуры после эрозии.