

ИССЛЕДОВАНИЕ PYTHON БИБЛИОТЕК ДЛЯ ЛИНЕЙНОЙ ФИЛЬТРАЦИИ РАСТРОВЫХ ИЗОБРАЖЕНИЙ

Манасян А. Л.¹, Дёмин А.Ю.^{2,3}

¹ ТПУ ИЩИТР ОИТ, 8В13 e-mail: alm5@tpu.ru

² ТПУ ИЩИТР ОИТ, e-mail: ad@tpu.ru

³ Томский государственный университет, доцент ИФПМКН

Аннотация

Данное исследование оценивает эффективность различных Python библиотек для линейной фильтрации растровых изображений. Анализируются библиотеки, такие как OpenCV, Pillow, scikit-image и SciPy. Исследование включает сравнение производительности и качества фильтрации каждой библиотеки для обработки растровых изображений.

Ключевые слова: Фильтрация изображений. Линейная фильтрация. Image processing.

Введение

Анализ реализации алгоритмов фильтрации изображений на Python является актуальной темой современной компьютерной графики и обработки изображений. В связи с растущим спросом на обработку изображений в таких областях, как компьютерное зрение [1], медицинская диагностика [2], развлекательная индустрия и многих других [3], алгоритмы фильтрации изображений становятся все более важными для исследований и практического применения. Исследование реализации алгоритмов фильтрации изображений позволит оценить их производительность и эффективность. Такой анализ может стать основой для выбора наилучшей библиотеки при разработке программ обработки изображений.

Теоретическая часть

Под термином «фильтрация изображений» понимается процесс обработки изображений, в результате которого получается изображение того же размера, что и исходное, с применением определённых правил, называемых фильтрами. Обычно цвет каждого пикселя в полученном изображении зависит от цветов пикселей, расположенных в его окрестности на исходном изображении [4]. Фильтрация изображений является важной составляющей в задачах компьютерного зрения, распознавания образов и обработки изображений.

В данном исследовании планируется использование таких типов фильтров, как фильтр Гаусса, контрастоповышающая фильтрация и фильтр обнаружения границ, для модификации изображений. Для этого проведём анализ четырёх популярных библиотек Python с целью определения их эффективности при применении указанных фильтров к изображениям.

Практическая часть

Создадим набор тестовых изображений различного размера, чтобы оценить производительность алгоритмов фильтрации на различных объёмах данных.

Для проведения сравнительного анализа реализации фильтрации изображений с учётом производительности на Python используем следующие библиотеки:

- Воспользуемся библиотекой time для измерения времени выполнения определённых участков кода.

- Для чтения и вывода изображения воспользуемся библиотекой matplotlib.

Построим графики зависимости скорости выполнения каждого фильтра от количества пикселей в Jupyter Notebook.

