ОБЗОР И ТЕСТИРОВАНИЕ ДЕТЕКТОРОВ ФРОНТАЛЬНЫХ ЛИЦ

И.А. Калиновский, В.Г. Спицын
Томский политехнический университет
(национальный исследовательский университет) (ТПУ), Томск, Россия

Аннотация

Статья посвящена сравнению разработанного авторами способа обнаружения лиц, основанного на каскаде компактных свёрточных нейронных сетей, с современными детекторами фронтальных лиц. Приведены результаты тестирования 16 алгоритмов на 2 открытых наборах данных, а также замеры скорости их работы. Выводится общая оценка качества алгоритмов.

Ключевые слова: детектирование лиц, каскадные классификаторы, свёрточные нейронные сети, глубинное обучение.

Цитирование: **Калиновский, И.А.** Обзор и тестирование детекторов фронтальных лиц / И.А. Калиновский, В.Г. Спицын // Компьютерная оптика. – 2016. – Т. 40, № 1. – С. 99-111. – DOI: 10.18287/2412-6179-2016-40-1-99-111.

Введение

Детектирование лиц является первым этапом при решении задач анализа лиц, таких как идентификация личности, распознавание эмоций, пола и возраста человека. Практический интерес к этим задачам связан с востребованностью подобной функциональности в цифровых камерах, смартфонах и прочих устройствах, широким распространением автоматизированных систем обеспечения безопасности, а также с потребностью в сервисах для управления фотографиями, публикуемыми в социальных сетях.

На сегодняшний день усилия исследователей направлены на разработку алгоритмов детектирования лиц людей в естественной среде с учётом 3 степеней свободы движения головы. Сложность проблемы заключается в большом разнообразии выражений лица, условий и поз, в которых человек может быть запечатлён, а также в малой площади лица по отношению ко всей площади фотографии.

Однако можно предложить достаточно много сценариев, для которых обнаружение лиц при любом ракурсе съёмки оказывается избыточным требованием. Например, при разработке систем биометрического контроля доступа и интерактивных рекламных стендов предполагается, что человек смотрит на камеру фронтально или под небольшим углом.

В случае поиска объектов в видеопотоке, помимо высоких показателей точности и полноты, детектор должен обеспечивать работу в режиме реального времени на относительно дешевом вычислительном оборудовании. Если не брать во внимание детекторы, основанные на эмпирических моделях (например, параметрические модели распределения оттенков кожи), плохо работающие в реальных условиях, то, как правило, такие алгоритмы имеют высокую вычислительную сложность и занимают большую часть времени обработки кадра. При этом скорость их работы зависит от разрешения видеопотока, минимального размера искомых объектов, коэффициента масштабирования кадра, а также от количества объектов, присутствующих в сцене. Варьирование значений этих параметров может привести к быстрой деградации производительности, поэтому повышение вычислительной эффективности подобных алгоритмов является актуальным.

Авторами был разработан новый подход к детектированию фронтальных лиц, основанный на каскаде свёрточных нейронных сетей с очень малым числом параметров [1]. В данной работе проводится его сравнение с 15 детекторами лиц, исходные коды или демо-версии которых находятся в открытом доступе.

1. Алгоритмы детектирования лиц

В основе многих современных алгоритмов детектирования объектов лежат идеи, предложенные Виолой и Джонсом в начале 2000-х годов [2]. Разработанная ими схема построения детектора позволяет существенно сократить его вычислительную сложность с сохранением обобщающей способности и основана на следующих положениях:

а) описание локальных особенностей изображения с помощью простых функций (примитивов) Хаара, которые могут быть эффективно рассчитаны через интегральную матрицу;
б) использование алгоритма AdaBoost для построения композиции (сильного классификатора) из простых пороговых решающих правил (слабых классификаторов), использующих функции Хаара для обнаружения искомых объектов;
в) организация детектора в виде каскада из нескольких сильных классификаторов (стадий) с различной мощностью для быстрого отсева фоновых участков изображения на ранних стадиях.

Использование каскадной структуры сегодня является стандартом при построении детекторов, работающих в реальном времени. Но простых признаков, извлекаемых функциями Хаара, недостаточно для надёжного обнаружения сложных объектов в естественных условиях (неоднородный фон, недостаточное освещение, перекрытия, перспективные искажения).

Существует множество работ, посвящённых улучшению классической схемы Виолы–Джонса (табл. 1), основная суть которых заключается в расширении набора примитивов Хаара [3] или использовании иных функций для извлечения признаков, а также в модификации слабых классификаторов.

Табл. 1. Каскадные детекторы лиц

Алгоритм	Признак	Классификатор	Тип*
Lienhart R. [3]	Хаара	бустинг над «решающими пнями»	фронтальный
Jain V. [4]	Хаара		
Subburaman [5]	MCT	бустинг над специфическим решающим правилом	
Markuš N. [6]	бинарный тест	бустинг над решающими деревьями	
Li J. [7]	SURF	бустинг над логистической регрессией	фронтальный / профильный
Barr J. [8]	Хаара	бустинг над «решающими пнями»	
Yang B. [9]	совокупность каналов		
Mathias M. [10]			

* разные модели для каждого положения лица

Для работы большинства детекторов лиц [2–8] достаточно изображений, представленных в градациях серого цвета. В [9, 10] предложен альтернативный подход, в котором классификаторы обучаются на комбинации различных цветовых каналов (градации серого, RGB, HSV, LUV) с добавлением карт дескрипторов HOG и магнитуды градиента. Т.е. явно учитывается как цветовая, так и геометрическая информация об объекте. При этом в [9] к полученным картам предварительно применялась операция субдискретизации с последующим «вытягиванием» в вектор, а в [10] использовалось их интегральное представление для быстрого вычисления признаков.

Недостатком каскадного классификатора Виолы–Джонса и ему подобных является зависимость времени обработки изображения от его содержания, т.к. заранее невозможно предсказать, на какой стадии каскада фоновый участок будет отброшен. Также возникают проблемы при классификации объектов, имеющих большую внутриклассовую дисперсию. Например, при решении задачи обнаружения лиц, как правило, обучают отдельные модели для различных углов поворота головы относительно камеры ($0° \pm \psi$ – фронтальный, $45° \pm \psi$ – полуфронтальный, $90° \pm \psi$ – профильный).

Помимо собственно обнаружения лица, представляет интерес определение наклона головы и расстановка ключевых точек (положение глаз, носа, губ и др.). Решение этих дополнительных задач непосредственно на этапе детектирования позволяет существенно сократить число ложных обнаружений. Такой подход обсуждался в работах [11, 12]. В [11] рассматривался двухуровневый детектор. Первый уровень представлял стандартный каскадный детектор лиц, а второй – многозадачную свёрточную нейронную сеть, осуществляющую дополнительную проверку детекций, определение позы лица и расстановку ключевых точек. В [12] предложена каскадная модель, одновременно решающая задачи обнаружения и выравнивания лица, что позволило повысить точность классификатора при сохранении приемлемой скорости работы.

К другому классу относятся методы, в которых поиск осуществляется посредством сравнения каждого участка изображения с заданным шаблоном или с деформируемой моделью объекта, позволяющей моделировать широкий диапазон вариаций его формы. Последние достижения в этих направлениях представлены в работах [13, 14]. В [13] исследуется смесь деформируемых моделей, отличительным свойством которой является возможность обнаружения лиц, определения их позы и расстановки ключевых точек в рамках единой процедуры. В [14] предложен эффективный метод поиска путём сопоставления с шаблоном, в котором, в отличие от традиционных алгоритмов этого типа, дополнительно используются отрицательные образы для подавления ложных детекций. Для ускорения расчёта карты откликов авторы применили обобщённое преобразование Хафа. Алгоритмы [13, 14] обеспечивают высокие показатели точности и полноты классификации на стандартных тестах, но не подходят для задач реального времени, т.к. имеют очень низкую скорость выполнения (в десятки раз медленнее каскадных классификаторов).

На сегодняшний день самые передовые системы по распознаванию образов построены на базе глубоких свёрточных нейронных сетей (СНС) [15, 16]. В отличие от остальных методов машинного обучения, требующих предварительного извлечения информативных признаков для осуществления классификации, свёрточные сети решают обе эти задачи в процессе обучения, напрямую используя цветовые каналы изображений. Первые попытки построения детекторов лиц на основе СНС были сделаны ещё в середине 2000-х годов [17, 18], но они не получили распространения и значительно уступают современным каскадным детекторам по качеству и скорости работы. Однако недавно СНС нового поколения были вновь применены для решения задачи обнаружения лиц людей в естественной среде [19–21] и превзошли по качеству описанные выше алгоритмы на стандартных бенчмарках.

