Литература

1. Шалыто А.А. Технология автоматного программирования // Труды Всероссийской научной конференции «Методы и средства обработки информации». – М.: МГУ, 2003 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/works/tech_aut_prog/, своб.
2. Клебан В.О., Шалыто А.А., Парфенов В.Г. Построение системы автоматического управления мобильным роботом на основе автоматного подхода // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53. – С. 281–285.
3. Клебан В.О., Шалыто А.А. Использование автоматного программирования для построения многоуровневых систем управления мобильными роботами // Сборник тезисов 19 Всероссийской научно-технической конференции «Экстремальная робототехника». – СПб: ЦНИИ РТК, 2008. – С. 85–87.
4. Шалыто А.А., Туккель Н.И. Switch-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. – 2001. – № 5. – С. 22–28.
5. Царев Ф.Н., Шалыто А.А. О построении автоматов с минимальным числом состояний для задачи об «Умном муравье» // Сборник докладов X международной конференции по мягким вычислениям и измерениям. – СПбГЭТУ «ЛЭТИ». – 2007. – Т. 2. – С. 88 – 91. [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf, своб.
6. Алексеев С.А. Программно-аппаратный комплекс для исследования автоматного управления мобильными роботами. – СПбГУ ИТМО. – 2010 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/papers/_alekseev_bachelor.pdf, своб.
7. Cyberbotics Ltd. Webots reference manual. 2009 [Электронный ресурс]. – Режим доступа: http://www.cyberbotics.com/cdrom/common/doc/webots/reference/reference.html, своб.
8. Тику Ш. Эффективная работа: SolidWorks. 2004. – СПб: Питер, 2005.

Алексеев Сергей Андреевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, alex.itmo@gmail.com

Клебан Виталий Олегович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, vk.developer@gmail.com

Шалыто Анатолий Абрамович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.932.72'1

ДЕТЕКТОРЫ ОСОБЕННОСТЕЙ В МЕТОДЕ ВИОЛЫ–ДЖОНСА, ПОСТРОЕННЫЕ НА ОСНОВЕ КОНЕЧНЫХ АВТОМАТОВ

П.А. Скорынин

Рассматривается модификация метода Виолы–Джонса – одного из самых эффективных методов классификации изображений. Вместо детекторов прямоугольных особенностей в предложенной модификации используются детекторы особенностей, построенные на конечных автоматах. Применение автоматов позволяет сократить число уровней в каскаде и число детекторов на каждом уровне за счет того, что детектор, управляемый автоматом, может обнаружить более сложные особенности, нежели простые прямоугольники.

Ключевые слова: классификация изображений, конечные автоматы, машинное обучение.

Введение

Задачи, связанные с компьютерным зрением, были всегда, пока существуют компьютеры. Основным шагом в создании эффективных алгоритмов в данной области стало применение в них методов машинного обучения. Одна из наиболее распространенных задач компьютерного зрения – задача о классификации. В рамках этой задачи требуется определить, принадлежит ли некоторое изображение или его часть к определенному классу. Примером такой задачи является задача о локализации лиц на изображении.

Один из примеров применения машинного обучения в задачах компьютерного зрения – метод Виолы–Джонса [1], являющийся одним из наиболее эффективных методов в своем классе. Основная его идея заключается в том, чтобы вместо одного сложного классификатора использовать каскад сильных классификаторов, построенных из слабых классификаторов, представляющих собой детекторы особенностей.

Слабые классификаторы составляют основу получаемого сильного классификатора и во многом определяют его эффективность. Заметим, что прямоугольные особенности не всегда достаточно хорошо могут описать характерные признаки искомого класса либо для этого требуется линейная комбинация достаточно большого числа таких особенностей. Таким образом, несмотря на высокую в среднем скорость работы получаемых классификаторов, в худшем случае требуется вычислять большое число особенностей. В работе Виолы и Джонса [1] для задачи детекции лиц был получен классификатор, в котором использовалось 4297 детекторов прямоугольных особенностей.

