

# 1. Эксперимент

В данном эксперименте проводится исследование производительности реализованной библиотеки по обработке изображений на CPU и GPU. Основная задача — выяснить, на каком из процессоров (центральной или графической) с использованием оптимального количества агентов будет быстрее обработан массив изображений с заданными трансформациями. В качестве оптимального количества агентов рассматривается количество логических процессоров системы.

## 1.1. Условия эксперимента

Для экспериментов использовалась рабочая станция с процессором Intel Core i5-10300H с тактовой частотой 2.50GHz, RAM DDR4 объемом 8гб, видекартой NVIDIA GeForce GTX 1650 Ti с объемом памяти 4гб под управлением ОС Windows 10.

## 1.2. Исследовательские вопросы

- Какой из методов по обработке массива изображений быстрее, параллельная версия на CPU с использованием количества потоков, равных количеству логических процессоров или версия обработки на GPU.

## 1.3. Метрики

В качестве метрик производительности используется время, требуемое на завершение обработки. Показатели времени полу-

чены с помощью библиотеки `BenchmarkDotNet v0.13.5`<sup>1</sup>. Для замеров были выбраны стандартные настройки по «прогреву» и количеству итераций для измерений.

## 1.4. Результаты

Замеры производились на массиве из 30 изображений с котами. Каждое изображение было повернуто два раза на 90° по часовой стрелке, затем отражено по горизонтали и, наконец, к каждому изображению был применён фильтр `FishEye`. Полученные результаты приведены в таблице 1.

Таблица 1: Производительность обработки массива изображений с помощью CPU и GPU на 8 агентах.

METHOD	TIME(S)
CpuProcessing	7.83 ± 0.08
GpuProcessing	3.86 ± 0.06

Данные замеры показали, что при распараллеливании задачи по обработке массива на оптимальное количество потоков для CPU, время обработки массива изображений на центральном процессоре будет превышать время обработки того же массива на графическом процессоре в два раза.

Результаты, полученные с помощью `BenchmarkDotNet` были проверены на адекватность с помощью библиотек `SciPy v1.10.1`<sup>2</sup>, `NumPy v1.24.3`<sup>3</sup> и `Matplotlib v3.7.1`<sup>4</sup> на языке Python. Выбор-

---

<sup>1</sup>Репозиторий библиотеки `BenchmarkDotNet`: <https://github.com/dotnet/BenchmarkDotNet> (дата доступа: 25 мая 2023 г.).

<sup>2</sup>Сайт библиотеки `SciPy`: <https://scipy.org/> (дата доступа: 25 мая 2023 г.).

<sup>3</sup>Сайт библиотеки `NumPy`: <https://numpy.org/> (дата доступа: 25 мая 2023 г.).

<sup>4</sup>Сайт библиотеки `MatPlotLib`: <https://matplotlib.org/> (дата доступа: 25 мая 2023 г.).

ку можно принять за нормальное распределение, так как был успешно пройден тест Шапиро-Уилка, а так же зачтён критерий согласия Пирсона. В случае замеров обработки на CPU  $p$ value равняется 0.64 и 0.72. В случае GPU — 0.71 и 0.57. Соответствующие им гистограммы приведены на рисунке 1 и рисунке 2.