Авторы [19] дообучили известную сеть AlexNet на коллекции крупномасштабных фотографий AFLW [22], содержащей большое разнообразие поз и выражений лиц людей, запечатлённых в естественных условиях. При этом обучающая выборка была увеличена за счёт поворота изображений на произвольный угол. В результате авторы получили единую модель, позволяющую детектировать как фронтальные, так и профильные лица с учётом их наклона и ориентации, а также имеющую низкую вероятность ложного обнаружения. Однако СНС AlexNet содержит $61 \cdot 10^6$ параметров и, ввиду современного уровня развития вычислительных устройств, не может обрабатывать HD-видеопоток в реальном времени на оборудовании, имеющем сравнительно приемлемую стоимость, но попытки оптимизации вычисления свёрточных сетей предпринимаются [23].

В статье [20] так же, как и в наших работах [1, 21], была сделана попытка повышения производительности

глубоких свёрточных сетей, решающих задачи обнаружения объектов, путём построения детектора с каскадной структурой в соответствии с идеями Виолы и Джонса. Предложенный авторами каскад, состоящий из 6 СНС, способен детектировать лица в широком диапазоне положений головы, но по-прежнему обладает высокой вычислительной сложностью. Приведённые в статье данные о производительности детектора показывают, что он может обрабатывать в реальном времени VGA видеопоток только на мощном Nvidia GeForce GTX TITAN Black GPU. Очевидно, что при этом время поиска сильно зависит от числа лиц в сцене, т.к. на последних стадиях каскада используются очень «медленные» сети с большим количеством свёрточных ядер.

2. Каскад компактных свёрточных НС

В последние годы свёрточные нейронные сети демонстрируют выдающиеся результаты в решении задач распознавания образов. Сегодня они позволяют идентифицировать на фотографиях тысячи различных классов объектов [24], при этом точность распознавания отдельных классов, например, номеров домов [25], сравнима со средними возможностями человека. Одной из причин такого успеха является наращивание количества нейронов и связей в сетях. Анализ звука или изображений с помощью СНС, содержащих миллиарды параметров, не представляется сложной проблемой ввиду роста объёмов вычислительных ресурсов облачных платформ, а главное, с появлением технологий виртуализации GPU (например, Nvidia GRID). В задачах анализа видеопотоков, генерируемых в мегапиксельной системе видеонаблюдения предприятия, объём данных значительно возрастает. Хотя направление VSaaS (видеонаблюдение как сервис) также активно развивается, но обычно возможности анализа видео в таких сервисах ограничиваются простыми функциями (например, детектором движения). Наиболее эффективным решением этой задачи является размещение вычислительных узлов непосредственно в цифровых камерах и перенос на них функций интеллектуального анализа, что решает проблемы с масштабированием, но требует адаптации алгоритмов под ограниченные вычислительные возможности встраиваемых систем.

Задача обнаружения фронтальных лиц является относительно простой задачей классификации, т.к. может быть решена даже с использованием элементарных признаков, например, MCT [5] или бинарного теста [6]. Основная сложность заключается в уменьшении вероятности ложного обнаружения, т.к. при обучении невозможно учесть все условия (фон, освещение), в которых будет работать алгоритм в реальных системах. Поэтому, во-первых, использование сложных моделей для решения этой частной задачи не является оправданным, особенно в условиях ограниченности вычислительных ресурсов. Во-вторых, для построения качественного и производительного классификатора требуется подбор признаков, сохраняющих баланс между информативностью описания объекта и сложностью извлечения.

Свёрточные нейронные сети обладают высокой гибкостью и позволяют заранее задавать сложность моделей, меняя количество слоёв, карт и размеры свёрточных ядер. Способность к тонкой настройке признаков, извлекаемых на каждом слое, при обучении распознаванию одного класса объектов позволяет СНС достигать высоких показателей точности при поиске объектов на сильно неоднородном фоне. Необходимо учитывать, что возможности нейронной сети к обобщению предъявляемых образов снижаются с уменьшением числа параметров, вследствие чего частота ошибок первого рода (ложное обнаружение) растёт. Но данная проблема может быть решена с помощью дополнительной проверки детекций более сложными сетями (т.е. способными обеспечить большую точность классификации), аналогично структуре каскадного классификатора Виолы–Джонса.

Структура каскада

Предлагаемый нами каскадный детектор лиц состоит из 3 СНС, структура которых представлена на рис. 1. Каждая сеть решает задачу бинарной классификации фон/лицо и содержит 797 (stage 1), 1819 (stage 2) и 2923 (stage 3) параметров. В качестве функции активации нейронов используется рациональная аппроксимация гиперболического тангенса:

$$f\left(x\right)=1,7159\cdot\tanh\left(\frac{2}{3}x\right),$$

$$\tanh\left(y\right)\approx\operatorname{sgn}\left(y\right)\left(1-\frac{1}{1+\left|y\right|+y^2+1,41645\cdot y^4}\right).$$

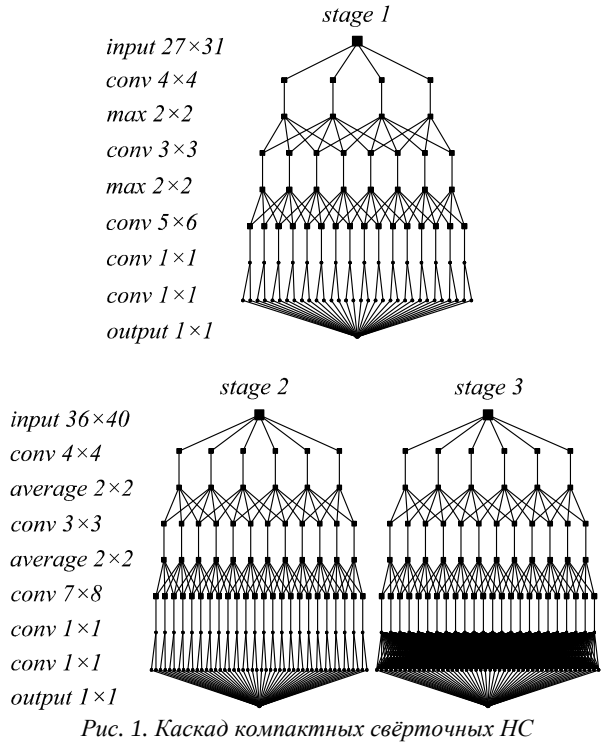

Рис. 1. Каскад компактных свёрточных НС

Нейроны подвыборочных слоёв дополнительно имеют один весовой коэффициент и смещение. Шаг операции свёртки – 1 пиксель, шаг подвыборки – 2 пикселя. В

первой и второй сетях, вместо традиционных полносвязных слоёв, используются разреженные слои (по аналогии с [17]), что на 50 % увеличивает скорость выполнения процедуры прямого распространения сигнала.

Обучающая выборка

При разработке детектора основной акцент был сделан на обработку видеоданных. В качестве обучающей выборки взяты изображения лиц из баз YouTube Faces Database [26]. Изображения фона отобраны из случайных клипов с видеохостинга YouTube в несколько этапов в процессе подготовки моделей. Также к негативам добавлены участки изображений лиц (глаза, нос и т.д.).

Тренировка моделей CHC

Для обучения свёрточных нейронных сетей авторами разработан фреймворк, поддерживающий многопоточное выполнение на CPU, а также возможность интеграции с библиотеками Intel IPP и Intel MKL для ускорения операций линейной алгебры. В общей сложности проведено порядка 1000 экспериментов по подбору оптимальной архитектуры сетей и параметров обучения. Целью экспериментов был поиск конфигурации сети с наименьшим числом параметров, способной классифицировать валидационную выборку с ошибкой не более 0,5 %. Время обучения варьировалось от нескольких часов до 2-3 дней в зависимости от размера сети (для 10^6 обучающих примеров на процессорах Intel Core i7).

Для обучения использовались изображения в градациях серого цвета. В ряде экспериментов предварительно проводилась нормализация обучающей выборки с помощью процедуры, сочетающей гамма-коррекцию и DoG-фильтр [27]. Применение такого преобразования к входному изображению позволяло обнаруживать лица даже при слабом источнике света. Но эти модели не вошли в состав финального варианта детектора из-за вычислительной сложности процедуры нормализации.

Конфигурация сетей осуществлялась по следующей схеме. Число свёрточных слоёв варьировалось от 1 до 4, число карт признаков на каждом последующем слое удваивалось. Подвыборочные слои осуществляли выборку из области 2×2 с шагом 2 пиксела по каждой оси, что уменьшало площадь карт в 4 раза. Количество карт из подвыборочного слоя, с которыми были связаны карты свёрточного слоя, варьировалось от 2 до 5. Число нейронов в предпоследнем слое сети было кратно числу карт признаков в последнем свёрточном слое.

Архитектура CHC stage 1 (рис. 1) является лучшим полученным решением, обладающим одновременно компактностью и достаточной обобщающей способностью. Нейронные сети с 2 и 3 картами на первом слое (399 и 598 параметров) не преодолели уровня ошибки в 0,5 %. Также не удалось обучить сеть с меньшим размером входа, например 20×20 (461 параметр). Использование функции ReLU вместо tanh приводило к ухудшению качества обучения. Отметим, что данная конфигурация содержит наименьшее число свёрточных ядер из всех архитектур CHC, предложенных ранее для задачи детектирования лиц [17–20]. При этом для сети stage 1 вероятность отклонения окна, содержащего фон, в 100 раз выше по сравнению с каскадом, описанным в [20].

