

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ РАСПОЗНАВАНИЯ ЛИЦ, ПОСТРОЕННЫХ С ИСПОЛЬЗОВАНИЕМ БЛОКОВ СТАНДАРТНЫХ АРХИТЕКТУР

Е. В. Каляшов

Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация
Адрес для переписки: ekalyashov@gmail.com

Аннотация—Предмет исследования. В статье рассмотрены варианты конструирования систем распознавания лиц с использованием готовых модулей, и возможности прикладного использования таких систем. **Метод.** Проведено сравнение качества распознавания и скорости обработки различных реализаций систем распознавания. **Основные результаты.** Приведены значения достижимой точности результатов. Даны оценки влияния источников исходных данных на результаты распознавания лиц. **Практическая значимость.** Представленные результаты могут быть использованы при реализации систем распознавания лиц.

Ключевые слова—системы распознавания лиц, детектирование лиц, LFW, Dlib, opencv.

Информация о статье

УДК 004.421

Язык статьи – русский.

Поступила в редакцию 21.07.20, принята к печати 23.09.20.

Ссылка для цитирования: Каляшов Е. В. Сравнительный анализ систем распознавания лиц, построенных с использованием блоков стандартных архитектур // Информационные технологии и телекоммуникации. 2020. Том 8. № 3. С. 94–101. DOI 10.31854/2307-1303-2020-8-3-94-101.

COMPARATIVE ANALYSIS OF FACE RECOGNITION SYSTEMS BUILT USING STANDARD ARCHITECTURE BLOCKS

E. Kalyashov*

The Bonch-Bruевич Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

*Corresponding author: ekalyashov@gmail.com

Abstract—Research subject. The article reviews ways of constructing face recognition systems based on standard modules. **Method.** The study is based on comparison of performance and recognition quality of various pipelines. **Core results.** Values of reached recognition quality and dependencies from a type of original data are presented. **Practical relevance.** The results could be used while implementing various face recognition system pipelines.

Keywords—face recognition, face detection, LFW, Dlib, opencv.

Article info

Article in Russian.

Received 21.07.20, accepted 23.09.20.

For citation: Kalyashov E.: Comparative analysis of face recognition systems built using standard architecture blocks // Telecom IT. 2020. Vol. 8. Iss. 3. pp. 94–101 (in Russian). DOI 10.31854/2307-1303-2020-8-3-94-101.

Введение

Системы идентификации и распознавания лиц в настоящее время активно внедряются во всем мире. Технологии распознавания лиц находят широкое применение в системах наблюдения, обеспечения доступа, учета рабочего времени, авторизации покупок и т. п.^{1, 2}. Существует ряд подходов для выделения лиц на изображении и классификации выделенных лиц. Лучшие результаты на данный момент обеспечиваются при использовании нейронных сетей. В открытом доступе представлены различные решения с открытым исходным кодом. Доступны как примеры использования, так и обученные модули детектирования и распознавания. Такие решения позволяют с минимальными затратами реализовать систему распознавания и идентификации. В данной работе сравниваются результаты применения ряда открытых решений для задач распознавания лиц.

Обзор существующих решений

В общем случае конвейер распознавания лиц состоит из блока детектирования лиц и определения особых точек лица (*Facial Landmark Detection*), блока пре-

¹ Использование функции Face ID на iPhone и iPad Pro [Электронный ресурс]. 2020. URL: <https://support.apple.com/ru-ru/HT208109>

² FindFace распознавание лиц [Электронный ресурс]. URL: <https://findface.pro/>

доработки отдельных лиц (обрезка, масштабирование, трансформация к заданной форме), модуля, выделяющего описатель лица (*embedding vector*), и классификатора. В данном исследовании рассматриваются различные варианты исполнения таких блоков, их совместная работа, оценивается качество распознавания конвейеров, собранных с использованием различных сочетаний блоков. В работе были использованы проект OpenFace³, реализация Face Recognition на PyTorch⁴, реализация сети FaceNet в Keras [1], и решения, предоставляемые библиотекой Dlib⁵. В проектах OpenFace и keras faceNet задействованы варианты нейронных сетей выделения описателя Facenet [2], FR pytorch использует сеть MobileFaceNet [3], в Dlib применяется модифицированная сеть ResNet-34 [4]. При этом в качестве блоков детектирования лица использовались также модули, предоставляемые библиотекой OpenCV⁶, и MTCNN детектор⁷.