Постановка задачи и обоснование метода

Целью данной работы является модификация метода Виолы–Джонса с использованием детекторов особенностей на основе конечных автоматов. Синтезированный при помощи модифицированного метода классификатор должен распознавать признаки не хуже классификатора, синтезированного методом Виолы–Джонса.

Существует два способа повышения быстродействия методов компьютерного зрения. Первый – алгоритмическое улучшение, второй – аппаратная реализация методов. Использование детекторов особенностей на основе конечных автоматов может дать преимущество в эффективности и позволит уменьшить число слабых классификаторов в каскаде, поскольку прямоугольные особенности ограничены формой, а автомат, «двигаясь» по пикселям изображения, может распознавать не только прямоугольные элементы. При этом конечные автоматы легко реализуются на программируемой логической интегральной схеме (ПЛИС) [2]. Главное их преимущество – отсутствие сложных операций. Все, что необходимо – это сравнение и сложение. Поэтому использование конечных автоматов в детекторах особенностей гарантирует их легкую реализацию на ПЛИС с минимальными затратами ресурсов.

Описание управляющего автомата

В детекторах особенностей используется конечный автомат Мили. Выходные воздействия такого автомата генерируются в зависимости от текущего состояния и входного воздействия. В данной работе используются конечные автоматы Мили из восьми состояний.

В процессе своей работы автомат «двигается» по окну изображения размером 19×19, начиная с некоторой фиксированной позиции. Находясь в некотором пикселе изображения, автомат «видит» текущий пиксель и еще четырех соседей. По разнице между текущим пикселем и соседями принимается решение о передвижении на один пиксель в одну из четырех сторон. Кроме того, на каждом шаге отдается голос «за» или «против» текущего окна.

Работа автомата заканчивается через некоторое число шагов. Опытным путем было выяснено, что наилучшим числом шагов для заданного размера окна является 10. При «выходе» автомата за край окна окно сразу отклоняется. Если число голосов «за» превышает пороговое значение T_v, рассматриваемое окно считается принятым. В противном случае оно также отклоняется. Оптимальный порог T_v необходимо находить в процессе обучения слабого классификатора.

Входные и выходные воздействия

Для описания входных воздействий применяется тернарная логика. Всего используется четыре входных переменных – по одной на каждого соседа текущего пикселя. Рассмотрим пример вычисления входной переменной для одного из соседних пикселей. Если значения в текущем и соседнем пикселях отличаются не более чем на пороговое значение T_d, то записываем в соответствующую переменную 0. В противном случае, если значение в текущем пикселе больше, записываем «1», иначе записываем «–1». Таким образом, получаем четыре значения для четырех соседей. Всего возможных вариантов оказывается $3^4 = 81$. На рис. 1 показан пример того, как вычисляется входное воздействие.

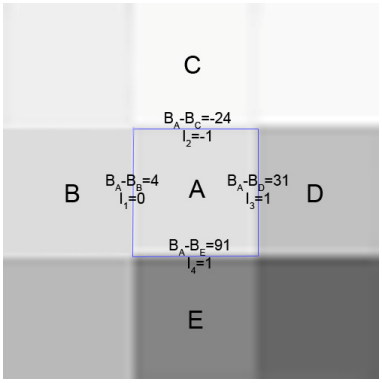


Рис. 1. Пример вычисления входного воздействия

На выходе автомат осуществляет следующие воздействия:

- направление движения (вверх, вниз, влево или вправо);
- голос «за»/«против»;
- новое состояние (одно из восьми).

Генерация автоматов

Чтобы оценить эффективность детекторов особенностей на конечных автоматах и сравнивать их между собой, определим следующую фитнесс-функцию:

$$f = k\frac{T_p}{P} + m\frac{T_N}{N},$$

где T_p – число положительных тестов, на которых автомат ответил «Да»; T_N – число отрицательных тестов, на которых автомат ответил «Нет»; P – общее число положительных тестов, N – общее число отрицательных тестов; k и m – весовые коэффициенты. В данной работе k= 3, а m=1. Выбор такой фитнесс-функции обоснован необходимостью получения низкого уровня ошибок I рода, так как в каскаде ошибки накапливаются. Поэтому вес правильно определенных положительных тестов выше, чем вес правильно определенных отрицательных тестов.