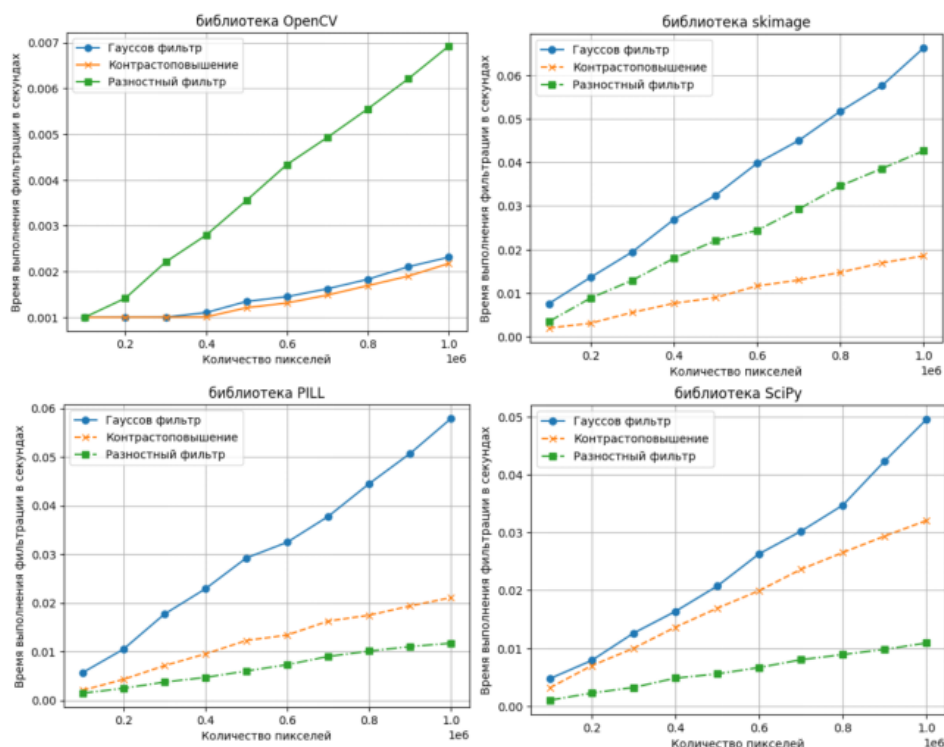


Рис. 1. Графики сложности алгоритмов, реализующих линейную фильтрацию в различных библиотеках Python

Для оценки результата фильтрации рассмотрим исходные и полученные изображения и графики, показывающие время выполнения одного и того же фильтра при использовании разных библиотек.

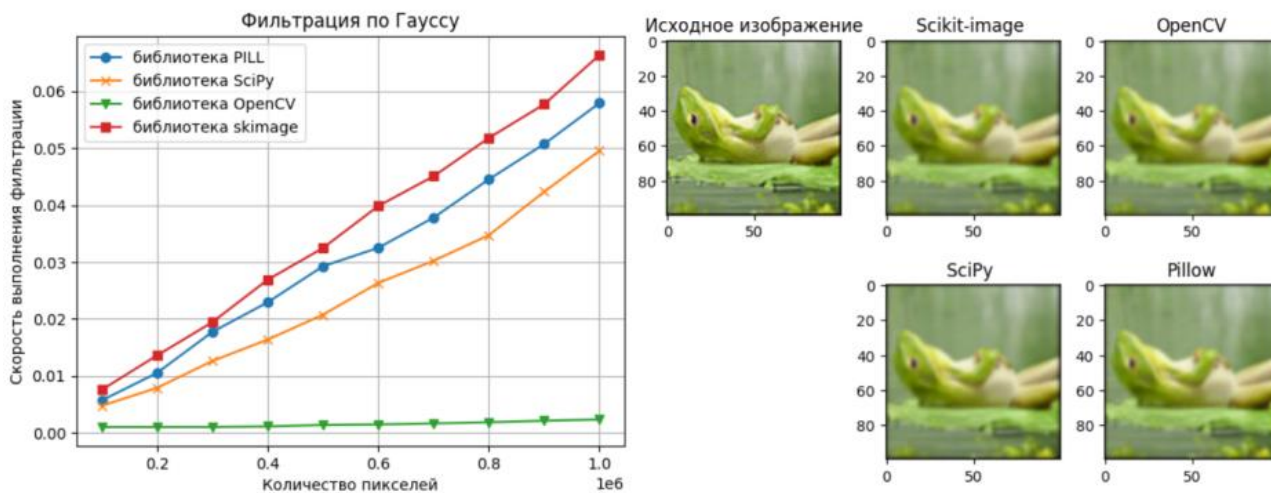
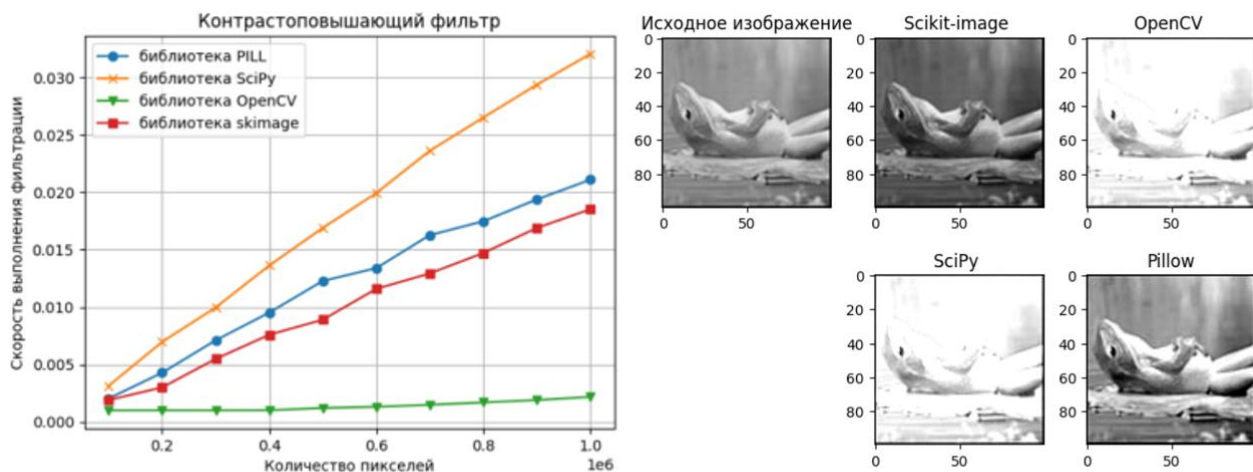


