МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №3
по курсу «Программирование графических процессоров»

Классификация и кластеризация изображений на GPU

Выполнил: Н.И. Лохматов
Группа: 8О-406Б
Преподаватель: А.Ю. Морозов

Москва, 2024

Условие

1. Цель работы: научиться использовать GPU для классификации и кластеризации изображений. Использование константной памяти и одномерной сетки потоков
2. Вариант 7: метод спектрального угла

Программное и аппаратное обеспечение

1. Графический процессор: Nvidia GeForce RTX 3050 Mobile
    a. Количество потоковых процессоров: 2560
    b. Частота ядра: 1552 МГц
    c. Количество транзисторов: 8.7 млрд
    d. Тех. процесс: 8 нм
    e. Энергопотребление: 80 Вт
2. ОС: Ubuntu 22.04
3. Текстовый редактор: VS Code
4. Компилятор: nvcc

Метод решения

Основная задача заключается в том, чтобы для каждого пикселя изображения найти класс, которому он наиболее соответствует, используя угол между вектором пикселя и средним вектором класса. Входные данные представляют те же данные, что и в ЛР2. На этапе предобработки вычисляются средние значения для каждого класса (координаты пикселей для расчётов передаются пользователем). Затем в ядре происходит сравнение каждого пикселя с каждым средним значением класса с использованием нормализованного косинусного угла. Для каждого пикселя определяется класс с максимальным значением косинусного угла.

Описание программы

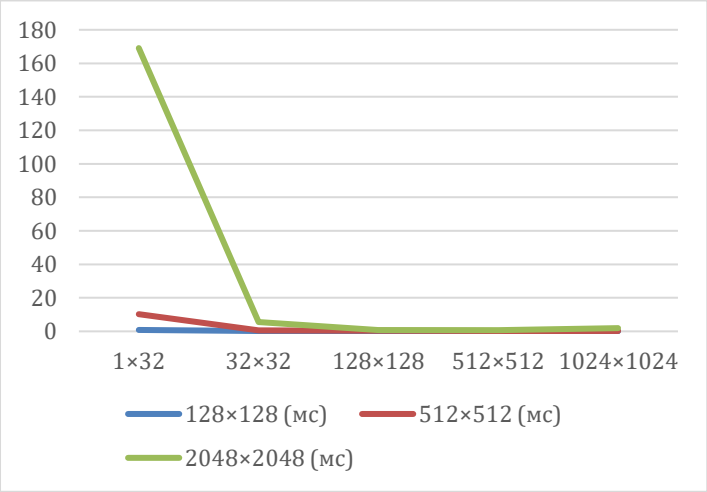
Программа состоит из основного файла, в котором реализована основная логика, макрос для обработки ошибок, а также CUDA-ядро для выполнения алгоритма на GPU. На первом этапе данные считываются из бинарного файла и копируются в память устройства. Для каждого пикселя вычисляется спектральный угол с каждым из классов, используя предобработанные средние значения, хранящиеся в константной памяти. В качестве результата в альфа-канал изображения записывается номер класса с максимальным значением косинусного угла. Программа также обрабатывает ошибки и корректно освобождает выделенные ресурсы.

Результаты

1. Зависимость времени выполнения программы от количества используемых потоков.

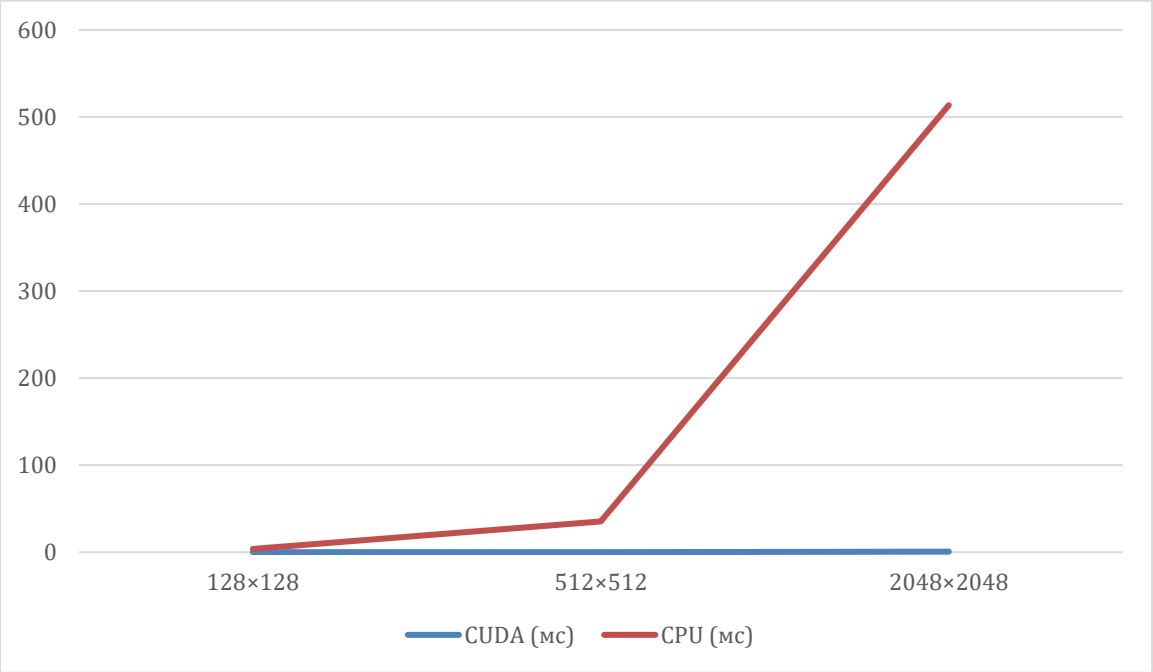
Количество потоков	Время, w = 128px, h = 128px (мс)	Время, w = 512px, h = 512px (мс)	Время, w = 2048px, h = 2048px (мс)

