
СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

УДК 004.93'12

РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ МЕТОДА ВИОЛЫ-ДЖОНСА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ CUDA

А. В. Акимов, А. А. Сирота

Воронежский государственный университет

Поступила в редакцию 03.06.2014 г.

Аннотация. Описана реализация метода Виолы-Джонса с использованием технологии параллельных вычислений на графических процессорах CUDA. Приведены результаты экспериментов по сравнению параллельной и последовательной версий различных этапов выполнения алгоритма. Показаны преимущества применения технологий параллельных вычислений на графических процессорах при реализации метода Виолы-Джонса.

Ключевые слова: обработка изображений, детектирование лиц, метод Виолы-Джонса, бустинг, параллельные вычисления на графических процессорах, CUDA.

Annotation. Viola-Jones method realization using CUDA technology of parallel computing on graphics processing units is described. The results of experiments of comparison between parallel and serial versions of different stages of the algorithm are presented. The advantages of using parallel computing on graphics processing units for Viola-Jones method realization are demonstrated.

Keywords: image processing, face detection, Viola-Jones method, boosting, parallel computing on graphics processing units, CUDA.

ВВЕДЕНИЕ

Задача распознавания изображений находит себе применение во многих областях человеческой деятельности, зачастую требующих от используемого подхода высокой производительности и достоверности. Современные методы характеризуются приемлемой скоростью работы, сильно зависящей, однако, от размера анализируемого изображения и дискретности перемещения окна сканирования при поиске локализованных объектов.

Для повышения производительности методов распознавания изображений перспективным выглядит применение ресурсов графических ускорителей современных компьютеров,

как относительно дешевой альтернативы мощным многопроцессорным системам. Состоящие из множества расположенных на одной плате и одновременно работающих арифметико-логических устройств современные видеокарты во многих задачах (например, [1] и [2]) обеспечивают большую производительность обработки за счет возможности реализации параллельных вычислений, чем центральный процессор, взамен упрощения механизмов работы с памятью и уменьшения скорости обращения к ней. Для исследования возможности использования такого подхода нами был выбран метод Виолы-Джонса распознавания объектов на изображениях (как правило, лиц) ([3], [4]), широко зарекомендовавший себя во многих работах как быстрый и эффективный способ анализа изображений.

© Акимов А. В., Сирота А. А., 2014

Следует отметить, что известные работы обходят стороной вопросы реализации параллельной версии метода, лишь вскользь упоминая о ее наличии ([4] и [5]). Другие ([6] и [7]) – посвящены исключительно распараллеливанию процесса анализа изображений уже готовым, ранее обученным классификатором, тогда как существенный интерес представляет также сокращение огромных затрат ресурсов и машинного времени на его обучение, проистекающих из больших объемов требуемых для обучения исходных данных.

Метод Виолы-Джонса основан на применении целого множества взаимосвязанных подходов и технологий, в центре которых лежит идея построения точного или, как еще говорят, сильного классификатора путем комбинирования других менее точных или слабых классификаторов.

Предварительно рассмотрим ряд базовых понятий метода, которые далее будут использованы по тексту статьи.

Признаки Хаара. Обработка всего изображения при поиске локализованных объектов (лиц) проводится путем сканирования окнами различного размера. При первичном анализе содержимого сканирующихся окон или областей изображений, для которых производится принятие решения при классификации, в методе Виолы-Джонса используются признаки или примитивы, фактически реализующие вычисление спектральных коэффициентов ортогонального разложения Хаара. Их можно определить, как разности сумм интенсивностей пикселей в смежных областях изображения. Эти признаки могут быть разного типа и размера и занимать различные положения внутри сканирующего окна. Впервые в данном качестве примитивы Хаара были представлены в работе Папагеоргиу, Орена и Погиу [8].