Все эксперименты проводились преимущественно с сетями с малым числом параметров, поэтому в качестве основного алгоритма обучения использовался метод Левенберга–Марквардта, имеющий более быструю сходимость (но высокую вычислительную стоимость итерации) в сравнении с распространёнными методами решения подобных оптимизационных задач: SGD, CG, L-BFGS [28]. Результаты обучения сетей, вошедших в состав детектора, приведены в табл. 2.

Тестовый набор данных, используемый для оценки обобщающей способности различных моделей, состоял из несколько видеоклипов. В табл. 3 представлено сравнение качества каждой стадии каскада и их совместной работы на тестовых данных.

Табл. 2. Уровни ошибок CHC, составляющих детектор

Набор данных	Количество изображений, тыс.	Ошибка классификации, %		
		stage 1	stage 2	stage 3
train	лица – 433 фон – 585	0,142	0,059	0,047
validation	лица – 239 фон – 233	0,484	0,481	0,353

Табл. 3. Оценка стадий каскада на тестовых данных

Метрика	stage 1	stage 2	stage 3	cascade
recall	0,891	0,931	0,945	0,844
precision	0,179	0,219	0,104	0,990

3. Протокол тестирования

В этом разделе приводятся характеристики 15 современных алгоритмов детектирования фронтальных лиц, с которыми сравнивался предложенный авторами детектор, а также описание тестовых наборов данных и метода оценки.

Алгоритмы и библиотеки, участвовавшие в тестировании

1) Каскад компактных свёрточных нейронных сетей, описанный в работе [1]. Обозначение: CompactCNN.
2) OpenCV 3.0.0 – популярная библиотека алгоритмов машинного зрения и обработки изображений с открытым исходным кодом. Содержит платформу для разработки детекторов объектов, основанную на модифицированном алгоритме Виолы–Джонса [3]. Поставляется с 5 детекторами фронтальных лиц, четыре из которых используют признаки Хаара и один – локальные бинарные шаблоны (LBP). Обозначения: OpenCV-default, OpenCV-alt, OpenCV-alt2, OpenCV-alt-tree, OpenCV-lbp.
3) MathWorks MatLab 2013b, Computer Vision Toolbox – пакет алгоритмов машинного зрения, включающий Хаар и LBP каскадные детекторы лиц. Обозначения: Matlab-CART, Matlab-LBP.

4) Алгоритм [7] основан на SURF-дескрипторах, использует логистическую регрессию в качестве слабого классификатора. Предоставляется в виде динамической библиотеки, содержащей две модели. Обозначения: SURF-24, SURF-32.
5) Алгоритм [6] в качестве признака использует сравнение интенсивности пар пикселей (бинарный тест). Предоставляется исходный код. Обозначение: PICO.
6) Алгоритм [29] – LBP-каскад, обученный с использованием средств OpenCV на базе изображений AFLW [22]. Предоставляется в формате модели для детектора объектов OpenCV. Обозначение: OpenCV-Köstinger.
7) Алгоритм [30] – Хаар-каскад, обученный с использованием средств OpenCV. Предоставляется в формате модели для детектора объектов OpenCV. Обозначение: OpenCV-Pham.
8) Алгоритм [13] – основан на смеси деформируемых моделей, включает модели для профильного положения лица. Предоставляется исходный код на языке Matlab, а также два обученных детектора. Обозначения: FDPL-small, FDPL-large.
9) Алгоритм [31] – каскад из машин опорных векторов (SVM). Предоставляется в виде динамической библиотеки. Обозначение: FDLIB.

В табл. 4 приведены некоторые характеристики перечисленных выше алгоритмов.

Табл. 4. Сводная таблица характеристик детекторов лиц

Алгоритм	Число стадий	Размер входа	Шаг поиска, пикс.
CompactCNN (our)	3	27×31	4
OpenCV-default	25	24×24	1
OpenCV-alt	22	20×20	1
OpenCV-alt2	20	20×20	1
OpenCV-alt-tree	47	20×20	1
OpenCV-lbp	20	24×24	1
Matlab-CART	-	20×20	-
Matlab-LBP	-	24×24	-
SURF-24	5	24×24	переменный
SURF-32	5	32×32	переменный
PICO	24	24×24	0,1· *minSize**
OpenCV-Köstinger	24	24×24	1
OpenCV-Pham	31	20×20	1
FDPL-small	нет	80×80	-
FDPL-large	нет	150×150	-
FDLIB	-	19×19	-

* *minSize* – минимальный размер искомых объектов

Тестовые наборы данных

Существует достаточно много наборов данных, специально разработанных для оценки детекторов лиц, например, FDDB [32], AFW [13], MALF [33], IJB-A [34] и др. Для некоторых наборов (FDDB, MALF, IJB-A) предоставляется стандартизированный алгоритм оценки. Тестирование детекторов было проведено на наиболее популярных и одновременно сложных бенчмарках FDDB и AFW, позволяющих оценить работоспособность алгоритмов для задачи писка лиц людей в естественной среде.

Face Detection Data Set and Benchmark (FDDB) – коллекция из 2845 фотографий (не более 0,25 Мп). Содержит аннотации для 5171 лица с размерами от 20×20 пикселей. Для оценки алгоритмов применяется перекрёстная проверка на 10 подмножествах изображений с последующим усреднением результатов.

Annotated Faces in the Wild (AFW) содержит 205 крупномасштабных фотографий (0,5–5 Мп) и аннотации для 468 лиц. Данный бенчмарк был разработан относительно недавно и в основном используется для оценки мультивидовых детекторов.

Качество бинарного классификатора можно оценить с помощью ROC и PR (Precision-Recall) кривых [35]. В данной работе применяются PR-кривые, отражающие зависимость точности алгоритма (precision = TP/(TP+FP)) от его полноты (recall = TP/(TP+FN)) при варьировании порога решающего правила. Однако на практике, особенно при обработке видеоданных, подстройка порога с целью получения наилучшего соотношения между полнотой и точностью классификации в конкретных условиях работы детектора является сложной задачей. Поэтому часто используется иной подход (см. OpenCV Object detection framework: http://docs.opencv.org/3.0-beta/index.html), в котором область изображения классифицируется как содержащая объект, если число соседних детекций (*minNeighbors*) внутри этой области превышает заданный порог (рис. 2). Такой способ обеспечивает жёсткое регулирование соотношения полноты и точности, при этом удобен в настройке ввиду дискретности параметра.

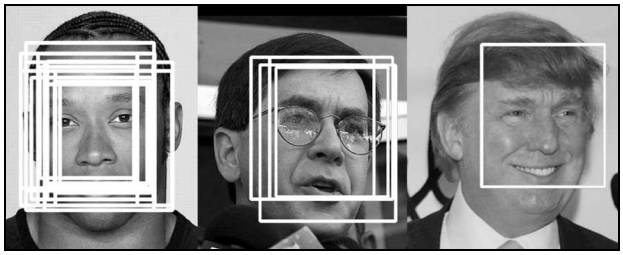

Рис. 2. Кластер детекций, формируемый классификатором при многомасштабном анализе изображения

При оценке качества алгоритма обнаружения объектов возникают трудности, связанные с неоднозначностью сопоставления областей локализации, найденных детектором (детекция) и указанных экспертом (аннотация). В настоящий момент широко распространён критерий оценки, предложенный для конкурса PASCAL Visual Object Classes [36]:

$$\sigma = \frac{S(A \cap D)}{S(A \cup D)}, \quad \delta(\sigma) = \begin{cases} \sigma \geq 0{,}5, & 1 \\ \textit{иначе}, & 0 \end{cases},$$

где σ – коэффициент перекрытия областей, S – площадь, A, D – аннотированная и найденная детектором область локализации объекта соответственно, δ –

оценка истинности или ложности детекции (каждой аннотации может соответствовать только одна детекция, остальные считаются ложными).

В случае обнаружения лиц проблема оценки дополнительно осложняется рядом факторов (рис. 3):

а) трудно указать точную границу лица, особенно для нефронтальных поз головы;
б) как правило, детекторы осуществляют классификацию прямоугольных областей, что не соответствует овальной форме лица;
в) каждый детектор определяет разную область локализации, зависящую от обучающей выборки и алгоритма объединения кластера детекций.

