

РЕАЛИЗАЦИЯ АЛГОРИТМА ВИОЛЫ-ДЖОНСА НА МИКРОКОНТРОЛЛЕРЕ С ОГРАНИЧЕННЫМИ РЕСУРСАМИ

Завалов Р.А.¹, Гараев Р.А.²

¹Завалов Роман Александрович – магистрант;

²Гараев Рашид Аюпович – кандидат физико-математических наук, доцент,
кафедра вычислительной техники и защиты информации,
Уфимский государственный авиационный технический университет,
г. Уфа

Аннотация: биометрическая идентификация личности по лицу, хотя и не обеспечивает абсолютно надежного результата, весьма привлекательна, главным образом, благодаря бесконтактности. Подобная технология может оказаться достаточно требовательной к вычислительным ресурсам и неэффективной при попытке реализации на маломощных вычислителях. Приводятся результаты экспериментальных исследований, направленных на получение реальных оценок затрат вычислительных ресурсов микроконтроллеров при выполнении алгоритма обнаружения лиц на изображении.

Ключевые слова: биометрическая аутентификация, локализация объектов на изображении, алгоритм Виолы-Джонса, микроконтроллеры, ограниченные вычислительные ресурсы.

Достаточно важным с практической точки зрения вариантом биометрической аутентификации является технология, основанная на распознавании человеческих лиц по фото или видеоизображениям. Хотя эта технология, в отличие от анализа отпечатков пальцев или радужной оболочки глаз, не обеспечивает должной надежности распознавания и может рассматриваться лишь как предварительная или вспомогательная, она обладает огромным преимуществом перед указанными технологиями в том, что не требует близкого контакта аппаратуры (камеры) и проверяемого (разыскиваемого) человека. В настоящее время в этом направлении достигнуты значительные результаты, что делает актуальным широкое внедрение соответствующих технологий, в том числе, в системы на базе микроконтроллеров (МК), вплоть до такой области, как Интернет вещей (IoT). Одной из отличительных черт вычислителей, применяемых в таких сферах, является сравнительно скромные вычислительные ресурсы в сравнении с современными компьютерами общего применения. Данное обстоятельство, с одной стороны, может рассматриваться как преходящий фактор, который будет постепенно преодолеваться по мере эволюционного развития аппаратных средств, с другой стороны, как повод развивать подходы, позволяющие реализовывать соответствующие алгоритмы с учетом ограничений, характерных для МК средней и малой мощности.

Системы биометрической идентификации и аутентификации на базе анализа изображений лиц могут, в принципе, основываться на таких конструктивных решениях, которые за счет взаимного расположения камеры и головы человека не требуют решения задачи детектирования или иначе локализации области, в которой располагается исследуемое лицо, однако наиболее привлекательным является подход, при котором этапу распознавания, т.е. идентификации личности предшествует автоматический поиск лица или лиц в пределах кадра. Успех операции локализации зависит от целого ряда изменяющихся факторов, в том числе, освещенности, наличия и смены теней и текстур, изменения масштаба при удалении или приближении субъекта к камере, от возможного изменения наклона головы и т.п. Тем не менее, во многих случаях с хорошей точностью операция локализации может быть выполнена на базе такого популярного метода, как алгоритм Виолы-Джонса [1].

В классическом варианте поиск объектов интереса на матрице изображения (в данном случае человеческих лиц) по алгоритму Виолы-Джонса основывается на следующих основных принципах [2]:

- представление исходного изображения в интегральном виде для ускорения процесса вычисления функций;
- применение простых прямоугольных функций, называемых признаками Хаара (Признаки Хаара организованы в каскадный классификатор);
- обучение системы распознавания объектов на основе метода «машинного обучения AdaBoost»;
- организация алгоритма распознавания объектов в виде «каскадного классификатора».

В интегральном представлении яркость каждого элемента изображения является суммой яркостей пикселей выше и левее данного включительно.

Признаки Хаара состоят из смежных прямоугольных областей. Они позиционируются на изображении, суммируются интенсивности пикселей в областях, затем вычисляется разность между суммами. Типичным для всех изображений лиц является то, что область в районе глаз темнее, чем область в районе щек, т.е. общим признаком Хаара для них является два смежных прямоугольных региона, лежащих на глазах и щеках.