Слабые классификаторы. Слабый классификатор в методе Виолы-Джонса представляет собой алгоритм принятия решения о классификации на основе значения, которое было получено при применении одного из признаков Хаара (1).

$$h(x, f, p, \theta) = \begin{cases} 1, & \text{если } pf(x) < p\theta; \\ 0, & \text{иначе.} \end{cases} \quad (1)$$

Здесь x – содержимое сканирующего окна, $f(x)$ – исходный признак Хаара, θ – граница, по итогам сравнения с которой результата применения признака Хаара принимается решение о классификации, p – параметр, определяющий большие или меньшие граничной величины θ результаты применения признака Хаара $f(x)$ будут вести к принятию решения об обнаружении объекта и $h(x, f, p, \theta)$ – итоговый слабый классификатор. Результат, равный единице, означает принятие решения о том, что внутри данного сканирующего окна был обнаружен объект.

Обучение слабого классификатора представляет собой определение значений параметров θ и p (1), обеспечивающих наибольшую точность детектирования на множестве изображений тренировочной выборки. Она вычисляется путем сложения весовых коэффициентов неправильно классифицированных изображений при данных значениях θ и p .

Иными словами, ищется такое положение между двумя соседними отсортированными результатами применения исходного признака Хаара, на основе которого строится данный слабый классификатор, к изображениям тренировочной выборки, которое будет наиболее корректно делить изображения тренировочной выборки на две группы с учетом их текущих весовых коэффициентов.

Бустинг или усиление слабых классификаторов. Процедура бустинга, предложенная в 1997 году Йоавом Фройндом и Робертом Шапире, [9] заключается в комбинировании множества слабых классификаторов в один сильный. Как алгоритм она представляется последовательностью действий по обучению слабых классификаторов, выбору наиболее точных из них и составлению их линейной комбинации (2). При этом соответствующие всем экземплярам обучающей выборки весовые коэффициенты (изначально они одинаковые) раз от разу модифицируются так, что каждый следующий слабый классификатор при своем обучении акцентирует внимание на тех изображениях тренировочного набора, с которыми не справились предыдущие. Также каждый член линейной комбинации имеет свой весовой коэффициент влияния

на итоговый результат применения сильно-го классификатора, определяемый степенью точности соответствующего слабого классификатора.

$$C(x) = \begin{cases} 1, & \text{если } \sum_{i=1}^T a_i h_i(x) \geq \frac{1}{2} \sum_{i=1}^T a_i \\ 0, & \text{иначе.} \end{cases} \quad (2)$$

Здесь $h_i(x) = h(x, f_i, p_i, \theta_i)$ – отдельный слабый классификатор со своими собственными параметрами из (1), a_i – коэффициент влияния данного слабого классификатора, T – число слабых классификаторов внутри сильного и $C(x)$ – итоговый сильный классификатор.

Каскад классификаторов. Для ускорения производительности метода авторами ([3] и [4]) используется следующий подход. Итоговый классификатор строится в виде каскада описанных выше сильных классификаторов, по одному на слой. При этом каждый из слоев каскада настраивается на достоверное распознавание как можно большего числа лиц, но учится отсеивать только те ложные положительные срабатывания, которые вернули предыдущие слои, каждый раз все более сложные с точки зрения их отличимости от искомым объектов. В итоге каждый последующий слой каскада содержит все большее количество слабых классификаторов, и при анализе запускается только тогда, когда предыдущий сообщает о возможности наличия в анализируемой области объекта поиска. Иначе текущее окно отбрасывается, и алгоритм переходит к следующему.

Применение обученного классификатора производится путем перемещения и масштабирования сканирующего окна. Осуществляется это посредством модифицирования координат обращения исходных признаков Хаара в соответствии с текущим размером и положением сканирующего окна.