1×32	0.84	10.25	169.07
32×32	0.19	0.54	5.54
128×128	0.15	0.31	0.76
512×512	0.12	0.27	0.72
1024×1024	**0.11**	**0.24**	**0.67**



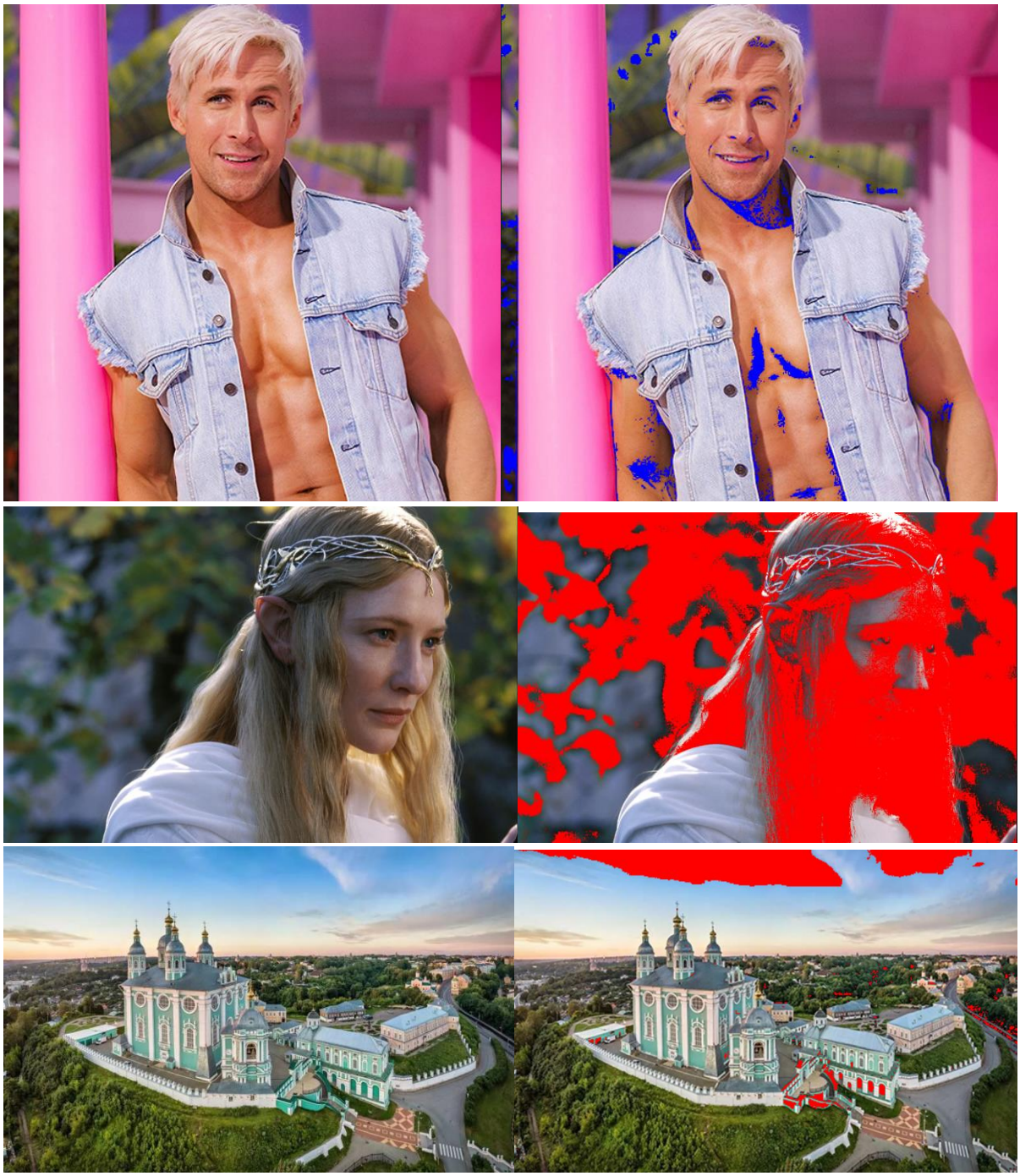
2. Сравнение программы на CUDA с 1024×1024 потоками и программы на CPU с одним потоком

Размер изображений	Время CUDA (мс)	Время CPU (мс)
128×128	**0,12**	**3.82**
512×512	0,22	35.47
2048×2048	0,69	513.61



3. Примеры обработанных изображений

Для того, чтобы наглядно показать, как работает кластеризация, немного изменим код. Альфа-канал трогать не будет, но если пиксель принадлежит какому-либо классу, будем выделять его определённым цветом (контрастным для всего изображения).



Выводы

В ходе выполнения лабораторной работы были исследованы возможности использования GPU для классификации изображений с применением метода спектрального угла. Были успешно разработаны и реализованы алгоритмы, которые позволяют производить эффективную обработку данных на GPU, используя константную память и одномерную сетку потоков. В результате удалось добиться

точной классификации пикселей на основе анализа угла между вектором пикселя и средним значением классов. Программа показала высокую эффективность и правильность работы на основе заданного формата данных и алгоритма.