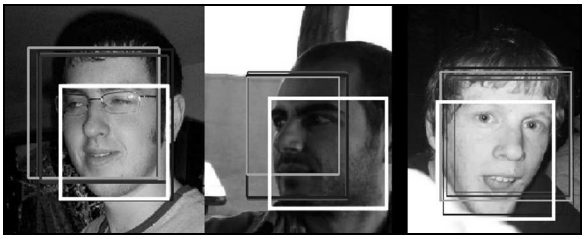

Рис. 3. Аннотации лиц из набора AFW (белая рамка) и области локализации, найденные разными детекторами

Всё это может приводить к ошибкам при сопоставлении детекции с аннотацией, когда оценка их взаимного расположения σ оказывается меньше 0,5, но визуально лицо содержится в локализованной алгоритмом области [19]. Как правило, эта проблема решается путём расширения границ детекций [20]. В данной работе для корректной оценки алгоритмов производится расширение и сужение границ аннотации по правилам, описанным в алгоритме 1 (рис. 4). Это позволяет использовать единую процедуру оценки для всех детекторов лиц на разных наборах данных.

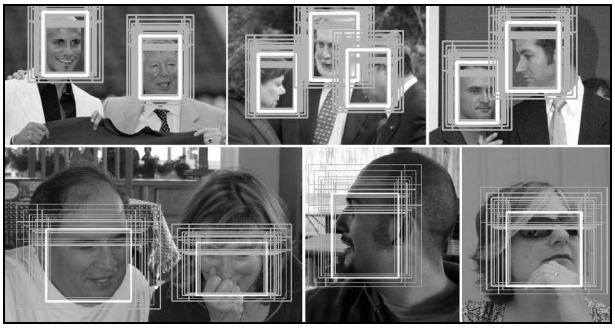

Рис. 4. Положения аннотаций FDDB (вверху) и AFW (внизу), генерируемые алгоритмом 1

Алгоритм 1. Оценка детекций

Input: X_a, Y_a, W_a, H_a, X_d, Y_d, W_d, H_d – координаты верхнего левого угла, ширина и высота области аннотации и детекции соответственно
Output: bool – истинность или ложность детекции

```
for j = -1 to 1 do
    for i = -1 to 1 do
        for s = -2 to 2 do
            if s < 0
                scale = 1,1^-s
            else
                scale = 0,95^s
            end if
            X1 = Xa – 0,5·(scale – 1)·Wa
            Y1 = Ya – 0,5·(scale – 1)·Ha
            X2 = X1 + scale·Wa
            Y2 = Y1 + scale·Ha
            if i < 0
                X1 = X1 + i·0,2·(X2 – X1)
            else
                X2 = X2 + i·0,2·(X2 – X1)
            end if
            Y1 = Y1 + j·0,2·(Y2 – Y1)
            X3 = Xb
            Y3 = Yb
            X4 = X3 + Wb
            Y4 = Y3 + Hb
            if ¬ (X1 ≥ X4 || X2 ≤ X3 || Y1 ≥ Y4 || Y2 ≤ Y3)
                if σ(X1, …, X4, Y1, …, Y4) ≥ 0,5
                    return true
                end if
            end if
        end for
    end for
end for
return false
```

Модификация аннотаций базы FDDB

Аннотации в базе FDDB представлены эллипсами, что позволяет точнее описать границу лиц по сравнению с прямоугольной областью. Однако в силу указанных выше причин сопоставление аннотаций эллиптической формы с прямоугольными детекциями приводит к ошибочной оценке последних (рис. 5).

Рис. 5. Результаты оценки CompactCNN на бенчмарке FDDB. Белым цветом обозначены детекции, признанные истинными, чёрным – признанные ложными, овалы – аннотации, числа – значения σ

В связи с этим для возможности использования предлагаемого способа оценки (алгоритм 1) эллипсы были заменены на ограничивающие их прямоугольники следующим образом:

$$W = 2\sqrt{a^2 + \left(b^2 - a^2\right)\cdot \sin^2(\omega)},$$

$$H = 2\sqrt{a^2 + \left(b^2 - a^2\right)\cdot \cos^2(\omega)},$$

$$X = x - 0{,}5\,W, \quad Y = y - 0{,}5\,H,$$

где X, Y, W, H – координаты верхнего левого угла, ширина и высота прямоугольника соответственно, a, b, ω, x, y – большая и малая полуоси, угол поворота и координаты центра эллипса соответственно.

Тестовые задачи и параметры алгоритмов

Были рассмотрены 3 задачи: обнаружение мелких (от 20 × 20 пикс.), средних (от 40 × 40 пикс.) и крупных (от 80 × 80 пикс.) лиц. Для каждого детектора рассчитывались PR-кривые, зависящие от параметра *minNeighbors*. В общей сложности алгоритмы были протестированы с 9 наборами значений параметров:

1) *minNeighbors* – {1, 2, 3} (при значении 1 превалирует полнота обнаружения, при значении 3 – точность);

2) минимальный размер искомых объектов (*minSize*) и связанный с ним масштабный коэффициент (*scaleFactor*), используемый при построении пирамиды изображений – {(20; 1,05); (40; 1,1), (80; 1,1)}.

Примечание 1. При проведении тестирования вокруг каждого изображения была добавлена чёрная рамка шириной 50 пикселей.

Примечание 2. Для детекторов Matlab, PICO и FDLIB не предусмотрен параметр *minNeighbors* и предоставляется возможность регулировки только порога решающего правила (*threshold*). Для этих детекторов порог устанавливался следующим образом:

а) Matlab, PICO: *threshold* = 2 + *minNeighbors*;

б) FDLIB: *threshold* = 2 · *minNeighbors*.

Примечание 3. Детектор FDPL не предоставляет интерфейса для регулирования параметров *minNeighbors*, *minSize* и *scaleFactor*, поэтому тестировался только с настройками по умолчанию.

Примечание 4. Детектор FDLIB не предоставляет интерфейса для регулирования параметров *minSize* и *scaleFactor*, поэтому его тестирование проводилось только для 3 значений *threshold*.

Примечание 5. Детекторы SURF и OpenCV-Köstinger были обучены на изображениях лиц, точно обрезанных по ширине глаз и уровню глаз и подбородка. При обучении остальных моделей, очевидно, использовалась более широкая область лица, включающая лоб. В результате детекторы второй группы обнаруживают более мелкие лица при фиксированном значении *minSize*. В связи с этим с целью корректного сравнения был эмпирически подобран пропорциональный коэффициент для *minSize*, используемый при инициализации SURF и OpenCV-Köstinger:

$$minSize' = 0{,}75 \cdot minSize.$$

При этом области локализации лиц, найденные этими алгоритмами, были расширены:

$$X' = X - 0{,}2W,\ Y' = Y - 0{,}3H,\ W' = 1{,}4W,\ H' = 1{,}6H,$$

где X, Y, W, H – координаты верхнего левого угла, ширина и высота ограничивающего прямоугольника соответственно.

Примечание 6. Значения дополнительных параметров, специфичных для каждого отдельного алгоритма, приведены в табл. 5.

Табл. 5. Специфические параметры детекторов

Алгоритм	Параметр	Значение
CompactCNN	T_1, T_2	0
OpenCV	*useOptimized*	true
	useOpenCL	false
SURF	*step*	1
	fast	true
PICO	*stridefactor*	0,1

4. Результаты и сравнение

Ниже приводятся результаты тестирования 16 детекторов фронтальных лиц на коллекциях фотографий FDDB и AFW (все данные, полученные в ходе тестирования, доступны по адресу https://github.com/Bkmz21/FD-Evaluation). Сравнение всех алгоритмов выполнено при одинаковых наборах значений параметров, а использованный протокол тестирования максимально приближен к реальным условиям эксплуатации. Этим проведённые нами тесты отличаются от существующих оценок эффективности некоторых алгоритмов (SURF, PICO, OpenCV-Köstinger, FDPL), выполненных методом ROC-анализа, который отражает соотношение уровней истинных и ложных детекций, но не даёт представление о поведении алгоритмов при выборе оптимального значения порога решающего правила.

На рис. 6 представлены PR-кривые, полученные при решении задачи обнаружения лиц с размером от 40 × 40 пикселей.