Реализация последовательной и параллельной версий алгоритма на основе метода Виолы-Джонса. Для создания последовательной версии метода была использована среда Matlab. Этот выбор обусловлен наличием в ней уже множества реализованных математических операторов и функций, зачастую

отсутствующих в других языках программирования. Также среда Matlab содержит весь необходимый инструментарий, применимый для проведения анализа работы реализации метода. Для создания его параллельной версии в качестве механизма осуществления параллельных вычислений на графическом процессоре была выбрана архитектура CUDA, как единственно доступная из среды Matlab. При проведении всех экспериментов по оценке производительности алгоритмов в качестве испытательного стенда использовался ноутбук со следующими характеристиками:

- процессор Intel Core i5-3210M 2.5 ГГц с возможностью ускорения во время выполнения до 3.1 ГГц;
- 6 ГБ DDR3 оперативной памяти;
- видеокарта – NVIDIA GeForce GT 640M с 2 ГБ памяти и 384 ядрами CUDA (имеет возможность оперирования с числами с плавающей запятой двойной точности – double);
- ОС – 64-х битная версия Ubuntu 14.04.

Метод Виолы-Джонса можно разделить на два независимых процесса: обучение классификатора и анализ изображений готовым классификатором. Нас, в первую очередь, будет интересовать процесс обучения, который подразделяется на несколько следующих одна за другой фаз (рис.1).

Первая фаза — это подготовка или создание набора признаков Хаара. Производится перебор всех типов примитивов с учетом их возможных размеров и положений внутри сканирующего окна размером 24×24 пикселей. Такой размер обусловлен компромиссом между количеством разных признаков (порядка 120000 для окна размером 24×24), которые может вместить такое окно, и временем обучения, так как оно по сути представляет собой процесс построения слабых классификаторов на основе этого набора признаков и осуществление выбора лучших из них.

Вторая фаза представляет собой подготовку положительных экземпляров обучающей выборки и получение первичных результатов анализа этой выборки полным набором

признаков, сформированным во время предшествующей фазы. В качестве положительных экземпляров выборки используются заранее выбранные и должным образом вырезанные

изображения лиц. Этот набор результатов анализа положительных экземпляров тренировочной выборки будет использоваться на протяжении всего последующего процесса обучения, так как каждый слой строящегося каскада должен верно определять изображения из данного набора как лица.

Оставшиеся три фазы выполняются в цикле до тех пор, пока не будут получены требуемые значения достоверности работы алгоритма распознавания.

Третья фаза. На этой фазе ([3], [4] и [5]) производится подготовка и первичный анализ полным набором признаков Хаара необходимого числа отрицательных экземпляров выборки. (Оно, как и на предыдущей фазе, определяется компромиссом между временем обучения и степенью точности работы каждого отдельного слоя каскада.) В качестве отрицательных экземпляров выборки используется содержимое сканирующих окон, вызвавших ложные срабатывания последнего слоя, имеющегося на данный момент каскада классификаторов при обходе им в рабочем режиме заведомо не содержащих лица изображений. (В начале, когда еще не создан ни один слой каскада, производится просто сбор содержимого всех сканирующих окон до достижения заданного числа отрицательных экземпляров выборки.)

Таким образом, реализуется возможность обучения каждого следующего слоя каскада отсеиванию тех изображений, которые были неверно приняты за содержащие лица предыдущими его слоями.

Существует также альтернативный вариант подготовки набора отрицательных экземпляров выборки [10], при котором он формируется заранее из не содержащих лица изображений, и затем используются для тренировки всех слоев каскада. При этом каждому последующему слою достается уменьшающаяся от раза к разу часть исходного набора отрицательных экземпляров тренировочной выборки, с которой не смогли справиться предыдущие слои каскада. Первый подход показал себя более производительным, ввиду того, что в целом позволяет обработать большее число отрицательных экземпляров