Для выбора конкретных используемых функций Хаара и установления пороговых уровней, в методе Виолы-Джонса используется метод машинного обучения AdaBoost. Он комбинирует много «слабых» классификаторов с целью создания одного «сильного».

ОСОБЕННОСТИ ИСПОЛЬЗОВАННОГО ПОДХОДА

Исходя из необходимости достижения высокой скорости локализации лиц при помощи вычислителя с ограниченной производительностью реально был выбран иной подход для описания областей изображения, основанный на применении локальных бинарных шаблонов [3]. Локальные бинарные шаблоны (Local Binary Patterns, LBP) — простой оператор, используемый для классификации текстур, представляющий собой описание окрестности пикселя изображения в двоичной форме. Таким образом, каждому пикселю исходного изображения соответствует 8-битное число, каждый бит которого является результатом сравнения яркости рассматриваемого пикселя с окружающими его восемью пикселями по часовой стрелке, начиная с верхнего левого.

Классификатор LBP представляет собой набор признаков, полученных в ходе обучения набором эталонных изображений. Каждый прямоугольный признак хранится в виде координат его верхнего левого угла, а также ширины и высоты прямоугольника.

Решение использовать классификатор LBP основано на результатах сравнения производительности методов [4], показавшем незначительную разницу в точности обнаружения при значительно меньшем времени исполнения.

В процессе выполнения работы предпринята попытка реализации алгоритма Виолы-Джонса в вычислителе на базе МК с ограниченными ресурсами для локализации значительного числа лиц (более 15-и) на изображении. Предполагается, что данная система должна применяться для биометрической идентификации / аутентификации в реальном времени по изображениям, получаемым при помощи видеокамеры. В качестве фоточувствительного элемента камеры предполагалась фотоматрица с разрешением 640×480. Примером подобной фотоматрицы является изделие LUPA-300 [5], пиксели которой устроены по принципу APS (active pixel sensors), выполненная в 48-пиновом корпусе по технологии (CMOS). Матрица снабжена встроенным 10-разрядным АЦП и позволяет регистрировать до 250 кадров в секунду. Однако с точки зрения рассматриваемой работы важна не предельная скорость работы данной или аналогичных матрицы, а возможность использования режима выборочного считывания яркости пикселей в заданной области интереса (sampling and windowed Region Of Interest readout) [5]. Это дает возможность периодически считывать в ограниченную оперативную память МК лишь заданное количество строк и столбцов, постепенно сканируя всю матрицу в темпе возможностей обрабатывающего алгоритма. Поскольку большинство современных МК позволяют использовать для обмена с устройствами ввода/вывода метод прямого доступа в память, затраты процессорного времени на собственно считывание фрагментов кадра оказываются минимальным.

В качестве аппаратной базы для реализации системы использовалась отладочная плата NUCLEO-F429ZI, содержащая микроконтроллер STM32F429ZI с частотой ядра 180 МГц, модулем вычислений с плавающей запятой (FPU), 2 МБ ПЗУ и 256 КБ ОЗУ. ОЗУ содержит 64 КБ памяти Core Coupled Memory (CCM), имеющую шину, связанную только с ядром МК, что ускоряет процесс считывания данных.

Отладочная плата не была физически подключена к фотоматрице и в качестве источника графической информации при проведении экспериментов использовался файл формата BMP, записанный на SD-карту.

Хотя классический алгоритм Виолы-Джонса считается довольно быстрым инструментом для решения задачи локализации объектов на изображении, его реализация во встраиваемых системах порождает ряд проблем, требующих отдельного рассмотрения.

Во-первых, объём ОЗУ МК весьма ограничен, поэтому нет возможности загружать в неё качественную картинку целиком. Если один пиксель изображения в градациях серого занимает 1 байт памяти, то размер картинки разрешением 640×480 составит 300 килобайт, что ничтожно для современного компьютера, но абсолютно неприемлемо для среднего МК, а так как в методе Виолы-Джонса применяется интегральное представление изображения, в ОЗУ надо хранить и массив значений типа double, что увеличивает занимаемый объём памяти в 8 раз относительно исходного.