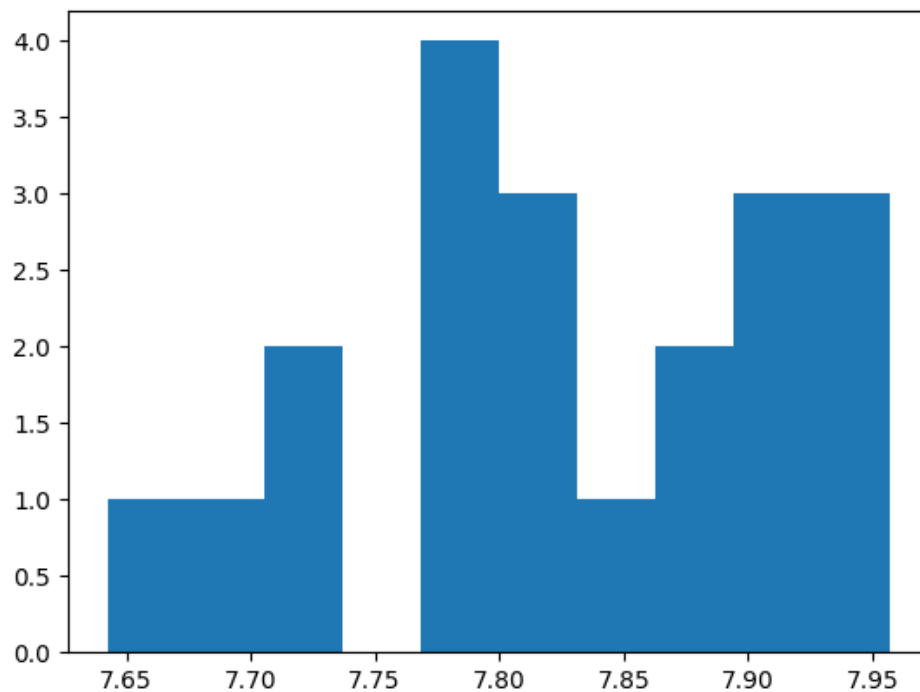


Рис. 1: Гистограмма распределения замеров времени по обработке массива изображений на CPU

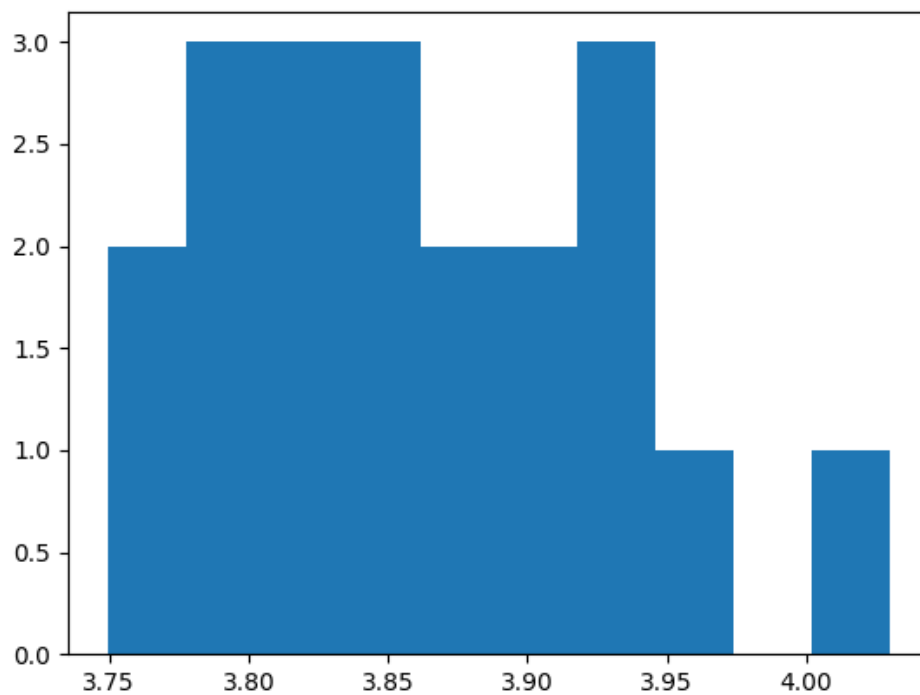


Рис. 2: Гистограмма распределения замеров времени по обработке массива изображений на GPU

Обработанные версии на GPU и CPU идентичны. Для примера, оригинальное изображение 3, обработанное на CPU 4, обработанное на GPU 5.