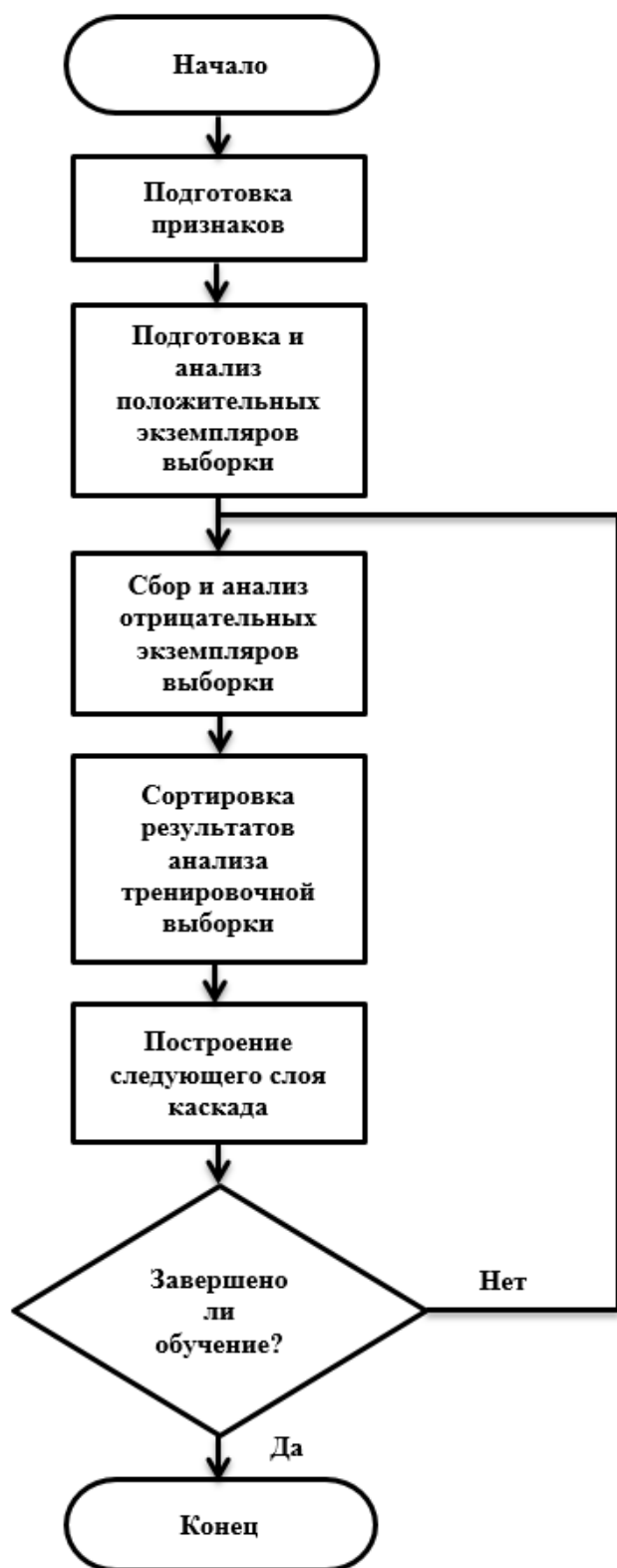


Рис.1. Схема процесса обучения в методе Виолы-Джонса

выборки, распределяя при этом нагрузку поровну на весь период обучения.

Четвертая фаза. На четвертой фазе алгоритма обучения для каждого отдельного вида признака Хаара производится объединение и сортировка по возрастанию первичных результатов обработки вместе положительных и отрицательных тренировочных экземпляров выборки.

Пятая фаза. Наконец, на пятой фазе работы алгоритма, осуществляется построение слоя каскада путем формирования его в ходе процедуры бустинга в виде линейной комбинации лучших из обученных на текущем наборе положительных и отрицательных экземпляров тренировочной выборки слабых классификаторов.

Выполненный анализ показал, что реализация параллельной версии процесса обучения классификатора принципиально возможна и целесообразна для второй и третьей фаз алгоритма, а также для пятой фазы алгоритма.

Создание параллельной версии для второй и третьей фаз алгоритма обучения. Процесс получения первичных результатов обработки признаками экземпляров обучающей выборки, как положительных, так и отрицательных, хорошо поддается распараллеливанию ввиду того, что анализ отдельным признаком отдельного экземпляра выборки производится независимо от остальных.