Как указывалось выше, для успешной реализации алгоритма локализации в столь стеснённых условиях было принято решение загружать в ОЗУ не всё изображение целиком, а лишь фрагмент, подлежащий текущей обработке алгоритмом, причём фрагмент этот должен быть представлен непосредственно в интегральной форме. Каждый пиксель при этом приводился к типу float, что вдвое уменьшает объём фрагмента и ускоряет процесс обработки, незначительно снижая точность. Смещение сканирующего окна на один шаг требует загрузки в память фрагмента размером, соответствующим размеру шага. Загрузка производится в область буфера, хранящую уже не обрабатываемые сканером фрагменты.

Во-вторых, потребовалось решить проблему обнаружения объекта, размер которого на входном изображении превышает размер сканирующего окна, заданного классификатором. Для применения

сканирующего окна на разных масштабах возможны два подхода: пошаговое уменьшение разрешения входного изображения или пошаговое увеличение размеров сканирующего окна. Поскольку качественное уменьшение изображения требует применения фильтров во избежание ступенчатости картинки, этот подход требует ощутимых затрат вычислительных ресурсов и частого обращения к памяти. Подход с масштабированием сканирующего окна в свою очередь требует лишь умножения значения множителя на размеры стороны окна и размеры элементов классификатора, что оказывает значительно меньшее воздействие на скорость выполнения.

Введение операций умножения чисел с плавающей запятой разумеется, даже при использовании блока FPU, оказывает влияние на производительность алгоритма. С целью оптимизаций вычислений, значения множителя в представлении чисел с плавающей запятой были преобразованы в набор

целочисленных компонентов двоично-рационального числа (dyadic) $i \cdot n / 2^m$. Значение x в масштабе

при этом вычисляется по формуле $x * i + n * (x \gg m)$, где «>>» – оператор побитового сдвига влево. Таким образом, масштабирование сканирующего окна осуществляется без применения FPU, что позволяет заменить одну операцию умножения с привлечением FPU четырьмя целочисленными операциями в пределах стандартного набора регистров.

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В реализованном алгоритме были успешно решены проблемы, описанные выше. Так, буфер имеет размер, достаточный для хранения фрагмента с оригинальной шириной и высотой, равной высоте сканирующего окна с максимальным множителем. При сканировании движение осуществляется с учётом дробного шага окна, арифметически округляемое значение которого служит индексом текущей строки буфера. При прохождении окном всего каскада классификаторов, координаты и размер окна записываются в буфер для дальнейшего вывода. По завершению обработки всего изображения в терминальный порт выводилась информация о времени выполнения алгоритма, количестве проходов через слабые классификаторы и среднее время, затраченное на проход одного слабого классификатора.

Ниже приведены результаты измерения скорости выполнения алгоритма на двух разных аппаратных платформах и с различными значениями параметров с целью оценки реальной производительности рассматриваемых платформ. Для достижения объективности оценки учитывалось только чистое время работы алгоритма, т.е. без учёта длительности операций чтения изображения и вывода отладочной информации. На рассматриваемых платформах применялся идентичный код алгоритма, адаптированный к имеющейся в каждой платформе периферии.

Использовались следующие входные параметры: размер изображения – 640×480, множитель размера окна – 2. Параметры классификатора: размер сканирующего окна – 24×24, глубина каскада – 20, количество слабых классификаторов – 100, количество признаков – 256. В качестве каскада классификаторов используется предобученный набор классификаторов и признаков `lbpcascade_frontalface.xml` из пакета OpenCV. С целью снижения объёма занимаемой памяти и упрощения обработки параметров классификатора, данные были предварительно переведены в двоичный формат.

Программа для МК реализована в среде разработки Keil uVision 5. Для взаимодействия с периферией использована библиотека STM32 Hardware Abstraction Library (HAL). Ввод и вывод данных осуществлялся по четырёхбитному интерфейсу SDIO с частотой 24 МГц в режиме обработки прерываний. Работа с файлами реализована с помощью библиотеки FatFs.

Программа компилировалась так, чтобы буфер изображения располагался в основном ОЗУ размером 192 Кб, а стек и переменные (в том числе и файл классификатора) – в области Core Coupled Memory для уменьшения латентности при регулярном обращении к памяти. Компиляция осуществлялась с флагом -O3 (полная оптимизация кода).

Время выполнения контролировалось при помощи аппаратного таймера TIM2, обладающего 32-х разрядным регистром счётчика и сконфигурированным на инкрементирование счётчика раз в микросекунду. Таймер работает независимо от ядра микроконтроллера и не влияет на выполнение алгоритма. В целях вычисления чистого времени выполнения алгоритма, таймер приостанавливался перед выполнением подгрузки данных в буфер и перед отправкой отладочных сообщений.