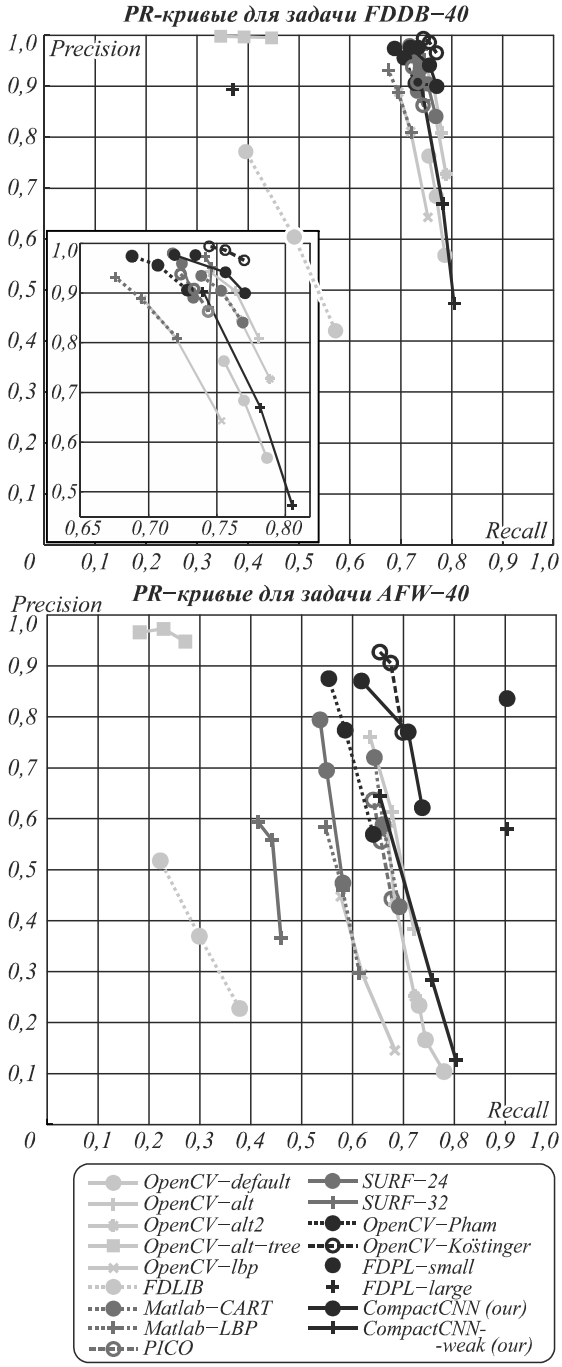

Рис. 6. Графики PR-кривых для задачи поиска лиц с размером от 40 × 40 пикселей на бенчмарках FDDB и AFW

На тесте FDDB лучшие результаты показал детектор OpenCV-Köstinger, CompactCNN имеет схожий уровень полноты, но меньшую точность. На тесте AFW с большим отрывом по показателю полноты ($R = 0{,}9$) лидирует детектор FDPL-small, но на фотографиях FDDB находит только 73 % от общего числа объектов, т.к. не рассчитан на обнаружение мелких и нечётких изображений лиц. Каскад OpenCV-Köstinger на коллекции AFW достигает более высокой точности по сравнению с CompactCNN, но при меньшей полноте обнаружения. При этом детектор CompactCNN на данных бенчмарках превосходит по точности все каскадные детекторы Виолы–Джонса, включенные в библиотеку OpenCV 3.0.0.

Оценка качества детекторов лиц

Качество бинарных классификаторов удобнее оценивать с помощью F-меры, объединяющей полноту R и точность P:

$$F = \left(\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R} \right)^{-1}, \alpha \in [0,1],$$

где α – весовой коэффициент.

В качестве единой оценки детектора на каждой тестовой задаче принимается среднее значение F-мер, рассчитанных вдоль PR-кривой (табл. 6). При этом для всех точек PR-кривой задаётся индивидуальный уровень α (табл. 7), т.к. для разных сценариев использования детекторов лиц точность и полнота не всегда равноценны (например, в биометрических системах точность имеет больший приоритет).

Табл. 6. Оценка детекторов лиц на тестовых задачах

	FDDB-20	FDDB-40	FDDB-80	AFW-20	AFW-40	AFW-80	$\overline{F}$
Compact CNN (our)	0,842	0,848	0,826	0,674	0,751	0,785	0,788
OpenCV-default	0,630	0,739	0,761	0,143	0,294	0,494	0,510
OpenCV-alt	0,806	0,837	0,803	0,462	0,662	0,754	0,721
OpenCV-alt2	0,751	0,816	0,798	0,319	0,552	0,712	0,658
OpenCV-alt-tree	0,665	0,597	0,531	0,475	0,402	0,398	0,511
OpenCV-lbp	0,697	0,778	0,763	0,225	0,420	0,578	0,577
Matlab-CART	0,795	0,831	0,794	0,441	0,647	0,752	0,710
Matlab-LBP	0,756	0,794	0,756	0,327	0,532	0,634	0,633
SURF-24	0,751	0,833	0,814	0,324	0,631	0,671	0,671
SURF-32	0,738	0,840	0,824	0,269	0,492	0,548	0,619
PICO	0,810	0,821	0,798	0,514	0,617	0,686	0,708
OpenCV-Koestinger	**0,873**	**0,863**	0,814	0,723	0,780	0,797	0,808
OpenCV-Pham	0,826	0,823	0,774	0,586	0,692	0,735	0,739
FDPL-small	0,842	0,842	**0,842**	**0,869**	**0,869**	**0,869**	**0,856**
FDPL-large	0,547	0,547	0,547	0,715	0,715	0,715	0,631
FDLIB	0,574	0,574	0,574	0,358	0,358	0,358	0,466

Табл. 7. Значения параметра α при расчёте F-меры

min Neighbors	*α*	Пояснение
1	0,2	приоритет получают детекторы с большим уровнем полноты
2	0,5	точность и полнота равноценны
3	0,8	приоритет получают детекторы с большим уровнем точности

В табл. 8 по каждой задаче приведено распределение мест среди детекторов в порядке уменьшения их F-меры, а также относительное отставание от победителя (процентная разница) – ΔF. В качестве общей относительной оценки качества алгоритма предлагается среднее значение ΔF по всем тестам. Оценка $\overline{\Delta F}$ показывает, как соотносится уровень F-меры алгоритма с наилучшим решением. Результат ранжирования детекторов лиц согласно указанному критерию приведён на рис. 7. Лучшим детектором является FDPL-small, но не на всех тестах, что отражается в оценке $\overline{\Delta F} = -1\%$. Он способен находить как фронтальные, так и профильные лица, что обеспечило ему преимущество по критерию полноты на бенчмарке AFW. За ним следует OpenCV-Köstinger с отставанием на 5,44 % процентных пункта. CompactCNN детектор занимает третье место с отставанием от FDPL-small на 7,81 % и от OpenCV-Köstinger на 2,37 % пункта.

Также была проведена оценка классификаторов с помощью теста Фридмана, который показал отсутствие статистически значимых различий в качестве работы детекторов CompactCNN, OpenCV-Köstinger и FDPL-small при уровне значимости 0,05. Отметим, что алгоритмы FDPL-small и OpenCV-Köstinger были обучены на мегапиксельных фотографиях из набора AFLW, в то время как для обучения CompactCNN использовались кадры, взятые из случайных видеоклипов с YouTube. Поэтому разработанный детектор может получить преимущество в задачах поиска лиц в видеопотоке, т.к. в этом случае обрабатываемые данные в большей степени соответствуют статистическому распределению обучающей выборки.

Табл. 8. Распределение мест среди детекторов лиц

	FDDB-20	FDDB-40	FDDB-80	AFW-20	AFW-40	AFW-80	$\overline{\Delta F}$, %
Compact CNN (our)	2 (-3,55)	2 (-1,74)	2 (-1,9)	4 (-22,4)	3 (-13,6)	3 (-9,67)	-8,81
OpenCV-default	12 (-27,8)	13 (-14,4)	10 (-9,62)	16 (-83,5)	16 (-66,2)	14 (-43,2)	-40,78
OpenCV-alt	5 (-7,67)	5 (-3,01)	5 (-4,63)	8 (-46,8)	6 (-23,8)	4 (-13,2)	-16,53
OpenCV-alt2	8 (-14)	10 (-5,45)	6 (-5,23)	13 (-63,3)	10 (-36,5)	8 (-18,1)	-23,75
OpenCV-alt-tree	11 (-23,8)	14 (-30,8)	14 (-36,9)	7 (-45,3)	14 (-53,7)	15 (-54,2)	-40,81
OpenCV-lbp	10 (-20,2)	12 (-9,85)	9 (-9,38)	15 (-74,1)	13 (-51,7)	12 (-33,5)	-33,11
Matlab-CART	6 (-8,94)	7 (-3,71)	7 (-5,7)	9 (-49,3)	7 (-25,5)	5 (-13,5)	-17,77
Matlab-LBP	7 (-13,4)	11 (-8)	11 (-10,2)	11 (-62,4)	11 (-38,8)	11 (-27)	-26,63
SURF-24	8 (-14)	6 (-3,48)	4 (-3,33)	12 (-62,7)	8 (-27,4)	10 (-22,8)	-22,28
SURF-32	9 (-15,5)	4 (-2,67)	3 (-2,14)	14 (-69)	12 (-43,4)	13 (-36,9)	-28,27
PICO	4 (-7,22)	9 (-4,87)	6 (-5,23)	6 (-40,9)	9 (-29)	9 (-21,1)	-18,04
OpenCV-Koestinger	**1 (0)**	**1 (0)**	4 (-3,33)	2 (-16,8)	2 (-10,2)	2 (-8,29)	-6,44
OpenCV-Pham	3 (-5,38)	8 (-4,63)	8 (-8,08)	5 (-32,6)	5 (-20,4)	6 (-15,4)	-14,41
FDPL-small	2 (-3,55)	3 (-2,43)	**1 (0)**	**1 (0)**	**1 (0)**	**1 (0)**	**-1,00**
FDPL-large	14 (-37,3)	16 (-36,6)	13 (-35)	3 (-17,7)	4 (-17,7)	7 (-17,7)	-27,03
FDLIB	13 (-34,3)	15 (-33,5)	12 (-31,8)	10 (-58,8)	15 (-58,8)	16 (-58,8)	-46,00