Из-за большого объема данных и ограниченных ресурсов оперативной памяти, этот процесс было решено разбить на несколько этапов с постепенным сохранением результатов обработки на диск с целью последующего их использования во время обучения (рис. 2). При этом в каждый файл для определенного

числа признаков из общего набора (это число определяются доступным объемом оперативной памяти) записываются результаты первичного анализа ими всех имеющихся экземпляров тренировочной выборки.

Замеры времени работы алгоритмов (табл. 1) показали, что параллельная версия процесса первичного анализа тренировочной выборки полным набором признаков Хаара по скорости работы превосходит последовательную версию более чем в сто раз.

Создание параллельной версии для пятой фазы алгоритма обучения. Было опробовано два варианта распараллеливания процесса обучения слабых классификаторов перед добавлением лучшего из них в текущий слой каскада во время процедуры бустинга. Общая схема их работы приведена на рис. 3.

Первый включает в себя перекомпоновку результатов обработки тренировочной выборки после ее сортировки так, что если до перекомпоновки в каждом файле хранились результаты обработки всей обучающей выборки определенной группой признаков из всего набора, то после – наоборот, в каждом файле для всего набора признаков хранится лишь малая часть (определяется имеющимся количеством оперативной памяти) отсортированных результатов обработки тренировочной выборки.

Благодаря этому и ввиду того, что процесс тренировки слабого классификатора независим только относительно каждого из исходных признаков Хаара, обеспечивается возможность запуска большого числа одновременно работающих потоков видеокарты, а именно равного количеству созданных на самом первом этапе признаков.

Однако, замеры производительности по-

Таблица 1

Показатели времени работы последовательной и параллельной версий алгоритма сбора первичных результатов анализа экземпляров обучающей выборки полным набором признаков Хаара

Число обработанных полным набором признаков Хаара экземпляров тренировочной выборки	Последовательная версия	Параллельная версия
2000	3 ч 23 мин	1 мин 33 с
4000	5 ч 13 мин	3 мин 3 с
6000	8 ч 38 мин	4 мин 26 с

казали, что использование данного подхода к распараллеливанию процесса обучения слабого классификатора дает преимущество в скорости над последовательной его версией лишь около в 1.56 раз (табл. 2). Причиной этому являются накладные расходы при формировании большого числа потоков видеокарты, обрабатывающих малый объем информации. При этом следует иметь в виду, что процесс перекомпоновки, занима-

ет довольно длительное время, тогда как для последовательной версии достаточно только проведения предварительной сортировки тренировочных данных (табл. 3).

Другой рассмотренный нами возможный вариант распараллеливания предполагает использование в своей работе файлов исходного формата, содержащих весь набор отсортированных первичных результатов обработки тренировочной выборки малым чис-