Рис. 3: Оригинальное изображение

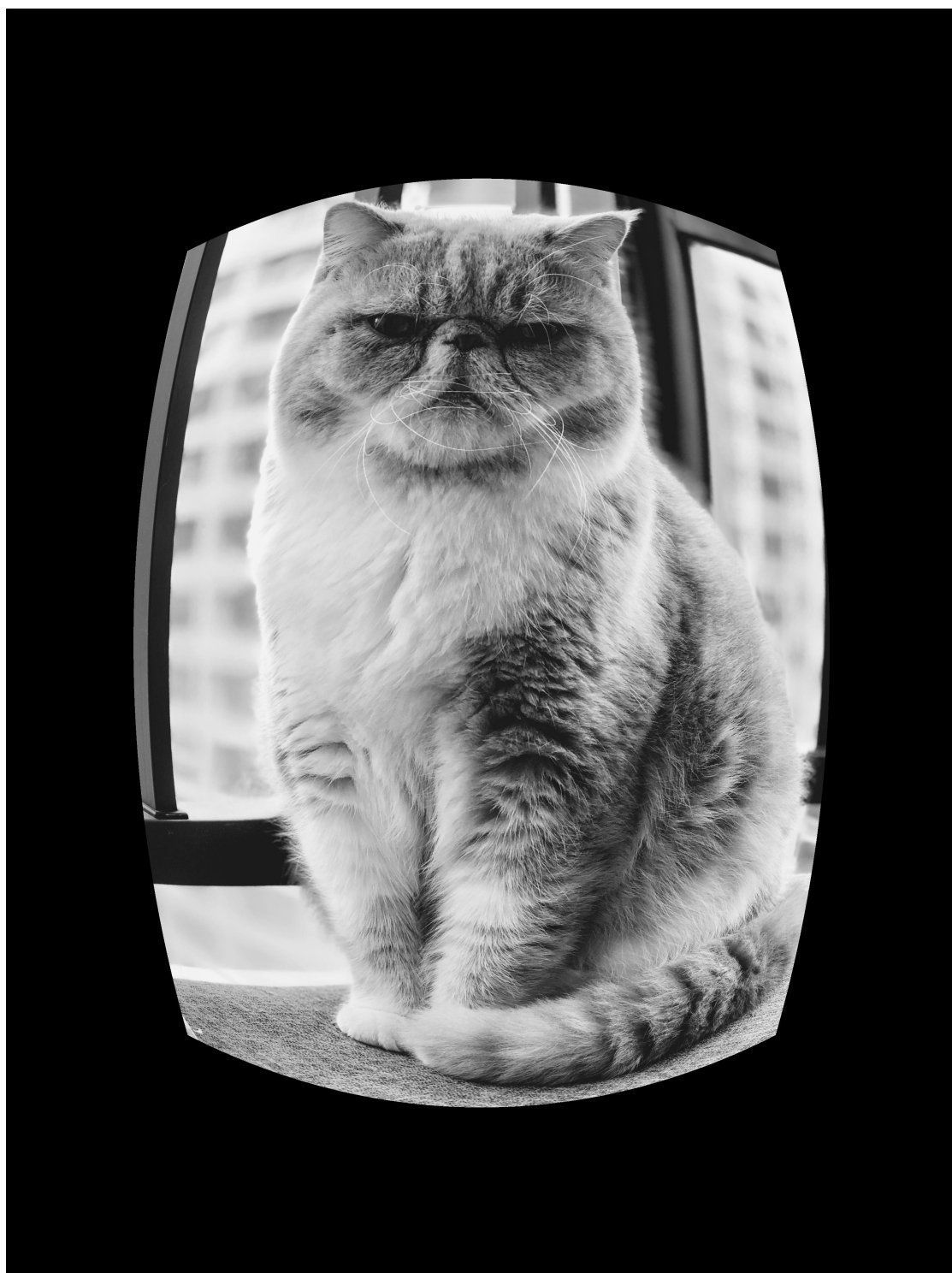


Рис. 4: Изображение после обработки на CPU



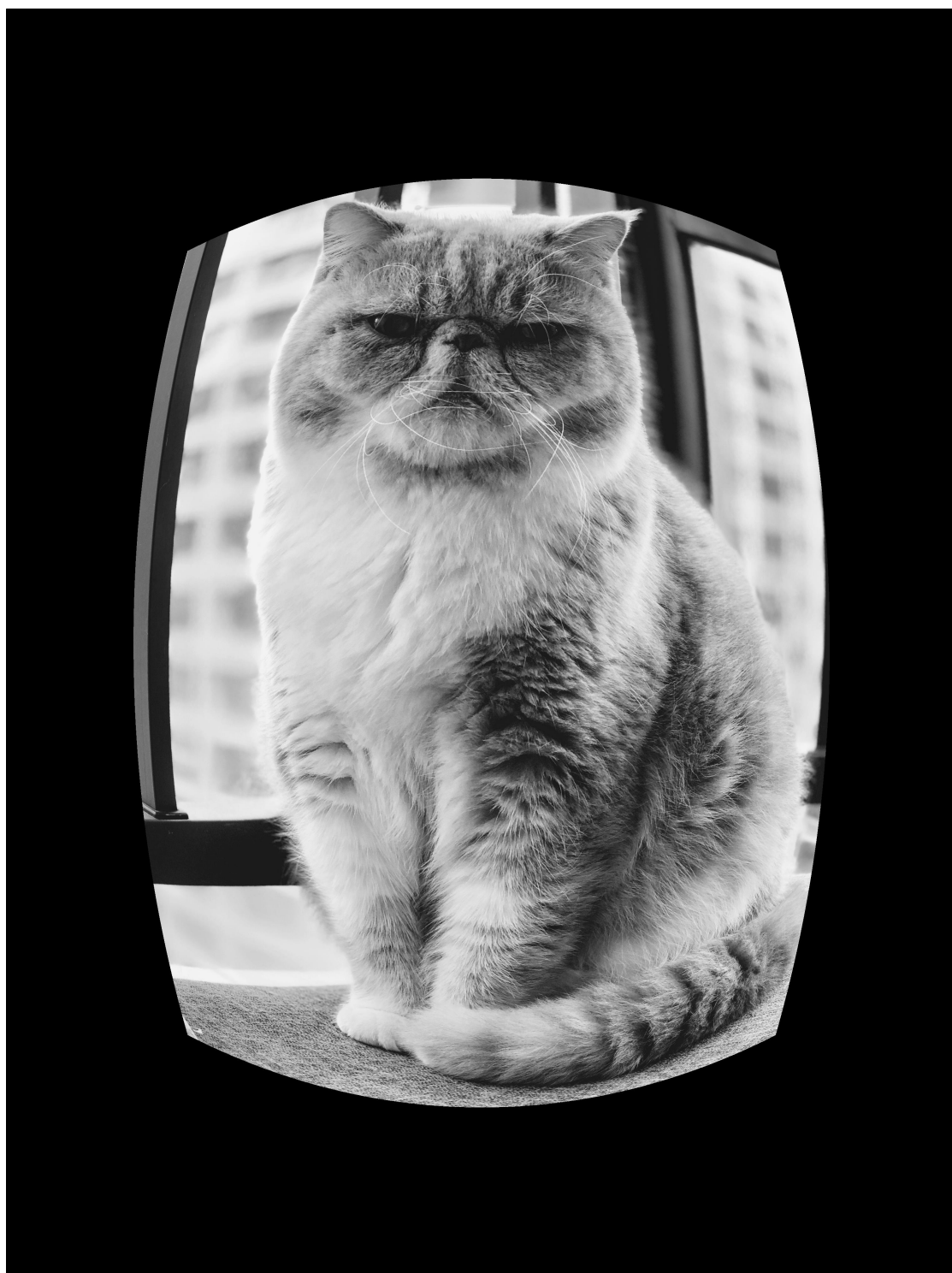


Рис. 5: Изображение после обработки на GPU

#### 1.4.1. RQ1

Исходя из полученных данных можно сделать вывод, что обработка массива изображений на GPU в два раза опережает параллельную обработку массива изображений на CPU при использовании количества потоков, равное количеству логических процессоров системы. Такой результат можно объяснить тем, что в графических процессорах используется программная модель SIMT (Single instruction, multiple threads), что позволяет выполнять одну и ту же инструкцию на гораздо большем числе независимых вычислителей.