Эксперименты по определению качества распознавания проводилась на тестовом наборе данных LFW⁸, в ряде случаев использовалась предоставленная разметка данных (*landmarks*). Для достижения общности проводимых оценок в качестве классификатора применялась косинусное расстояние⁹, дающее нормированный результат в границах от 0 до 2. Для тестовых целей был использован компьютер с параметрами: процессор Intel® Core™ i7 7700 HQ 2.8 GHz, GPU GeForce® GTX 1050 Ti 4 GiB, память 16 GiB, операционная система Ubuntu 18.04. Были использованы версии библиотек OpenFace 0.2.1, Tensorflow 1.7, Dlib 19.19.0, Keras 2.2.0, MTCNN 0.0.9 .

Выделение лиц и предварительная обработка

Сравнение качества детектирования вариантов детекторов, предлагаемых библиотеками Dlib и OpenCV, приведены, например, в [5], готовый код сравнения при считывании изображения с веб-камеры в FaceDetectionComparison¹⁰. Сравнение скорости детектирования приведено в таблице 1, при сравнении исходные кадры с веб-камеры были масштабированы к размеру 300 × 300 точек.

Таблица 1.

Сравнение работы различных детекторов положения лица

Тип детектора	Произв-ть, CPU, fps	Произв-ть, GPU, fps	Качество детектирования
opencv Haar	27	–	низкое
opencv dnn	23	–	высокое
Dlib Hog	39	–	среднее
Dlib cnn	2.9	62	высокое
MTCNN	17	21	высокое

³ OpenFace [Электронный ресурс]. 2019. URL: <https://github.com/cmusatyalab/openface>

⁴ Face Recognition in PyTorch [Электронный ресурс]. 2019. URL: https://github.com/grib0ed0v/face_recognition.pytorch

⁵ Dlib C++ Library [Электронный ресурс]. 2019. URL: <http://dlib.net/python/index.html>

⁶ OpenCV: Open Source Computer Vision Library [Электронный ресурс]. 2020. URL: <https://github.com/opencv/opencv>

⁷ MTCNN [Электронный ресурс]. Электрон. текстовые дан. 2019. URL: <https://github.com/ipazc/mtcnn>

⁸ Labeled Faces in the Wild [Электронный ресурс]. 2018. URL: <http://vis-www.cs.umass.edu/lfw/>

⁹ Cosine similarity [Электронный ресурс]. 2020. URL: https://en.wikipedia.org/wiki/Cosine_similarity

¹⁰ FaceDetectionComparison [Электронный ресурс]. 2018. URL: <https://github.com/spmallick/learnopencv/tree/master/FaceDetectionComparison>

Результаты сравнения говорят в пользу детекторов, реализованных с использованием нейронных сетей. В случае наличия графического ускорителя, поддерживающего архитектуру CUDA, приоритет можно отдать cnn детектору библиотеки Dlib. В то же время, при использовании CPU, данный детектор показывает чрезвычайно низкую скорость работы, и предпочтение может быть отдано dnn детектору OpenCV. Детектор MTCNN показал приемлемую скорость, при работе с CPU скорость работы снижается не сильно. Кроме того, указанный детектор одновременно выделяет особые точки лица (*facial landmarks*), которые могут использоваться для предобработки изображения перед распознаванием. Для выделения особых точек Dlib предоставляет методы, выделяющие наборы из 5 или 68 особых точек (*shape predictor*), и соответствующие процедуры выравнивания лица. Существуют также сторонние алгоритмы выравнивания лица по полученным наборам особых точек¹¹. Модифицированный вариант данного алгоритма (*face align*) использовался для выравнивания изображения по точкам, выделяемым MTCNN. Сравнение выделяемых лиц для различных видов детекторов приведено на рис. 1. Заметно, что регионы, выделенные Dlib cnn детектором, существенно отличается от других.

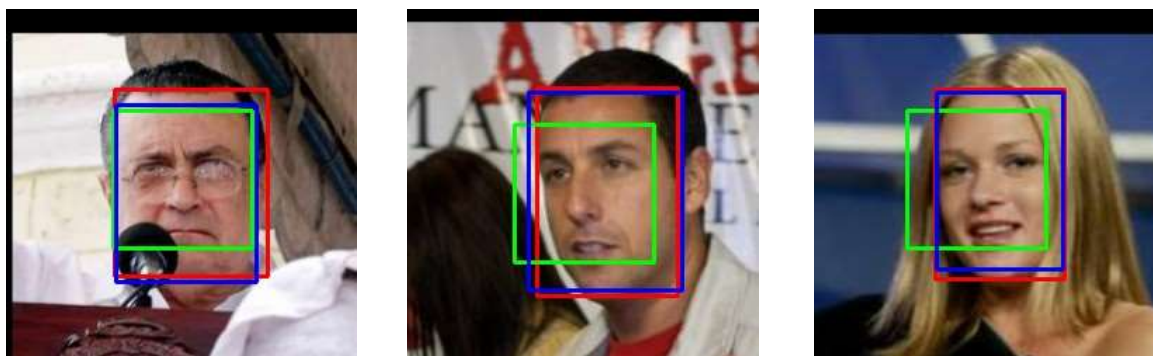


Рис. 1. Сравнение работы различных детекторов.
Красный цвет – регион, выделенный OpenCV dnn детектором,
зеленый – Dlib cnn детектором, синий – MTCNN детектором