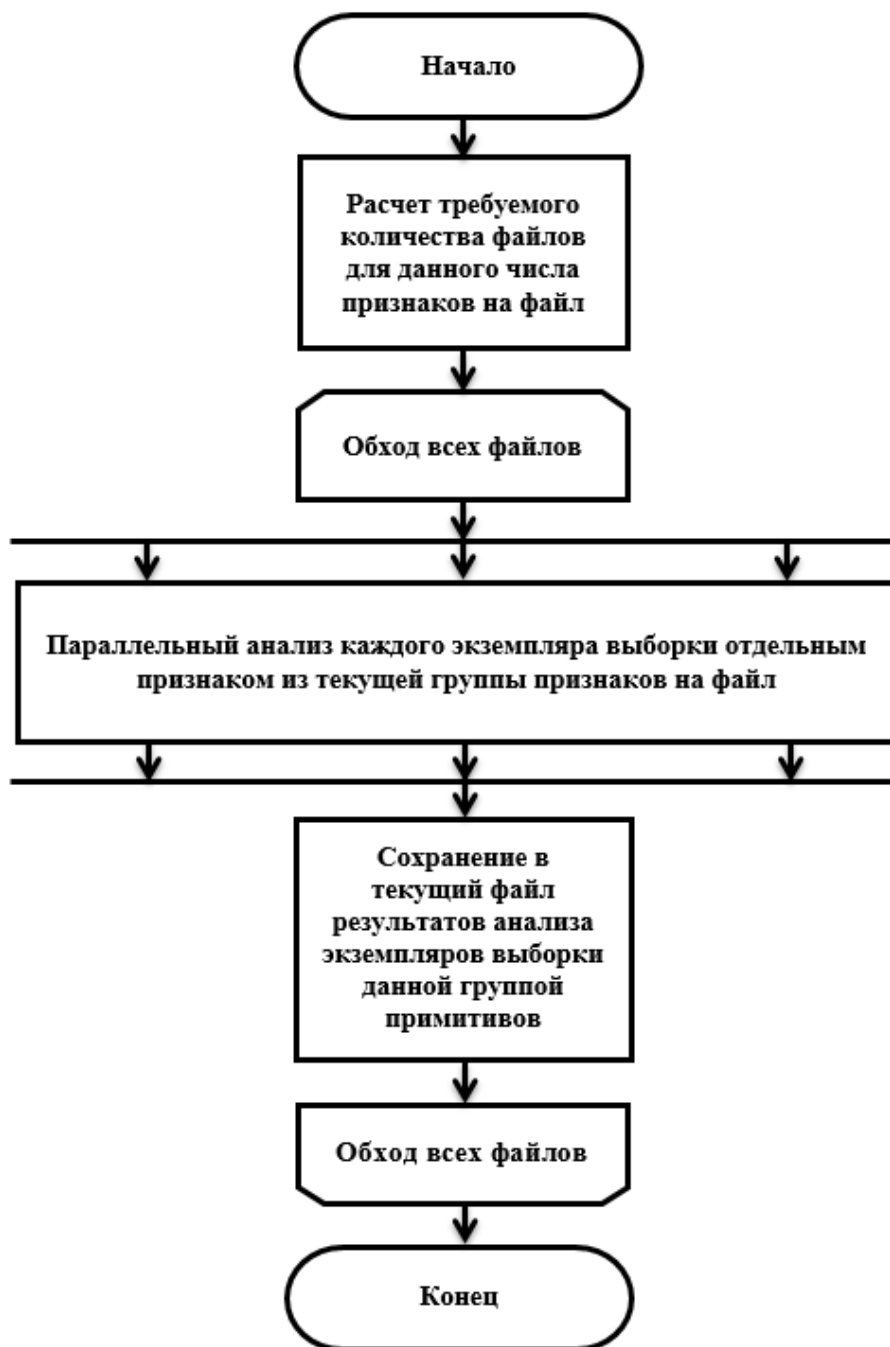


Рис. 2. Схема параллельной версии алгоритма сбора первичных результатов обработки тренировочных экземпляров выборки полным набором классификаторов