Таблица переходов для каждого из восьми состояний автомата генерируется случайно. Среди сгенерированных автоматов выбираются только те, уровень ошибок I и II рода которых является приемлемым при некоторых пороговых значениях T_d и T_v. В данной работе для этих параметров были эмпирически подобраны значения в 5% и 60% соответственно.

После этого запускается процесс последовательного улучшения автомата. Для этого применяется (1+1) эволюционная стратегия [3]. Мутацией в данной работе является изменение случайно выбранного перехода. Так как при одной мутации изменяется лишь один из 648 переходов, можно говорить о малом изменении автомата в ходе мутации.

Сильные классификаторы

Слабым классификатором называется такой классификатор, который правильно классифицирует изображения с вероятностью более 50%. Усиление слабых классификаторов – подход к решению задачи классификации путем комбинирования нескольких слабых классификаторов в один сильный. Для этой задачи может подойти любой алгоритм усиления (boosting). В данной работе для выбора особенностей, а также для обучения сильного классификатора используется алгоритм AdaBoost [4].

Фройнд и Шапир (Freund, Schapire) в работе [4] доказали, что ошибка обучения у сильного классификатора приближается к нулю в экспоненциальной зависимости от числа циклов. Кроме того, метод AdaBoost показал высокую эффективность на практике.

Вычисление классификатора из двух особенностей на основе конечных автоматов требует около 80 команд микропроцессора. Сканирование же простых шаблонов изображений или вычисление простого персептрона [5] требует в 15 раз больше операций на одно окно.

Каскад

В работе [1] описан алгоритм построения каскада классификаторов, который позволяет увеличить эффективность классификации и сократить время вычислений. Идея алгоритма заключается в том, что могут быть построены меньшие и, следовательно, более эффективные сильные классификаторы, которые отклоняют большую часть отрицательных окон при обнаружении почти всех положительных. При этом простые классификаторы будут отклонять большую часть отрицательных окон до того, как будут использованы более сложные классификаторы для достижения низкого уровня ошибок I рода.

Уровни каскада строятся из сильных классификаторов, обученных с помощью AdaBoost. Начиная с сильного классификатора из одной особенности, эффективный классификатор может быть получен путем корректировки порога сильного классификатора для минимизации ошибки I рода.

На рис. 2 представлен общий вид процесса классификации – это вырожденное дерево решений, называемое *каскадом*.

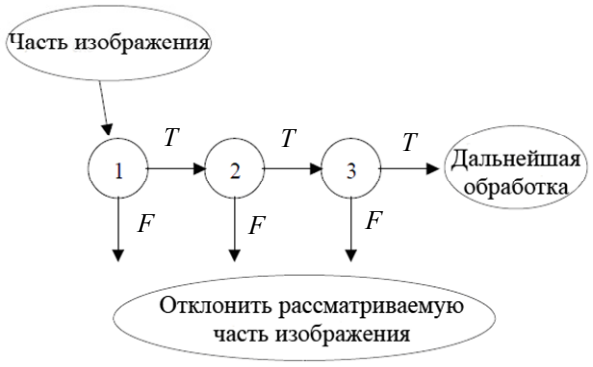


Рис. 2. Общий вид каскада сильных классификаторов

Осуществляется последовательное применение классификаторов к изображению:

- простые классификаторы отбрасывают часть отрицательных окон, при этом принимая почти все положительные окна;
- положительный отклик первого классификатора запускает вычисление второго, более сложного, классификатора, положительный отклик второго запускает третий и так далее;
- отрицательный отклик на любом этапе приводит к немедленному отклонению окна.

