Платформа для создания цифрового профиля посетителей на основе изображений лиц

В.И. Пшенин[1]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. В настоящее время определение персональных характеристик людей является актуальной задачей машинного обучения. Эта статья описывает автоматизированную систему для создания цифрового профиля посетителя на основе изображения его лица. Рассматриваются различные методы обнаружения лиц, описывается процесс обучения нейронной сети для распознавания конкретного человека. Было произведено экспериментальное исследование детекторов лиц и разработанного программного комплекса. Обученная модель имеет точность 99,38% на датасете Labeled Faces in the Wild. Система работает в режиме реального времени.

1. Введение

Извлечение информации о клиентах является ключевой исследовательской задачей в области бизнеса. Чем больше информации компании получают, тем более информативные решения они могут сделать и, соответственно, улучшить экономические показатели. В данной работе рассмотрены методы получения различной информации по лицу человека с последующим созданием его цифрового профиля.

Цифровой профиль — это совокупность цифровых записей о физических лицах, которые включает в себя: пол, возраст, эмоции и уникальную информацию для идентификации человека.

Задачи, которые предстоит решить для создания цифрового профиля всегда находились в ряду самых приоритетных задач для исследователей, работающих в области систем машинного зрения и искусственного интеллекта.

Алгоритмы, осуществляющие автоматический анализ и распознавание лица человека, находят применение в системах технического зрения, робототехнике, системах видеонаблюдения, контроля доступа и безопасности, в маркетинговых исследованиях, а также в качестве повышения работы биометрических систем.

2. Локализация лиц

Из всего многообразия существующих алгоритмов обнаружения лиц можно выделить несколько наиболее актуальных и заслуживающих внимания методов. Рассмотрим особенности, достоинства и недостатки каждого из них.

2.1. Метод Виола-Джонса

Метод Виола-Джонса был предложен Полом Виолой и Майклом Джонсом в 2001 году [1] и стал первым методом, демонстрирующим высокие результаты при обработке изображений в

реальном времени. В алгоритме используется набор признаков, близких к признакам Хаара совместно с вариацией алгоритма AdaBoost.

Достоинства: алгоритм является самым популярным и широко распространённым методом обнаружения лиц, высокая скорость обнаружения за счет использования каскадного классификатора.

Недостатки: требуется большая обучающая выборка и большое время обучения, большое количество ложных обнаружений, ограничения на положение лица при обнаружении.

2.2. Нейросетевые методы

Нейросетевые методы состоят из целого класса различных алгоритмов. Парадигма, лежащая в их основе – это последовательное преобразование сигнала параллельно работающими функциональными элементами, нейронами. Суть процесса обучения таких сетей сводится к уменьшению среднеквадратичной ошибки. Системы обнаружения объектов на изображениях, основанные на нейронных сетях, используют иерархическую структуру. Вначале вектор признаков обрабатывается грубой сетью с высоким уровнем ошибок второго рода, далее, если вектор не был классифицирован как не объект, решение корректируется более точной и медленной сетью [2].

Достоинства: высокая точность обнаружения при правильной настройке параметров сети.

Недостатки: чувствительность к шуму, необходимость в тщательной и кропотливой настройке параметров нейронной сети для получения хороших результатов, склонность к переобучению.

2.3. Гистограмма направленных градиентов

Гистограмма направленных градиентов (англ. Histogram of Oriented Gradients, HOG) — дескрипторы особых точек, которые используются в компьютерном зрении и обработке изображений с целью распознавания объектов. Навнит Далал и Билл Триггс, исследователи INRIA, впервые описали гистограмму направленных градиентов в своей работе на CVPR в июне 2005 года [3]. Данная техника основана на подсчете количества направлений градиента в локальных областях изображения. Метод вычисляется на плотной сетке равномерно распределенных ячеек и использует нормализацию перекрывающегося локального контраста для увеличения точности.

Достоинства: высокая точность обнаружения.

Недостатки: низкая скорость обнаружения, ограничения на положение лица при обнаружении.

3. Распознавание лиц

Далее требуется провести измерения найденных лиц. Измерения, которые кажутся людям очевидными (например, цвет глаз), на самом деле не имеют смысла для компьютера, рассматривающего отдельные пиксели изображения. Исследователи показали, что самый точный подход – это позволить компьютеру самому измерить то, что ему нужно. Глубокое обучение, определяет, какие части лица нужно измерять, лучше, чем люди.

Решение заключается в использовании обученной нейронной сети Resnet34, чтобы получить 128 измерений для каждого лица. Неизвестно, каким частям лица соответствуют эти измерения. Интерес представляет только то, что сеть генерирует почти одинаковые числа по двум изображениям одного и того же человека.