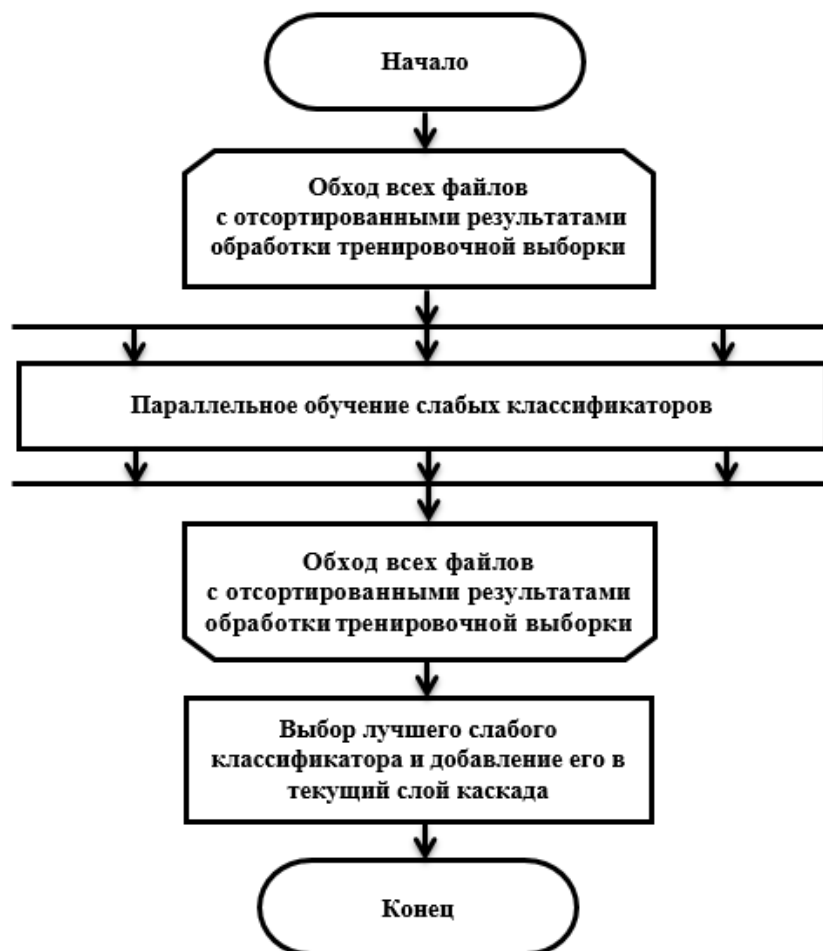


Рис.3. Схема параллельной версии процесса обучения слабых классификаторов перед добавлением лучшего из них в текущий слой каскада

Таблица 2

Показатели времени выполнения последовательной и параллельных версий алгоритма обучения слабых классификаторов и выбора лучшего из них в ходе процедуры бустинга

Число экземпляров ользуемых при обучении слабого классификатора	Последовательная версия	Параллельная версия с перекомпоновкой данных	Параллельная версия без перекомпоновки данных
2000	52 с	37 с	24 с
4000	2 мин 4 с	1 мин 14 с	1 мин 9 с
6000	2 мин 47 с	1 мин 42 с	1 мин 53 с

Таблица 3

Показатели времени, затрачиваемого на сортировку и перекомпоновку первичных результатов обработки тренировочной выборки против времени, затрачиваемого только на ее сортировку

Число экземпляров тренировочной выборки, используемых при обучении слабого классификатора	Сортировка	Сортировка и перекомпоновка данных
2000	2 мин 50 с	23 мин 13 с
4000	6 мин 5 с	1 час 31 мин
6000	8 мин 38 с	2 часа 48 мин

Таблица 4

Показатели замеров производительности последовательной и параллельной версий алгоритма анализа изображений готовым классификатором

Размер изображения в пикселях	Последовательная версия	Параллельная версия
160×120	13 мин 30 с	1.16 с
320×240	1 ч 54 мин	4.15 с
640×480	9 ч 32 мин	21.87 с

лом признаков. В этом случае количество одновременно работающих потоков невелико, и, ввиду ограничений памяти, с увеличением объема тренировочной выборки их число падает, что оказывает соответствующее влияние на скорость выполнения алгоритма (табл. 2). Однако по времени своей работы он все равно превосходит последовательный вариант и при этом не требует дополнительного времени на перекомпоновку первичных результатов обработки тренировочной выборки.

Таким образом, применение первого способа распараллеливания процесса обучения слабых классификаторов и выбора лучшего из них оправдано лишь при больших количествах слабых классификаторов внутри слоя каскада и при больших объемах тренировочной выборки.

Применение ранее обученного классификатора для распознавания лиц. На этапе использования ранее обученного классификатора производится перемещение сканирующего окна по всему изображению с постепенным увеличением масштаба анализа, и для каждого из положений сканирующего окна рассчитывается возвращаемое полученным при обучении каскадом значение в виде результата классификации. Также производится объединение нескольких близких положительных результатов детектирования в один.

При создании параллельной версии алгоритма анализа изображения нами был реализован параллелизм сканирующих окон в ходе выполнения вычислительного процесса применения каскада. Из-за ограничений по объему памяти масштабирование сканирующих окон проводится в последовательном режиме. Замеры производительности показали в среднем более чем тысячекратное превосходство параллельной версии алгоритма (табл. 4).