Обучение каскада классификаторов

В большинстве случаев классификаторы с большим числом особенностей достигают более высокого уровня распознавания и низкого уровня ошибок I рода. В то же время классификаторы с большим числом особенностей требуют больше времени для вычисления. Это может послужить базой оптимизации, при которой для оптимизации ожидаемого числа особенностей N, решающих задачу для уровня ошибки II рода F и уровня распознавания для каскада классификаторов I, варьируются следующие величины: количество уровней классификатора, количество особенностей n_i на каждом уровне, порог на каждом уровне.

Поиск оптимальных значений параметров является трудной задачей. В настоящей работе применяется простой, но демонстрирующий хорошие результаты алгоритм оптимизации.

Структура полученного классификатора

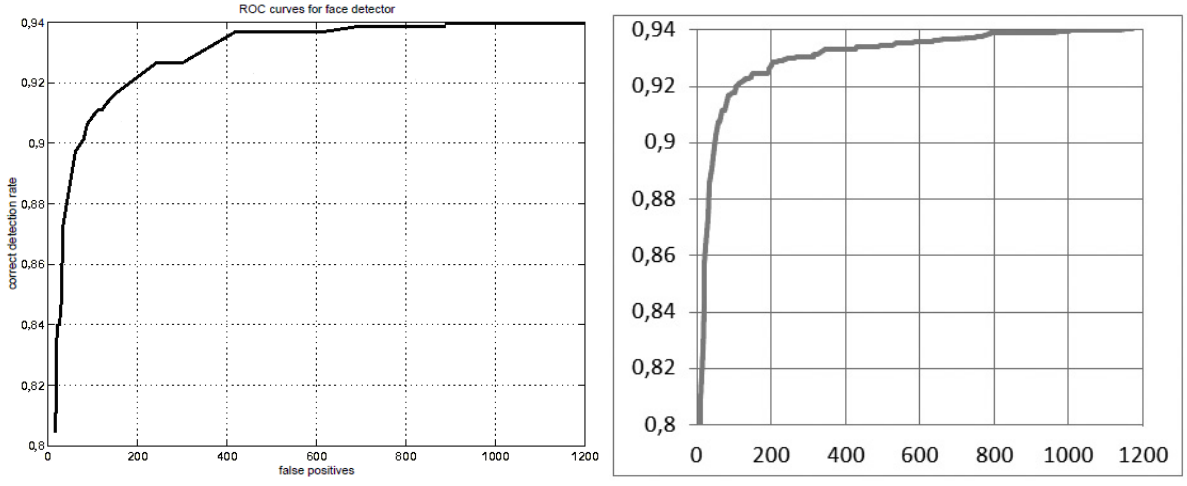
При синтезе детекторов особенностей на конечных автоматах для решения задачи детекции лиц были использованы те же базы изображений, что и в работах [6] и [7]. На первом этапе работы метода было синтезировано 250 детекторов особенностей на конечных автоматах. Уровень детекции положительных примеров у них колеблется от 96% до 99%. Ошибка II рода составляет от 23% до 55%. Получившиеся детекторы удовлетворяют требованиям для слабых классификаторов.

Для обучения каскада были выбраны следующие значения параметров: максимально допустимый уровень ложных срабатываний – 50% на слой, минимально допустимый уровень обнаружения – 99% на слой. При таких параметрах обучения максимальная общая ошибка II рода в 0,1% может быть достигнута каскадом, состоящим всего из 10 слоев.

В результате работы обучающего алгоритма был построен каскад сильных классификаторов, состоящий из 10 слоев. Первый слой состоит из одного детектора особенностей, который отсеивает более 75% отрицательных окон. Уровень распознавания первого слоя близок к 100%. Второй слой состоит уже из трех детекторов. После него отсеиваются более 90% отрицательных окон при уровне распознавания по-прежнему близкому к 100%. Следующие слои представляют собой линейные комбинации слабых классификаторов и включают в среднем по 6 детекторов. Общее число использованных в каскаде слабых классификаторов равно 52.