* в скобках указана процентная разница ΔF с алгоритмом, имеющим максимальную F-меру по каждой задаче

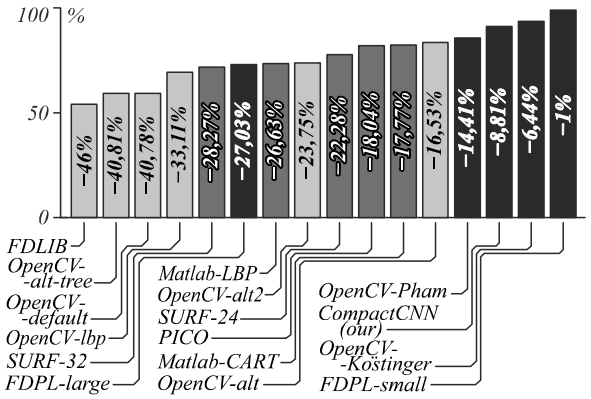

Рис. 7. Ранжирование детекторов по оценке $\overline{\Delta F}$

Оценка производительности детекторов лиц

Вычислительная эффективность является важной характеристикой детекторов объектов, особенно при решении задач реального времени и обработки больших объёмов данных. В таких условиях детекторы с каскадной структурой обладают преимуществом, т.к. способны быстро отсеять большую часть изображения, не содержащую искомых объектов. Например, первой стадии каскада Хаара OpenCV-alt, состоящей из 3 слабых классификаторов, для осуществления классификации требуется около 26 арифметических операций (при использовании интегральной матрицы и прямоугольных признаков). Алгоритмическая сложность CHC значительно выше из-за необходимости вычисления выходных сигналов нескольких тысяч нейронов. К примеру, для получения карты откликов CHC stage 1 (3905 нейронов, рис. 1) на изображении с разрешением 1280×720 пикселей требуется $\approx 340 \cdot 10^6$ операций (с шагом окна в 4 пикселя), в то время как каскаду OpenCV-alt необходимо только $\approx 23 \cdot 10^6$ (с шагом окна в 1 пиксель).

Помимо сложности вычисления, большое влияние на производительность оказывает точность сильных классификаторов, а также их количество (длина каскада). От длины каскада зависит производительность детектора в худшем случае, когда изображение полностью заполнено искомыми объектами. Общее число слабых классификаторов в 22-стадийном детекторе OpenCV-alt равно 2135. Следовательно, в худшем случае ему потребуется $\approx 16 \cdot 10^9$ операций, при этом для детектора CompactCNN только $\approx 1{,}5 \cdot 10^9$.

Распределение вероятности ошибки первого рода по стадиям каскада определяет скорость работы детектора в обычных условиях работы (т.е. уровень между обработкой абсолютно чёрного изображения и полностью заполненного лицами). Каскады Хаара из OpenCV отсеивают 60-70 % положений скользящего окна на первой стадии, используя от 3 до 9 признаков. Детектор [7], основанный на более сложных признаках – SURF-дескрипторах, отсеивает 95 % окон, но производит при этом больший объём вычислений. CompactCNN, извлекающий высокоуровневые признаки, оптимизированные для обнаружения лиц, способен уже на первой стадии отклонить более 99,99 % всех положений окна при нулевом пороге решающего правила [1]. Таким образом, скорость его работы слабо зависит от структуры фона на фотографии.

Микроархитектура современных процессоров является суперскалярной, содержит блоки переупорядочения инструкций и переименования регистров, а исполнительные устройства способны выполнять большое количество разнообразных скалярных и векторных операций над данными. В связи с этим производительность алгоритма зависит не только от вычислительной сложности, но также от его структуры (наличие ветвлений, порядок обращения к памяти и др.), типа данных и используемых инструкций, способности к эффективной векторизации и распараллеливанию.

Базовыми элементами большинства детекторов, построенных с помощью процедуры бустинга, явля-

ются решающие деревья (см. табл. 1). Поиск объектов осуществляется с помощью скользящего окна, в котором в разных фиксированных точках вычисляются признаки, что не является оптимальным с точки зрения загрузки данных в кэш-память. Процедура обхода дерева плохо поддаётся векторизации из-за зависимости результата от последовательности переходов. В связи с этим достаточно сложно построить эффективный алгоритм вычисления детекторов такого типа, использующий преимущества SIMD-расширений современных CPU и массивно-параллельных архитектур GPU. Решению этой задачи посвящено множество работ [37, 38].

С точки зрения структуры вычислений СНС значительно эффективнее алгоритма Виолы–Джонса и его модификаций:

1) карта откликов сети может быть просто рассчитана без использования скользящего окна путём применения ко всему изображению линейной операции корреляции с наборами фильтров, операции подвыборки и нелинейного попиксельного преобразования [39]. Благодаря этому свойству чтение данных осуществляется непрерывными блоками памяти, что позволяет эффективно использовать кэш процессора;
2) на каждую операцию чтения приходится несколько десятков операций с данными (без ветвлений внутри циклов), что покрывает латентность кэш-памяти;
3) нейронная сеть по своей сути является массивно-параллельным алгоритмом, поэтому очень просто поддаётся векторизации и распараллеливанию.

Перечисленные особенности СНС особенно важны при переносе вычислений на GPU. Графические процессоры позволяют в полной мере раскрыть все преимущества естественного параллелизма нейронных сетей, а также имеют аппаратную поддержку выборки из двумерных массивов.

Недостатком СНС является необходимость хранения большого объёма промежуточных вычислений (карт признаков), что увеличивает объём требуемой памяти, а также снижает эффективность многопоточного выполнения при недостаточном объёме кэш-памяти процессора.

Детектор CompactCNN был реализован с помощью 3 технологий: SIMD-расширение процессоров семейства ×86 (для каждого из 3 наборов векторных инструкций: SSE, AVX, AVX2 – поддерживаемых в микроархитектурах Intel Sandy/Ivy Bridge и Haswell/Broadwell), Nvidia CUDA, OpenCL. Вычисления осуществляются с одинарной точностью, а характеристика precision-recall идентична для всех реализаций.

На рис. 8 приведено сравнение производительности детекторов лиц, участвовавших в тестировании. Замеры проводились для однопоточного режима выполнения на процессоре Intel Core i7-3610QM (3,1 ГГц) при обработке первого подмножества изображений из коллекции FDDB (*minSize* – 40 × 40, *scaleFactor* – 1,1, *minNeighbors* – 2, без добавления рамки). Исполняемые файлы и библиотеки всех детекторов, кроме SURF и FDLIB, являются 64-битными. Использовался C++ компилятор, входящий в состав IDE Microsoft Visual Studio Community 2013, ОС Microsoft Windows 8.1 (64-bit).

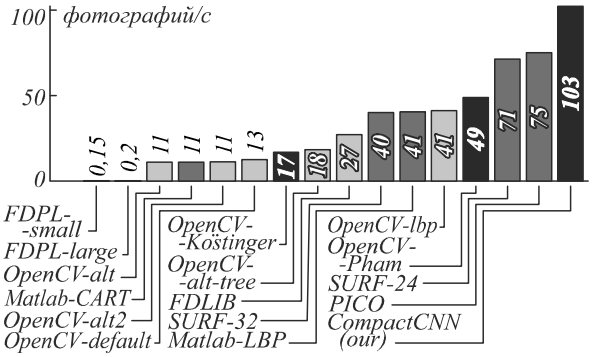

Рис. 8. Ранжирование детекторов по скорости работы для задачи FDDB-40

Благодаря оригинальному алгоритму вычисления свёрточных нейронных сетей и оптимизации программного кода с помощью векторных intrinsic-функций, с учётом ограниченности количества логических регистров и особенностей микроархитектур CPU Intel, CompactCNN демонстрирует рекордную скорость обработки данных (при тестировании использовалась реализация, оптимизированная с помощью AVX intrinsic-функций).

Заключение

Результаты проведённого тестирования 16 детекторов фронтальных лиц показывают, что предложенный нами детектор на основе каскада компактных свёрточных нейронных сетей занимает 3-ю позицию по уровню *F*-меры, отставая в среднем на 8,8 % от лучшего решения. При этом он занимает лидирующую позицию по скорости обработки данных, на 37 % улучшая предыдущее достижение (PICO), в 687 (FDPL-small) и в 6 (OpenCV-Köstinger) раз превосходя более качественные детекторы.

Разработанный каскадный детектор состоит всего из 3 стадий. При этом 99,99 % окон, содержащих фон, отклоняется уже на первой стадии, что существенно снижает зависимость скорости работы детектора от содержания изображения [1]. Благодаря наличию реализаций для 3 вычислительных технологий, включая OpenCL, CompactCNN может быть запущен практически на любом устройстве. При этом реализация для CPU Intel и GPU Nvidia была высоко оптимизирована с учётом особенностей микроархитектур процессоров. Каскад небольших СНС оказался очень эффективным решением для задачи детектирования фронтальных лиц и впервые позволил обрабатывать видеопоток сверхвысокого разрешения 4K даже на маломощных вычислительных устройствах [1].