При тестировании производительности использовался состоящий из 85 слоев и 7468 слабых классификаторов детектор с точностью работы 81.5 % верно определенных лиц и показателем ошибочного распознавания 36 ложных срабатываний на 100 изображений тестовой выборки.

ЗАКЛЮЧЕНИЕ

Результаты проведенных экспериментов выявили заметные преимущества в скорости параллельной версии реализации алгоритма Виолы Джонса над последовательной и позволяют рекомендовать применение параллельных вычислений на графических процессорах как перспективного способа увеличения производительности метода Виолы-Джонса с учетом относительной простоты и низкой стоимости их использования.

СПИСОК ЛИТЕРАТУРЫ

1. Запрягаев С. А. Вычисление и обучение искусственных нейронных сетей прямого распространения на графическом процессоре / С. А. Запрягаев, А. А. Карпушин // Вестник Воронеж. гос. ун-та, Сер.: Системный анализ и информационные технологии. – 2011. – № 1. – С. 157–164.
2. Вахтин А. А. Реализация численного вейвлетного преобразования на графических адаптерах архитектуры NVIDIA CUDA / А. А. Вахтин, Я. А. Туровский // Вестник Воронеж. гос. ун-та, Сер.: Системный анализ и информационные технологии. – 2012. – № 1. – С. 69–72.
3. Viola P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M. Jones // Proceedings of the 2001 IEEE Conference

On Computer Vision And Pattern Recognition 2001. – 2001. – Vol. 1. – P. 511–518.

4. *Viola P.* Robust Real-Time Face Detection / P. Viola, M. Jones // International Journal of Computer Vision. – Netherlands: Kluwer Academic Publishers. – 2004. – № 57(2). – P. 137–154.

5. *Jensen O. H.* Implementing the Viola-Jones Face Detection Algorithm / O. H. Jensen. – DK-2800 Kongens Lyngby, Denmark: Technical University of Denmark Informatics and Mathematical Modelling, 2008. – 36 p.

6. *Harvey J. P.* GPU Acceleration of Object Classification Algorithms Using NVIDIA CUDA : A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering / J. P. Harvey. – Rochester, NY, 2009. – 80 p.

7. *Krpec J.* Face Detection CUDA Accelerating / J. Krpec, M. Němec // ACHI 2012: The Fifth

International Conference on Advances in Computer-Human Interactions. – 2012. – P. 155–160.

8. *Papageorgiou C. P.* A general framework for object detection / C. P. Papageorgiou., M. Oren, T. Poggio // International Conference on Computer Vision. – 1998. – P. 555–562.

9. *Freund Y.* A Short Introduction to Boosting / Y. Freund, R. Schapire // Japan: Journal of Japanese Society for Artificial Intelligence. – 1999. – N. 14(5). – P. 771–780.

10. *Акимов А. В.* Распределенная реализация алгоритма Viola-Jones с использованием графического процессора обработки данных в среде Matlab / А. В. Акимов, А. А. Сирота // Информатика: проблемы, методология, технологии. Мат. XII Международ. науч.-метод. конференции. – Воронеж : ИПЦ ВГУ, 2013. – Т. 1. – С. 36–40.

Акимов А. В. – магистр технических наук, аспирант кафедры Технологий обработки и защиты информации, факультет компьютерных наук, Воронежский государственный университет. E-mail: frinatella@gmail.ru

Akimov A. V. – Magister of Technical Sciences, Aspirant, Department of Processing Technology and Information Security, Computer Sciences Faculty, Voronezh State University. E-mail: frinatella@gmail.ru

Сирота А. А. – д. т. н., профессор, заведующий кафедрой Технологий обработки и защиты информации, факультет компьютерных наук, Воронежский государственный университет. E-mail: sir@cs.vsu.ru

Sirota A. A. – Doctor of Technical Sciences, Professor, Department of Processing Technology and Information Security, Computer Sciences Faculty, Voronezh State University. E mail: sir@cs.vsu.ru