Рис. 2. Результаты фильтрации по Гауссу

По результатам анализа графиков и изображений можно сделать вывод, что оптимальным выбором для применения фильтрации по Гауссу является библиотека OpenCV. Данная библиотека демонстрирует значительно более высокую скорость при выполнении указанной задачи по сравнению с фильтрами других библиотек. Например, среднее время выполнения фильтрации самого крупного изображения (разрешение 1000x1000) составляет 0.0023 секунды, в то время как для остальных библиотек данное значение составляет 0.05 секунды и выше.



9

Рис. 3. Результаты контрастоповышающей фильтрации

По результатам анализа контрастоповышающей фильтрации изображений с одинаковыми параметрами следует отметить фильтр, представленный в библиотеке Pill, который демонстрирует удовлетворительное среднее время выполнения операций обработки изображений. Несмотря на высокую производительность фильтрации изображений в OpenCV, окончательное обработанное изображение с использованием аналогичных параметров не соответствовало ожиданиям в отношении качества и желаемого эффекта.

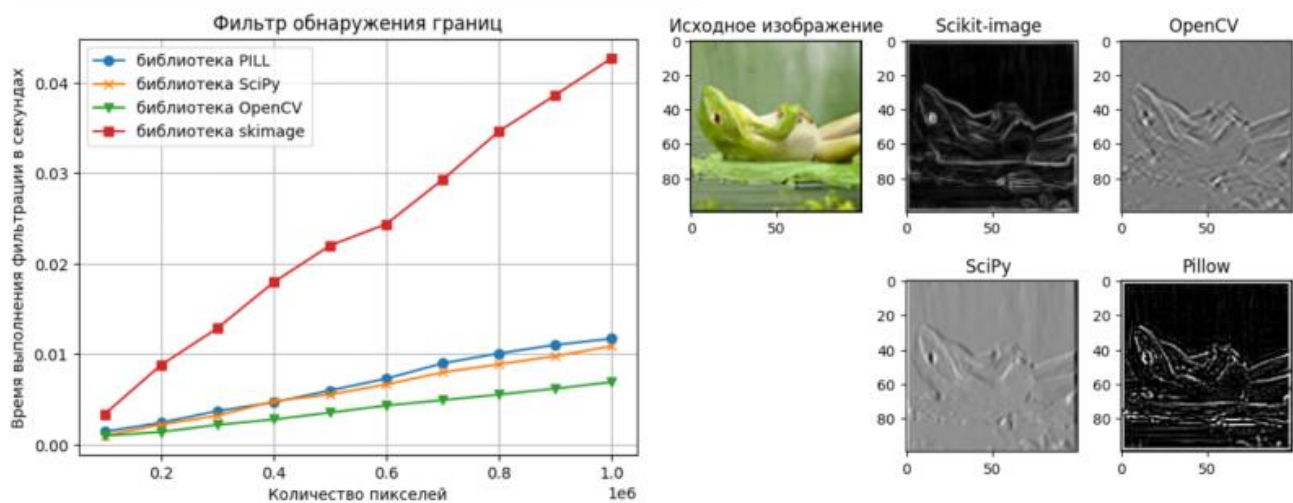


Рис. 4. Результаты фильтрации обнаружения границ

В результате проведения фильтрации с целью обнаружения границ наиболее эффективными библиотеками оказались SciPy и OpenCV. Эти библиотеки продемонстрировали оптимальные временные характеристики выполнения фильтрационных процессов и выдали окончательные изображения, соответствующие поставленным требованиям. Рекомендуется воздержаться от применения библиотеки Scikit-image в данной задаче, поскольку она показала наибольшее время обработки изображений.

Анализ результатов и рекомендации

После анализа четырёх библиотек, проведём обобщённые выводы по каждой из них в контексте наилучшего подхода к задаче фильтрации.

OpenCV. Все три фильтра данной библиотеки показали высокую скорость работы по сравнению с аналогичными библиотеками. Несмотря на более длительное время выполнения фильтра обнаружения границ (в среднем 0.007 секунды) по сравнению с фильтром Гаусса и контрастоувеличивающим фильтром (в среднем 0.002 секунды), этот показатель значительно ниже средних временных параметров аналогичных фильтров из других библиотек. Проведённый анализ конечных изображений подтверждает рекомендацию использовать данный фильтр для задач размытия и выявления границ