3.1. ResNet

Победителем ILSVRC 2015 с top-5 ошибкой в 3,57 % стал ансамбль из шести сетей типа ResNet (Residual Network), разработанный в Microsoft Research [4].

Авторы ResNet обнаружили, что с повышением числа слоёв свёрточная нейронная сеть может начать деградировать, то есть у неё понижается точность на валидационном множестве. Можно сделать вывод, что проблема состоит не в переобучении сети, так как падает точность и на тренировочном множестве.

Было сделано предположение, что если свёрточная нейронная сеть достигла своего предела точности на некотором слое, то все следующие слои должны будут выродиться в тождественное преобразование, но из-за сложности обучения глубоких сетей этого не происходит. Для того чтобы реализовать эту идею, было предложено ввести пропускающие соединения (Shortcut Connections), изображённые на рисунке 1 [4].

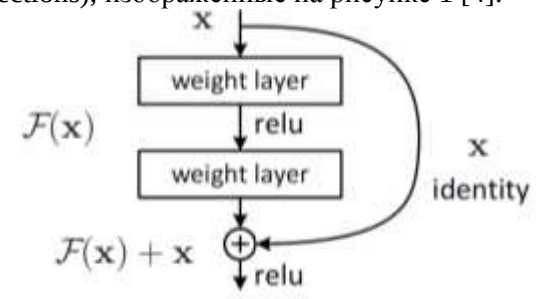

Рисунок 1. Shortcut Connections.

Пусть оригинальная сеть должна вычислять функцию $H(x)$. Определим её остаточную функцию как $F(x) = H(x) - x$, которая теоретически должна быть проще обучаема сетью. Добавив пропускающие соединения, как показано на рисунке 5, сеть учится остаточной функции, которая затем складывается с тождественным преобразованием.

Архитектура сетей ResNet представлена на рисунке 2 [4].

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				

Рисунок 2. Архитектура сети ResNet.

3.2. Обучение сверточной нейронной сети

Нейронная сеть, используемая в данной работе, состоит из последовательных блоков, вида, показанного на рисунке 2 и содержит в себе 34 слоя, выходной слой состоит из 128 измерений.

Функция потерь имеет главное значение при обучении нейронной сети, поскольку она является мерой того, насколько далеко от правильного решения достигается оптимальное решение задачи.

Будем использовать кусочно-линейную функцию потерь [5] (или hinge loss):

$$L(t) = \max(0, 1 - t).$$

Потери зависят от того, больше зазор исследуемого примера 1 или нет: если больше, то связанные с ним потери равны нулю, в противном случае потери тем больше, чем меньше зазор.

Рассмотрим задачу классификации с двумя классами: А и Б. Пусть $А1, А2 \in А$, а $Б1 \in Б$. Тогда запишем функцию потерь для данной задачи:

$$L = \begin{cases} max\ (0, \|А1 - А2\|_2\ - threshold); \\ max\ (0, threshold\ -\ \|А1 - Б1\|_2). \end{cases} \quad (1)$$

Функция потерь отражает тот факт, что все объекты из одного класса находятся на расстоянии threshold друг от друга. И наоборот, если два объекта из разных классов, то они должны находится на расстоянии большем, чем threshold друг от друга. Таким образом, эта функция потерь описывает правила для принятия решения о том, принадлежат ли два объекта одному и тому же классу.

Обучение сверточной сети начиналось со случайно инициализированных весов и использовало функцию потерь (1), которая отображает все объекты в непересекающиеся шары радиусом 0,6. Также на уровне mini-batch в качестве негативных примеров, брали случайные пары объектов, которые имели высокое значение схожести на предыдущем этапе обучения. Этот процесс называется hard negative mining. Для вычисления градиента использовали метод обратного распространения ошибки, основанного на стохастическом градиентном спуске.

Сеть была обучена с нуля на наборе данных около 3 миллионов лиц. Этот набор данных является производным от нескольких наборов данных: FaceScrub [6], VGG [7], изображения из интернета. Количество уникальных классов (людей) в наборе данных составляет 7485. Обучение сети заняло приблизительно 1 сутки на NVIDIA Tesla. Модель имеет точность 99,38% на датасете Labeled Faces in the Wild [8].

4. Результаты экспериментальных исследований

Ранее были описаны различные методы обнаружения лиц на изображениях, а также обучена модель для извлечения признаков. Проверим детекторы лиц и весь алгоритм на предмет быстродействия и точности, что является необходимой процедурой при разработке системы распознавания в режиме реального времени.