В ходе оценки алгоритма рассматривались пять наборов аргументов с различным шагом сканирующего окна: 2, 1,75, 1,5, 1,25, 1. Объём выполняемой работы при этом составляет соответственно 50%, 62,5%, 75%, 87,5% и 100% от полного.

Для оценки производительности выбранной МК платформы, измерения также проводились и на персональном компьютере (ПК), оснащённом четырёхъядерным микропроцессором Intel i5-3570 с фиксированной тактовой частотой 3,4 ГГц в среде ОС Windows 10 (многопоточность в программе не использовалась). Компиляция приложения осуществлялась в среде разработки Microsoft Visual Studio Community 2017 в режиме максимальной оптимизации.

Для оценки времени выполнения алгоритма на ПК применялась функция таймера с высоким разрешением `QueryPerformanceCounter`, определённая в файле `windows.h`. Как и в случае с МК, таймер

приостанавливался перед выполнением подгрузки данных в буфер и перед отправкой отладочных сообщений. Так как в обеих рассматриваемых платформах применялся идентичный алгоритм с одинаковыми входными данными, каждому набору аргументов соответствует одинаковое количество проходов слабых классификаторов, и также равны наборы окон, прошедших каскад сильных классификаторов.

На рис. 1 показан вывод результата работы алгоритма с шагом окна 1, т.е. проход сканирующего окна выполнен без исключений по всем строкам и столбцам пикселей. Из 17 полностью видимых лиц на входном изображении, алгоритм корректно выявил 15 (88%). При этом только 8 окон из 170 прошедших каскад полностью или целиком не содержат искомого объекта, что показало устойчивость к ложноположительным срабатываниям в 95,3%.