Литература

1. **Kalinovskii, I.A.** Compact Convolutional Neural Network Cascade for Face Detection / I.A. Kalinovskii, V.G. Spitsyn [Электронный ресурс]. – 2015. – URL: http://arxiv.org/abs/1508.01292.pdf (дата обращения 01.02.2016).

2. **Viola, P.** Rapid object detection using a boosted cascade of simple features / P. Viola, M.J. Jones // IEEE Conference on Computer Vision and Pattern Recognition. – 2001. – Vol. 1. – P. 511-518.
3. **Lienhart, R.** An extended set of Haar-like features for rapid object detection / R. Lienhart, J. Maydt // IEEE International Conference on Image Processing. – 2002. – Vol. 1. – P. 900-903.
4. **Jain, V.** Online domain adaptation of a pre-trained cascade of classifiers / V. Jain, E. Learned-Miller // IEEE Conference on Computer Vision and Pattern Recognition. – 2011. – P. 577-584.
5. **Subburaman, V.** Fast bounding box estimation based face detection / V. Subburaman, S. Marcel // European Conference on Computer Vision, Workshop on Face Detection. – 2010. – P. 1-14.
6. **Markuš, N.** A method for object detection based on pixel intensity comparisons organized in decision trees / N. Markuš, M. Frljak, I.S. Pandžić, J. Ahlberg, R. Forchheimer // [Электронный ресурс]. – 2013. – URL: http://arxiv.org/abs/1305.4537.pdf (дата обращения 01.02.2016).
7. **Li, J.** Learning SURF cascade for fast and accurate object detection / J. Li, Y. Zhang // IEEE Conference on Computer Vision and Pattern Recognition. – 2013. – P. 3468-3475.
8. **Barr, J.R.** The effectiveness of face detection algorithms in unconstrained crowd scenes / J.R. Barr, K.W. Bowyer, P.J. Flynn // IEEE Winter Conference on Applications of Computer Vision. – 2014. – P. 1020-1027.
9. **Yang, B.** Aggregate channel features for multi-view face detection / B. Yang, J. Yan, Z. Lei, S.Z. Li // IEEE International Joint Conference on Biometrics. – 2014. – P. 1-8.
10. **Mathias, M.** Face detection without bells and whistles / M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool // European Conference on Computer Vision. – 2014. – P. 720-735.
11. **Zhang, C.** Improving multiview face detection with multi-task deep convolutional neural networks / C. Zhang, Z. Zhang // IEEE Winter Conference on Applications of Computer Vision. – 2014. – P. 1036-1041.
12. **Chen, D.** Joint cascade face detection and alignment / D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun // European Conference on Computer Vision. – 2014. – P. 109-122.
13. **Zhu, X.** Face detection, pose estimation, and landmark localization in the wild / X. Zhu, D. Ramanan // IEEE Conference on Computer Vision and Pattern Recognition. – 2012. – P. 2879-2886.
14. **Li, H**. Efficient boosted exemplar-based face detection / H. Li, Z. Lin, J. Brandt, X. Shen, G. Hua // IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – P. 1843-1850.
15. **Zeiler, M.** Visualizing and understanding convolutional networks / M. Zeiler, R. Fergus // European Conference on Computer Vision. – 2014. – P. 818-833.
16. **Szegedy, C.** Going deeper with convolutions / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich // [Электронный ресурс]. – 2014. – URL: http://arxiv.org/abs/1409.4842.pdf (дата обращения 01.02.2016).
17. **Garcia, C.** Convolutional face finder: A neural architecture for fast and robust face detection / C. Garcia, M. Delakis // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – P. 1408-1423.
18. **Osadchy, M.** Synergistic face detection and pose estimation with energy-based models / M. Osadchy, Y. LeCun, M. Miller // Journal of Machine Learning Research. – 2007. – P. 1197-1215.
19. **Farfade, S.S.** Multi-view face detection using deep convolutional neural networks / S.S. Farfade, M. Saberian, L.-J. Li // International Conference on Multimedia Retrieval. – 2015.
20. **Li, H.** A Convolutional neural network cascade for face detection / H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua // IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – P. 5325-5334.
21. **Калиновский, И.А.** Алгоритм детектирования лиц на видео сверхвысокого разрешения / И.А. Калиновский, В.Г. Спицын // Техническое зрение в системах управления. – 2015. – С. 95-96.
22. **Köstinger, M.** Annotated Facial Landmarks in the Wild: A Large-scale, real-world database for facial landmark localization / M. Köstinger, P. Wohlhart, P.M. Roth, H. Bischof // IEEE International Conference on Computer Vision Workshops. – 2011. – P. 2144-2151.
23. **Vasilache, N.** Fast convolutional nets with fbfft: A GPU performance evaluation / N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, Y. LeCun // [Электронный ресурс]. – 2014. – URL: http://arxiv.org/abs/1412.7580.pdf (дата обращения 01.02.2016).
24. **Ioffe, S.** Batch normalization: accelerating deep network training by reducing internal covariate shift / S. Ioffe, C. Szegedy // [Электронный ресурс]. – 2015. – URL: http://arxiv.org/abs/1502.03167.pdf (дата обращения 01.02.2016)
25. **Lee, C.-Y.** Deeply-supervised nets / C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu // // [Электронный ресурс]. – 2014. – URL: http://arxiv.org/abs/1409.5185.pdf (дата обращения 01.02.2016)
26. **Wolf, L.** Face recognition in unconstrained videos with matched background similarity / L. Wolf, T. Hassner, I. Maoz // IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – P. 529-534.
27. **Калиновский, И.А.** Алгоритм обнаружения лиц на основе сверточной нейронной сети / И.А. Калиновский, В.Г. Спицын // Нейрокомпьютеры: разработка и применение. – 2013. – № 10. – С. 48-53.
28. **Le, Q.V.** On Optimization Methods for Deep Learning / Q.V. Le, A. Coates, B. Prochnow, A.Y. Ng // International Conference on Machine Learning. – 2011. – P. 265-272.
29. **Köstinger, M.** Efficient metric learning for real-world face recognition / M. Köstinger. – Graz University of Technology: PhD thesis, 2013.
30. **Pham, M.T.** Fast training and selection and Haar features using statistics in boosting-based face detection / M.T. Pham, T.J. Cham // IEEE International Conference on Computer Vision. – 2007. – P. 1-7.
31. **Kienzle, W.** Face detection: efficient and rank deficient / W. Kienzle, G. Bakir, M. Franz, B. Scholkopf // Advances in Neural Information Processing Systems. – 2005. – P. 673-680.
32. **Jain, V.** FDDB: A Benchmark for face detection in unconstrained settings / V. Jain, E. Learned-Miller // Technical Report UM-CS-2010-009. – University of Massachusetts. – 2010.
Yang, B. Fine-grained evaluation on face detection in the wild / B. Yang, J. Yan, Z. Lei, S.Z. Li // IEEE International Conference on Automatic Face and Gesture Recognition. – 2015.
33. **Klare, B.F.** Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. / B.F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, A.K. Jain // IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – P. 1931-1939.
34. **Davis, J.** The relationship between Precision-Recall and ROC curves / J. Davis, M. Goadrich // International Conference on Machine Learning. – 2006. – P. 233-240.
35. **Everingham, M.** The PASCAL visual object classes (VOC) challenge / M. Everingham, L.V. Gool, C. Williams, J. Winn, A. Zisserman // International Journal of Computer Vision. – Vol. 88(2). – 2010. – P. 303-338.
36. **Oro, D.** Real-time GPU-based face detection in HD video sequences / D. Oro, C. Fernandez, J.R. Saeta, X. Martorell, J. Hernando // IEEE International Conference Computer Vision Workshops. – 2011. – P. 530-537.

37. **Nguyen, T.** A software-based dynamic-warp scheduling approach for load-balancing the Viola–Jones face detection algorithm on GPUs / T. Nguyen, D. Hefenbrock, J. Oberg, R. Kastner, S. Baden // Journal of Parallel and Distributed Computing. – Vol. 73(5). – 2013. – P. 677-685.

38. **Sermanet, P.** OverFeat: Integrated recognition, localization and detection using convolutional networks / P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun // [Электронный ресурс]. – 2013. – URL: http://arxiv.org/abs/1312.6229.pdf (дата обращения 01.02.2016).

Сведения об авторах

Калиновский Илья Андреевич, 1990 года рождения, в 2011 году получил степень бакалавра по направлению «Прикладная математика и информатика» в Тихоокеанском государственном университете г. Хабаровска. В 2013 году получил степень магистра по направлению «Информатика и вычислительная техника» на кафедре вычислительной техники Национального исследовательского Томского политехнического университета. С 2013 г. является аспирантом этой кафедры. Область научных интересов: обработка и анализ изображений, распознавание образов, искусственные нейронные сети и методы глубинного обучения. Является автором 20 научных статей. E-mail: *kua_21@mail.ru* .

