

**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:	И.А. Мариничев
Группа:	8О-408Б
Преподаватели:	К.Г. Крашенинников А.Ю. Морозов

Москва, 2022

Условие

Цель работы: научиться использовать GPU для классификации и кластеризации изображений. Использование константной памяти и одномерной сетки потоков.

Вариант 1: метод максимального правдоподобия.

Программное и аппаратное обеспечение

Compute capability	:	2.1
Name	:	GeForce GT 545
Total Global Memory	:	3150381056
Shared memory per block	:	49152
Registers per block	:	32768
Warp size	:	32
Max threads per block	:	(1024, 1024, 64)
Max block	:	(65535, 65535, 65535)
Total constant memory	:	65536
Multiprocessors count	:	3

Processor	:	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz
RAM	:	16 GB
Drive	:	349G

OS	:	Ubuntu 16.04.6 LTS
IDE	:	Visual Studio Code
Compiler	:	NVIDIA (R) Cuda compiler driver V7.5.17

Метод решения

Считываем входные данные, в том числе из файла. По обучающим данным рассчитываем на CPU вектор средних, матрицы ковариаций, определитель матрицы ковариаций и обратные матрицы ковариаций, затем нужные массивы копируем в константную память. После этого вызываем ядро, в котором для каждого пикселя по формуле ММП определяем предсказанный класс. В конце записываем результат в выходной файл и освобождаем память.

Описание программы

```
// выборочное мат ожидание
__constant__ double3 avg[32];

// log(|det(cov)|)
__constant__ double logAbsDet[32];
```

```
// обратная выборочная матрица ковариаций
__constant__ double covInv[3 * 32][3];

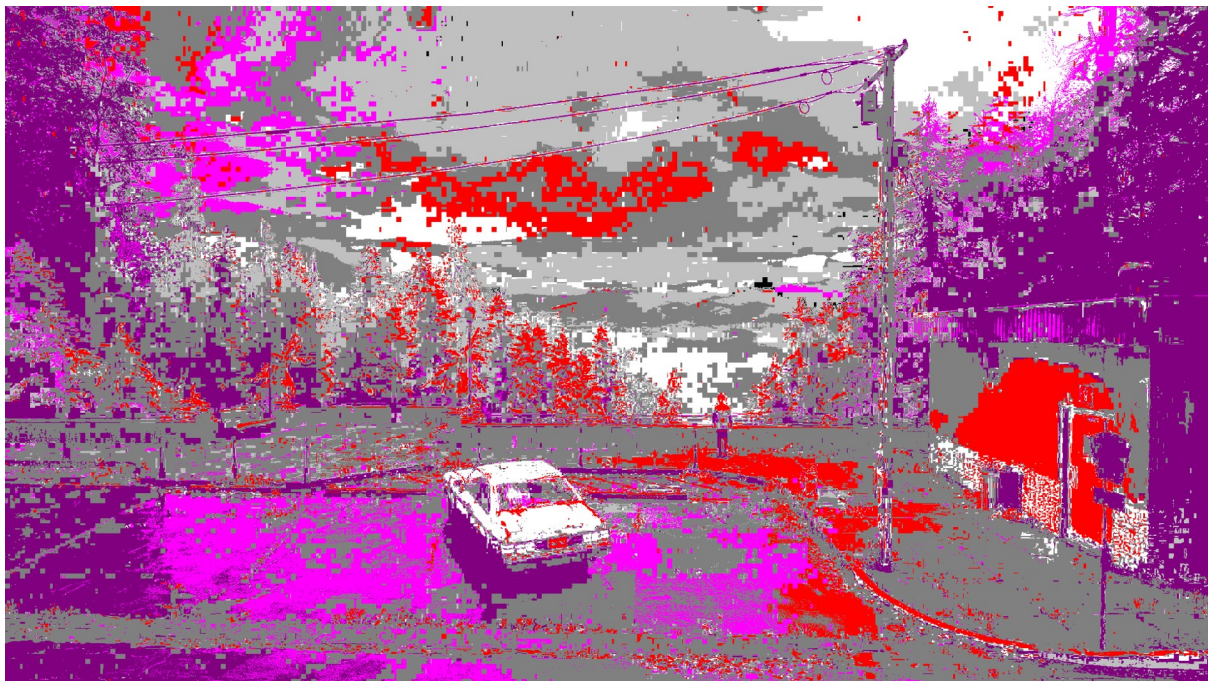
// дискриминантная функция
__device__ double D(uchar4 p, int I)

// метод максимального правдоподобия (ММП)
__global__ void MLE(uchar4 *data, int w, int h, int nc)
```

Результаты

Конфигурация	Размер теста (время указано в ms)					
	O(10 ³)	O(10 ⁴)	O(10 ⁵)	O(10 ⁶)	O(10 ⁷)	O(10 ⁸)
<<< 1, 32 >>>	0.274368	21.489729	138.679428	488.761566	20936.947266	105839.875000
<<< 32, 32 >>>	0.047008	2.123680	13.483776	43.437599	2060.410645	10215.825195
<<< 256, 256 >>>	0.047648	1.760736	11.071200	34.177055	1709.251221	8422.470703
<<< 512, 512 >>>	0.100224	1.849216	11.312864	34.537567	1716.713135	8455.764648
<<< 1024, 512 >>>	0.159488	1.904416	11.463168	34.793633	1716.880005	8455.961914
CPU	253.0	22876.0	85212.0	232836.0	1.12594e+07	5.54774e+07

Сравнение изображений



Результат разделения на 7 классов при случайном распределении классов в обучающей выборке.

Выводы

Как видно из результатов сравнения, GPU дает значительный прирост по времени, особенно на больших изображениях. Данный алгоритм относится к группе алгоритмов обучения с учителем, так для его работы нам нужно изначально знать для каждого пикселя исходный класс. Отсюда следует, что практическая применимость этого алгоритма довольно спорная, так как у нас вряд ли будет стоять задача, в которой нам заранее известен ответ и мы должны его же предсказать. Понятно, что формула пришла из теории вероятностей и мат. статистики и применяется, когда мы имеем данные о довольно крупной выборке, которая является подспорьем для предсказания на новых данных.