Тестирование получившегося классификатора и сравнение с аналогами

Для тестирования классификатора использовалась база изображений лиц CMU (Carnegie Mellon University) [7]. Та же база использовалась и для тестирования классификатора в работе Виолы и Джонса [1]. База изображений включает в себя 472 изображений лиц и 23573 изображений других типов (не лиц). При тестировании использовались те данные, которые не использовались при обучении. Та же методика использовалась Виолой и Джонсом.



а б

Рис. 3. ROC-кривая для классификатора, построенного: оригинальным методом Виолы-Джонса (а); на детекторах особенностей на конечных автоматах (б)

На рис. 3 показаны для сравнения ROC-кривые [8] для классификаторов, построенных в работе Виолы и Джонса и в данной работе.

В таблице приведены значения уровня распознавания для некоторых значений числа ложных срабатываний.

Число ложных срабатываний	10	31	50	65	78	95	110	167	422
Автоматы	81,5%	87,5%	89,9%	90,9%	91,4%	91,7%	92,0%	92,4%	93,3%
Виола-Джонс	78,3%	85,2%	88,8%	89,8%	90,1%	90,8%	91,1%	91,8%	93,7%

Таблица 1. Уровень распознавания при различном числе ложных срабатываний

Заключение

Результаты проделанной работы:

- построены слабые классификаторы на основе конечных автоматов;
- разработан и реализован метод автоматического синтеза слабых классификаторов на конечных автоматах;
- проведен анализ полученных результатов, который показал, что синтезированный классификатор не уступает по качеству классификатору, построенному методом Виолы–Джонса, но при этом обладает более высокой скоростью работы;
- реализован метод построения каскада классификаторов на основе детекторов особенностей на конечных автоматах.

В дальнейшем планируется разработка метода автоматического преобразования получаемого классификатора в язык описания схем и реализация полученного классификатора на реальном аппаратном обеспечении. Для достижения высокой надежности также возможно комбинирование аппаратной реализации метода с программной реализацией других методов: области, выделенные классификатором на автоматах, следует подвергать дополнительной проверке программными методами. При этом существенно сократится время программной обработки.

Кроме того, одним из направлений в развитии метода является усовершенствование структуры автомата в детекторах особенностей. Применение сокращенных таблиц [9] и увеличение числа состояний может помочь улучшить поведение автомата. Также существует возможность применять дополнительные фильтры над исходным изображением для того, чтобы подчеркнуть на нем некоторые особенности или, наоборот, сгладить шумы. При этом требуется, чтобы любые модификации не уменьшали скорость метода и не усложняли процесс реализации на ПЛИС.

Литература

1. Jones M. Robust Real-time Object Detection // Journal of Computer Vision. – 2004. – № 57(2). – P. 137–154.
2. Стешенко В.Б. ПЛИС фирмы Altera: элементная база, система проектирования и языки описания аппаратуры. – М.: Додэка-XXI, 2007.
3. Beyer H.-G. Evolution strategies – A comprehensive introduction // Natural Computing: an international journal. – 2002. – № 1. – P. 3–52.
4. Schapire R.E. Boosting the margin: A new explanation for the effectiveness of voting methods // Ann. Stat. – 1998. – № 26(5). – P. 1651–1686.
5. Kanade T. Neural network-based face detection // Patt. Anal. Mach. Intell. – 1998. – № 20. – P. 22–38.
6. Sung K.-K. Learning and Example Selection for Object and Pattern Detection: PhD thesis: 13.03.1996. Massachusetts Institute of Technology. 1996.
7. Heisele B. Face Detection in Still Gray images / A.I. memo AIM-1687, Artificial Intelligence Laboratory, MIT. – 2000.
8. Fawcett T. An introduction to ROC analysis // Pattern Recognition Letters. – 2006. – № 27. – P. 861–874.
9. Точилин В.Н. Метод сокращенных таблиц для генерации автоматов с большим числом входных воздействий на основе генетического программирования: Магистерская диссертация. – СПб: СПбГУ ИТМО, 2008. – 130 с.

Скорынин Павел Александрович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, pavel.skorynin@gmail.com