Спицын Владимир Григорьевич, 1948 года рождения, в 1970 году окончил Томский государственный университет по специальности «Радиофизика и электроника», профессор, д.т.н., профессор Национального исследовательского Томского политехнического университета. Член редколлегии журнала «Известия Томского политехнического университета». С 2012 г. является действительным членом Международной академии информатизации. Область научных интересов: нейронные сети, генетические алгоритмы, обработка изображений, распознавание образов, многократное рассеяние электромагнитных волн в случайно-неоднородных средах. Является автором 180 научных статей. E-mail: *spvg@tpu.ru* .

Поступила в редакцию 1 февраля 2016 г.
Окончательный вариант – 16 февраля 2016 г.

REVIEW AND TESTING OF FRONTAL FACE DETECTORS

I.A. Kalinovskii, V.G. Spitsyn
Tomsk Polytechnic University

Abstract

This paper presents comparison results for the proposed face detection algorithm based on a compact convolutional neural network cascade and modern frontal face detectors. Test results for 16 frontal view face detectors on two public benchmarks datasets are shown. A comparative assessment of the performance of face detection algorithms is made.

Keywords: face detection, cascade classifiers, convolutional neural networks, deep learning.
Citation: Kalinovskii IA, Spitsyn VG. Review and testing of frontal face detectors. Computer Optics 2016; 40(1): 99-111. DOI: 10.18287/2412-6179-2016-40-1-99-111.

References

[1] Kalinovskii IA, Spitsyn VG. Compact Convolutional Neural Network Cascade for Face Detection. Source: ⟨http://arxiv.org/abs/1508.01292.pdf⟩.
[2] Viola P, Jones MJ. Rapid object detection using a boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition; 2001; 1: 511-518.
[3] Lienhart R, Maydt J. An extended set of Haar-like features for rapid object detection. IEEE International Conference on Image Processing; 2002: 1: 900–903.
[4] Jain V, Learned-Miller E. Online domain adaptation of a pre-trained cascade of classifiers. IEEE Conference on Computer Vision and Pattern Recognition; 2011; 577–584.
[5] Subburaman V, Marcel S. Fast bounding box estimation based face detection. European Conference on Computer Vision, Workshop on Face Detection; 2010; 1–14.
[6] Markuš N, Frljak M, Pandžić IS, Ahlberg J, Forchheimer R. A method for object detection based on pixel intensity comparisons organized in decision trees. Source: ⟨ http://arxiv.org/abs/1305.4537.pdf⟩.
[7] Li J, Zhang Y. Learning SURF cascade for fast and accurate object detection. IEEE Conference on Computer Vision and Pattern Recognition; 2013; 3468–3475.
[8] Barr JR, Bowyer KW, Flynn PJ. The effectiveness of face detection algorithms in unconstrained crowd scenes. IEEE Winter Conference on Applications of Computer Vision; 2014; 1020–1027.
[9] Yang B, Yan J, Lei Z, Li SZ. Aggregate channel features for multi-view face detection. IEEE International Joint Conference on Biometrics; 2014; 1-8.
[10] Mathias M, Benenson R, Pedersoli M, Van Gool L. Face detection without bells and whistles. European Conference on Computer Vision; 2014; 720-735.
[11] Zhang C, Zhang Z. Improving multiview face detection with multi-task deep convolutional neural networks. IEEE Winter Conference on Applications of Computer Vision; 2014; 1036-1041.

[12] Chen D, Ren S, Wei Y, Cao X, Sun J. Joint cascade face detection and alignment. European Conference on Computer Vision; 2014; 109-122.
[13] Zhu X, Ramanan D. Face detection, pose estimation, and landmark localization in the wild. IEEE Conference on Computer Vision and Pattern Recognition; 2012; 2879-2886.
[14] Li H, Lin Z, Brandt J, Shen X, Hua G. Efficient boosted exemplar-based face detection. IEEE Conference on Computer Vision and Pattern Recognition; 2014; 1843-1850.
[15] Zeiler M, Fergus R. Visualizing and understanding convolutional networks. European Conference on Computer Vision; 2014; 818-833.
[16] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. Source: ⟨http://arxiv.org/abs/1409.4842.pdf⟩.
[17] Garcia C, Delakis M. Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence; 2004; 1408-1423.
[18] Osadchy M, LeCun Y, Miller M. Synergistic face detection and pose estimation with energy-based models. Journal of Machine Learning Research; 2007; 1197-1215.
[19] Farfade SS, Saberian M, Li L-J. Multi-view face detection using deep convolutional neural networks. International Conference on Multimedia Retrieval; 2015.
[20] Li H, Lin Z, Shen X, Brandt J, Hua G. A Convolutional neural network cascade for face detection. IEEE Conference on Computer Vision and Pattern Recognition; 2015; 5325-5334.
[21] Kalinovskii IA, Spitsyn VG. Algorithm for face detection on Ultra HD video [In Russian]. Conference on technical vision in control systems; 2015; 95-96.
[22] Köstinger M, Wohlhart P, Roth PM, Bischof H. Annotated Facial Landmarks in the Wild: A Large-scale, real-world database for facial landmark localization. IEEE International Conference on Computer Vision Workshops; 2011; 2144-2151.
[23] Vasilache N, Johnson J, Mathieu M, Chintala S, Piantino S, LeCun Y. Fast convolutional nets with fbfft: A GPU performance evaluation. Source: ⟨http://arxiv.org/abs/ 1412.7580.pdf⟩.
[24] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Source: ⟨http://arxiv.org/abs/1502.03167.pdf⟩.
[25] Lee C-Y, Xie S, Gallagher P, Zhang Z, Tu Z. Deeply-supervised nets. Source: ⟨http://arxiv.org/abs/ 1409.5185.pdf⟩.
[26] Wolf L, Hassner T, Maoz I. Face recognition in unconstrained videos with matched background similarity. IEEE Conference on Computer Vision and Pattern Recognition; 2014; 529-534.
[27] Kalinovskii IA, Spitsyn VG. Face detection algorithm based on the convolutional neural network [In Russian]. Neurocomputers: Development and Applications; 2013; 10: 48-53.
[28] Le QV, Coates A, Prochnow B, Ng AY. On Optimization Methods for Deep Learning. International Conference on Machine Learning; 2011; 265-272.
[29] Köstinger M. Efficient metric learning for real-world face recognition. Graz University of Technology. PhD thesis; 2013.
[30] Pham MT, Cham TJ. Fast training and selection and Haar features using statistics in boosting-based face detection. IEEE International Conference on Computer Vision; 2007; 1-7.
[31] Kienzle W, Bakir G, Franz M, Scholkopf B. Face detection: efficient and rank deficient. Advances in Neural Information Processing Systems; 2005; 673-680.
[32] Jain V, Learned-Miller E. FDDB: A Benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009. University of Massachusetts; 2010.
[33] Yang B, Yan J, Lei Z, Li SZ. Fine-grained evaluation on face detection in the wild. IEEE International Conference on Automatic Face and Gesture Recognition; 2015.
[34] Klare BF, Klein B, Taborsky E, Blanton A, Cheney J, Allen K, Grother P, Mah A, Burge M, Jain AK. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. IEEE Conference on Computer Vision and Pattern Recognition; 2015; 1931-1939.
[35] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves // International Conference on Machine Learning; 2006; 233-240.
[36] Everingham M, Gool LV, Williams C, Winn J, Zisserman A. The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision; 2010; 88(2): 303-338.
[37] Oro D, Fernandez C, Saeta JR, Martorell X, Hernando J. Real-time GPU-based face detection in HD video sequences. IEEE International Conference Computer Vision Workshops; 2011; 530–537.
[38] Nguyen T, Hefenbrock D, Oberg J, Kastner R, Baden S. A software-based dynamic-warp scheduling approach for load-balancing the Viola–Jones face detection algorithm on GPUs. Journal of Parallel and Distributed Computing; 2013; 73(5): 677–685.
[39] Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. OverFeat: Integrated recognition, localization and detection using convolutional networks. Source: ⟨http://arxiv.org/abs/1312.6229.pdf⟩.

Authors' information

Ilya Andreevich Kalinovskii was born in Russia in 1990. He received a master's degree in "Computer Science and Technology" from Tomsk Polytechnic University in 2013. Now he continues to study as a postgraduate student of the Computer Engineering department. His interests include Image processing and analysis, Object recognition, Artificial neural networks and Deep learning methods. E-mail: *kua_21@mail.ru* .

Vladimir Grigorievich Spitsyn (b. 1948) graduated from Tomsk State University in 1970, Radio-Physics department. He works as the Professor of Tomsk Polytechnic University. His research interests are currently focused on neural networks, image processing, and electromagnetic wave propagation in random discrete media. E-mail: *spvg@tpu.ru* .

Received February 1, 2016. The final version – February 16, 2016.