Python Imaging Library (PIL). Исследование показало, что данная библиотека демонстрирует превосходное качество выходных изображений при использовании контрастоувеличивающей фильтрации по сравнению с другими аналогичными библиотеками. Кроме того, её результаты по Гауссовской фильтрации оказались на уровне средних значений. Обобщенно все рассмотренные фильтры показали сравнимую среднюю скорость выполнения операций.

SciPy. Наиболее впечатляющие результаты были достигнуты в области фильтрации обнаружения границ, где библиотека продемонстрировала второе наилучшее время выполнения и соответствующее качественное окончательное изображение. Не рекомендуется использовать эту библиотеку для задач контрастоувеличения из-за наиболее длительного времени выполнения фильтрации в соответствующей области применения. По фильтрации Гаусса библиотека показала средние показатели.

Scikit-image. Проведённые исследования показали, что данная библиотека выделяется самым продолжительным временем выполнения операций фильтрации по Гауссу и обнаружению границ. В то же время, в контексте задач контрастоувеличения, фильтр, представленный в этой библиотеке, демонстрировал средние временные характеристики, при этом обеспечивая одно из наилучших качеств выходных изображений среди изученных методов.

Заключение

В данной работе изучены различные методы линейной фильтрации и их реализация в наиболее популярных библиотеках Python. Для оценки качества полученных результатов в фильтрации был подобран ряд изображений. Были построены графики сложности алгоритмов фильтрации для различных библиотек, на основе которых проведён анализ их производительности, и предложены рекомендации по их использованию. В дальнейшем планируется исследование кода библиотек с целью разработки эффективных высокоскоростных методов обработки изображений в Python.

Список использованных источников

1. Методы фильтрации изображений в рентгеновской компьютерной томографии. – Текст : электронный // URL:<https://cyberleninka.ru/article/n/metody-filtratsii-izobrazheniy-v-rentgenovskoy-kompyuternoy-tomografii/viewer> (дата обращения: 10.04.2024).
2. Методы фильтрации изображений в рентгеновской компьютерной томографии. – Текст: электронный // CyberLeninka. – URL: <https://cyberleninka.ru/article/n/metody-filtratsii-izobrazheniy-v-rentgenovskoy-kompyuternoy-tomografii/viewer> (дата обращения: 10.04.2024).
3. Filtering in Computer Graphics. – Текст: электронный // GeeksforGeeks. – URL: <https://www.geeksforgeeks.org/filtering-in-computer-graphics/> (дата обращения: 10.04.2024).
4. Растровые алгоритмы– Текст : электронный // – URL: https://portal.tpu.ru/SHARED/a/AD/Education/Tab2/CG_filtering.pdf (дата обращения 10.04.2024).