4.1. Тестирование детекторов лиц

Протестированы 4 детектора лиц: Haar Cascade Face Detector (OpenCV) [9], Deep Learning based Face Detector (OpenCV) [9], HoG Face Detector (Dlib) [10], Deep Learning based Face Detector (Dlib) [10].

В первую очередь была протестирована возможность нахождения лиц на изображении разрешением 1716 x 1392 пикселей для каждого из детекторов.

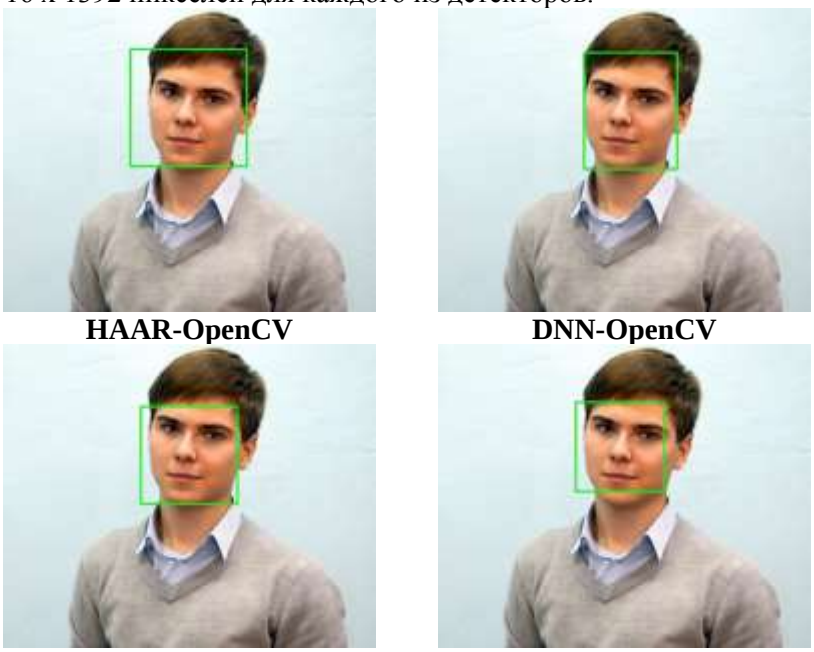

Рисунок 4. Тест детекторов.

Было обнаружено, что ограничивающие рамки отличаются для каждого из детекторов. Рамки детекторов Dlib заметно меньше и часто обрезают лоб или подбородок.

Далее было вырезано лицо и производилось последовательное уменьшение масштаба изображения, чтобы узнать, какой из детекторов может находить самые маленькие лица на изображениях.
Из таблицы 1 видно, что детектор DNN-OpenCV лучше всего справляется с обнаружением маленьких лиц на изображениях.

Таблица 1. Тестирование детекторов при разных размерах изображения.

Размер изображения	HAAR-OpenCV	DNN-OpenCV	HoG-DLIB	MMOD-DLIB
140x140	+	+	+	+
70x70	+	+	+	-
35x35	+	+	-	-
18x18	-	+	-	-
9x9	-	-	-	-

OpenCV предлагает скрипт с набором данных для оценки точности своих моделей, но данная оценка не является справедливой для моделей Dlib, поскольку модели обучались на разных наборах данных, в результате чего при обнаружении лиц ограничивающие рамки сильно отличаются у моделей Dlib и OpenCV. Вместо оценки точности было проведено тестирование детекторов в реальных условиях на видео с разрешением 480x270 и длительностью 26 секунд. Результат представлен в таблице 2.

Таблица 2. Тестирование детекторов в различных условиях.

Тест	HAAR-OpenCV	DNN-OpenCV	HoG-DLIB	MMOD-DLIB
Обычные условия	PASS (+/-)	PASS	PASS	PASS
Non-frontal	FAIL	PASS	PASS (+/-)	PASS
Слабая окклюзия	FAIL	PASS	PASS	PASS
Окклюзия	FAIL	PASS (+/-)	FAIL	PASS

Лучший результат по результатам тестирования показал детектор MMOD-DLIB, за ним следует DNN-OpenCV, HoG-DLIB и HAAR-OpenCV.

Скорость работы детекторов была проверена на видео с разрешением 480x270 и длительностью 26 секунд. Было использовано оборудование: процессор Intel Core i5-6500, видеокарта NVIDIA GeForce GTX 970.
Каждый метод запускался 3 раза, усредненные результаты представлены на рисунке 5.