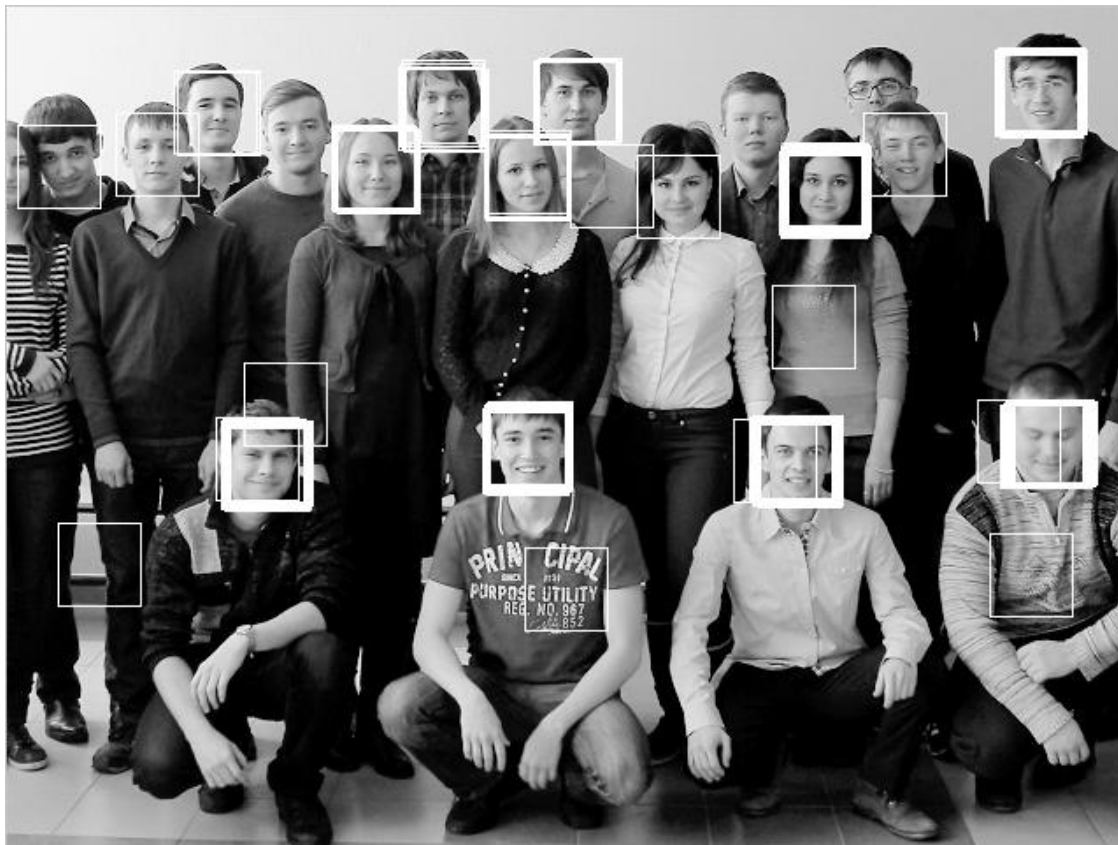


Рис. 1. Локализация лиц в результате работы алгоритма

В таблице 1 приводятся значения среднего времени, затраченного на прохождение слабого классификатора, полученные в каждом из тестов.

Таблица 1. Результаты измерения временных затрат

Шаг	STM32F429, float, мкс	STM32F429, dyadic, мкс	Intel i5-3570, dyadic, мкс
2	5,329581	4,216022	0,080935
1,75	5,329401	4,215845	0,086607
1,5	5,329089	4,215536	0,087037
1,25	5,329351	4,215799	0,082387
1	5,389260	4,212711	0,087489
Среднее	5,341336	4,215183	0,08588

Полученный результат показывает значительную эффективность замены вычислений с плавающей запятой на целочисленные – обработка осуществляется быстрее на 22%. Также заметно, что ПК выполняет описанный алгоритм в 49 раз быстрее.

Поскольку минимальным блоком фиксированной длины в алгоритме является код прохождения слабого классификатора, то оценкой производительности алгоритма $P_{алг}$ может служить количество

таких проходов за единицу времени. Также возможно оценить производительность аппаратной архитектуры $P_{ап}$ путём приведения значения производительности алгоритма к частоте вычислительного ядра. В таблице 2 приводятся вышеперечисленные показатели для рассматриваемых платформ.

Таблица 2. Результаты оценки производительности

Архитектура	$t_{с.к.}, \text{ мкс}$	$P_{алг}, \text{ с.кл./с}$	$f_{ядра}, \text{ МГц}$	$P_{ап}, \text{ с.кл./с/МГц}$
STM32F429	4,215183	237 238	180	1 318
Intel i5-3570	0,08588	11 644 154	3 400	3 425

Исходя из полученных данных, можно сделать вывод, что МК выполняет алгоритм примерно в 2,6 раза менее эффективно, чем процессор ПК. На результат оказывает влияние множество факторов программного и аппаратного характера: применяемые компилятором методы оптимизации кода, используемый конечный набор инструкций, скорость обмена данными с ОЗУ и кэшем, наличие или отсутствие кэша процессора, применение иных технологий оптимизации вычислений.

Таким образом, определено, что производительность алгоритма на МК STM32F429ZI в 49 раз ниже, чем на компьютере с процессором Intel i5-3570. Удельная же производительность МК при этом оказалась лишь в 2,6 раза ниже производительности настольного ПК. С другой стороны, систему, базирующуюся на микроконтроллере, характеризует сверхнизкое потребление электроэнергии – при выполнении вычислений оно составило всего 0,18 Вт, что является значительным преимуществом при реализации автономных решений, реализующих распознавание объектов.

Список литературы

1. Viola. P. Robust Real-Time Object Detection: tech. report / P. Viola, M. Jones. (city: Cambridge) Cambridge, 2001. 320 p.
2. Великий Я.О. Анализ принципа распознавания объектов на изображении методом Виолы–Джонса / Открытые информационные и компьютерные интегрированные технологии, 2015. № 68. С. 162-166.
3. Chang-yeon Jo. Face Detection using LBP features. / CS 229 Machine Learning Final Projects, Stanford, 2008. [Электронный ресурс]. Режим доступа: <http://cs229.stanford.edu/proj2008/Jo-FaceDetectionUsingLBP features.pdf/> (дата обращения: 06.06.2018).
4. Juliano E.C. Cruz. A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery. / Juliano E.C. Cruz, Elcio H. Shiguemori, Lamartine N.F. Guimaraes – J. Comp. Int. Sci., 2015. 6 (3):121-136.
5. LUPA300 CMOS Image Sensor. [Электронный ресурс]. Режим доступа: <https://www.onsemi.com/pub/Collateral/NOIL1SM0300A-D. PDF/> (дата обращения: 06.06.2018).