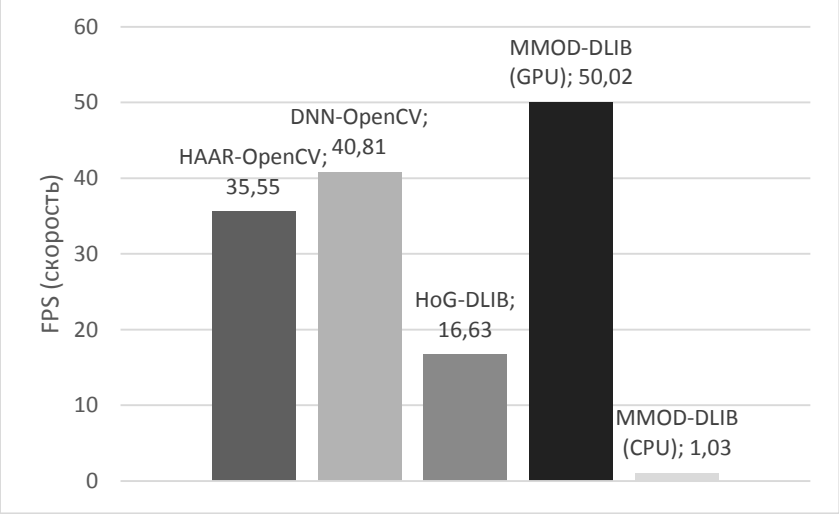

Рисунок 5. График производительности детекторов.

Можно отметить, что все алгоритмы выполняются в режиме реального времени, кроме HoG-Dlib и MMOD-DLIB (CPU). Наилучшую скорость выполнения показал MMOD-DLIB (GPU).

Проведено исследование 4 детекторов лиц. Было выяснено, что лидером по скорости и точности является MMOD-DLIB на графическом процессоре. DNN-OpenCV показал отличные результаты для изображений небольшого размера, но все же уступил по точности детектору MMOD-DLIB. HoG-DLIB оказался гораздо медленнее своих конкурентов и не во всех случаях мог обнаружить лица. HAAR-OpenCV для системы распознавания лиц не подходит из-за большого числа ложных обнаружений.

4.2. Исследование производительности разработанных алгоритмов

Ранее был описан процесс создания цифрового профиля человека в режиме реального времени. Если принять во внимание, что для этого задействованы две нейронных сети и классификатор признаков, задача может показаться слишком сложной для работы в режиме реального времени.

Оценка скорости выполнения производилась на 2 разных настольных компьютерах:

- Intel Core i5-6500 (3,60 GHz) + Nvidia GeForce GTX950;
- Intel Core i7-9700K (4,90 GHz) + Nvidia GeForce GTX970.

В качестве исходных данных использовалось видео с разрешением 1920×1080 пикселей и длительностью 15 секунд. Запущенный без каких-либо оптимизаций, метод выдавал скорость обработки 2,57 FPS на процессоре Intel Core i7.

Оптимизация производилась изменением размера входного изображения. При его уменьшении в 4 раза производительность растет до 18,68 FPS. Далее сменили детектор на MMOD-DLIB и задействовали технологию CUDA. Испытания проводились 4 раза, в таблице 2 приведены усредненные значения.

Таблица 3. Результаты измерения скорости обработки видеофайла в зависимости от разрешения.

Аппаратное обеспечение	Разрешение видеофайла	FPS
CPU, Intel Core i7	1920×1080	2,57
CPU, Intel Core i5	1920×1080	2,25
GPU, Nvidia GTX970	1920×1080	13,97
GPU, Nvidia GTX950	1920×1080	9,23
CPU, Intel Core i7	960×540	9,04
CPU, Intel Core i5	960×540	5,98
GPU, Nvidia GTX970	960×540	30,97
GPU, Nvidia GTX950	960×540	24,30
CPU, Intel Core i7	480×270	24,41
CPU, Intel Core i5	480×270	10,77
GPU, Nvidia GTX970	480×270	47,10
GPU, Nvidia GTX950	480×270	41,01

Таким образом была достигнута максимальная производительность в 47,10 FPS при использовании компьютера, оснащенного Nvidia GeForce GTX970.

4.3. Тестирование разработанного программного обеспечения

Для тестирования использовалась веб-камера и заранее подготовленный видеофайл. Чтобы провести экспериментальные испытания необходимо сначала создать массив характеристик лиц для дальнейшего распознавания.

По рисунку 6 можно сказать о том, что разработанная система работает верно, удалось распознать трех человек: Алексея Раку, Екатерины Пономаревой и Владислава Пшенина. Данные о втором человеке на фото, отмеченного как Unknown, не были использованы при

обучении классификатора. В случае посетителей данные будут храниться в обезличенном формате, привязка будет осуществлять по дате последнего посещения.

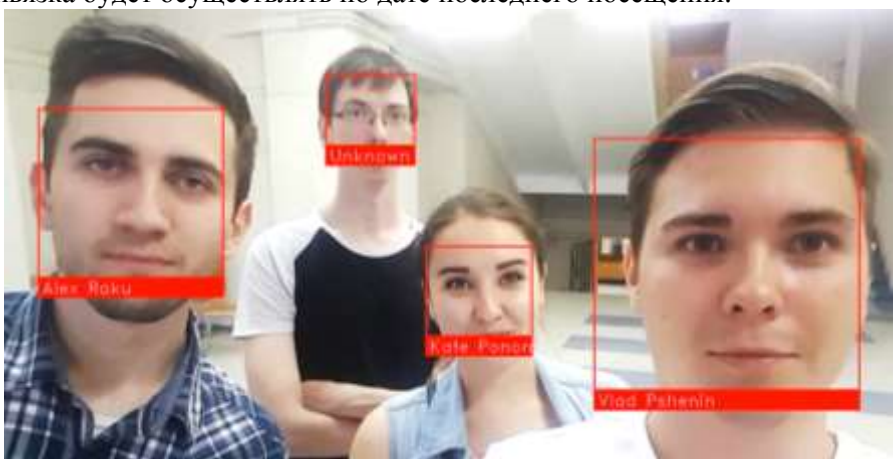

Рисунок 6. Работа программы.

5. Заключение

В ходе работы изучены методы по локализации и распознаванию лиц. Была обучена сверточная нейронная сеть ResNet34. Обученная модель имеет точность 99,38% на датасете Labeled Faces in the Wild. Написана программа для создания цифрового профиля посетителей на основе изображений лиц. Разработанный метод показал отличный результат производительности при работе на системе, оснащённой Nvidia GeForce GTX970. GPU очень сильно увеличивает производительность описанного метода. Итоговая скорость обработки равна 47,10 FPS. Использование глубоких нейронных показывает лучшие результаты по сравнению с альтернативными методами в области компьютерного зрения. Одна из причин успешного применения глубоких нейронных сетей заключается в том, что сеть автоматически выделяет из данных важные признаки, необходимые для решения задачи.

В дальнейшем планируется усовершенствование разработанного программного комплекса и обучение модели распознавания на других популярных архитектурах сверхточных нейронных сетей с использование еще большего набора данных. А также добавить поддержку распознавания возраста, пола и эмоций человека.

6. Благодарности

Работа выполнена при поддержке и руководстве Якимова Павла Юрьевича.

7. Литература

[1] Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2001. – Vol. 1(1). – P. 511-518.

[2] Rowley, H.A. Neural Network-Based Face Detection / H.A. Rowley, S. Baluja, T. Kanade // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1998. – Vol. 20(1). – P. 23-38.

[3] Dalal, N. Histograms of oriented gradients for human detection / N. Dalal, B. Triggs // International Conference on computer vision & Pattern Recognition. – 2005. – Vol. 1. – P. 886-893.

[4] Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. – P. 770-778.

[5] Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах – М.: ДМК Пресс, 2015. – 400 с.

[6] Ng, H. A data-driven approach to cleaning large face datasets / H. Ng, S. Winkler // IEEE International Conference on Image Processing (ICIP), 2014. – P. 343-347.

[7] Parkhi, O.M. Deep face recognition / O.M. Parkhi // British Machine Vision Conference (BMVC). – 2015. – Vol. 1(3). – P. 1-12.
[8] Huang, G.B. Labeled faces in the wild: A database for studying face recognition in unconstrained environments / G.B. Huang // Computer Vision Research Laboratory, 2008 [Electronic resource]. – Access mode: http://vis-www.cs.umass.edu/lfw/ (02.03.2019).
[9] Библиотека обработки изображений OpenCV [Электронный ресурс]. – Режим доступа: http://opencv.org (01.08.2019).
[10] Универсальная кроссплатформенная библиотека программного обеспечения, написанная на языке программирования C++ Dlib [Электронный ресурс]. – Режим доступа: http://dlib.net/ (01.08.2019).

Platform for creating a digital profile of visitors based on face images

V.I. Pshenin[1]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. Currently, the definition of personal characteristics of people is a necessary machine learning. This article is an automated system for creating a digital profile. The process of training a neural network for recognizing people. An experimental study of face detectors and software developers was carried out. The trained model has an accuracy of 99.38% on the dataset. Marked faces in the wild. The system works in real time.