Предварительная обработка существенно повышает качество распознавания лиц. Она включает обрезку заданной области изображения, выравнивание (состоящее из поворота и трансформации лица) для приведения к стандартизованному положению, масштабирование полученной области изображения в соответствии с входными размерами распознающей части. Сравнение качества распознавания в зависимости от уровня предобработки изображений, на примере библиотеки Dlib, приведено в таблице 2, использовался тестовый набор данных LFW.

¹¹ Class FaceAligner [Электронный ресурс]. 2018. URL: [https:// github.com/ jrosebr1/ imutils/ blob/master/imutils/face_utils/facealigner.py](https://github.com/jrosebr1/imutils/blob/master/imutils/face_utils/facealigner.py)

Таблица 2.

Сравнение качества распознавания лиц в зависимости от предобработки

Предварительная обработка	Точность, 1 чел.	Точность, 2 чел.	Средняя точность	AUC
Нет	0.6603	0.6677	0.6640	0.7349
Вырезание лица	0.9430	0.9663	0.9547	0.9859
Трансформация и вырезание лица	0.9680	0.9900	0.9790	0.9924

Здесь столбец «Точность, 1 чел.» содержит точность распознавания двух изображений одного человека, «Точность, 2 чел.» – точность распознавания изображений двух разных людей, AUC – площадь под ROC-кривой¹².

Выделяемые особые точки лица и алгоритмы трансформации определяют результирующие изображения, обрабатываемые уже непосредственно модулями распознавания. На рис. 2 приведены результаты обработки входных изображений с использованием библиотеки Dlib. Изображения, полученные с использованием особых точек, выделяемых сетью MTCNN, представлены на рис. 3. Рис. 4 представляет результаты обработки изображений с использованием имеющейся разметки – LFW landmarks.

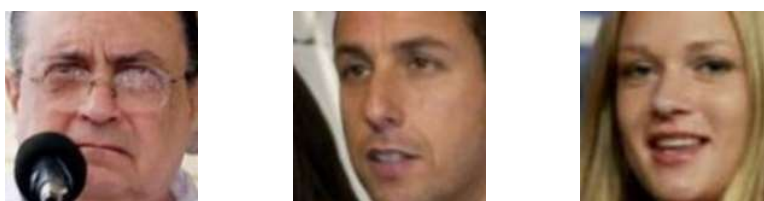


Рис. 2. Результаты предобработки лица средствами Dlib



Рис. 3. Результаты предобработки лица с использованием сети MTCNN

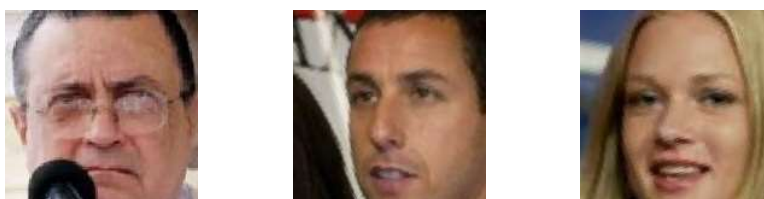


Рис. 4. Результаты предобработки лица с использованием разметки LFW landmarks

Как можно видеть, выделенные лица имеют различный масштаб в отношении исходных изображений, различные углы поворота относительно исходных изображений, а также разные смещения относительно центра изображения.

¹² ROC-кривая [Электронный ресурс]. 2020. URL: [https:// ru.wikipedia.org/ wiki/ ROC-кривая](https://ru.wikipedia.org/wiki/ROC-кривая)

При рассмотрении образцов выделения лиц, полученных разными модулями, становится очевидным, что оценка качества работы таких модулей сравнением выделяемых регионов и трансформированных изображений не имеет смысла. Получить осмысленную оценку работы детекторов можно при сравнении результатов работы конвейеров, в состав которых они входят.

Распознавание лиц

На следующем этапе работы было проведено сравнение качества распознавания конвейеров, составленных из различных блоков, предоставляемых рассматриваемыми решениями. Такое сравнение позволяет оценить устойчивость систем распознавания к качеству предварительной обработки изображений. Результаты сравнения представлены в таблице 3. Значения FPS для конвейера Openface определялись с использованием CPU, для всех остальных конвейеров был задействован графический процессор. Размерность вектора-описателя лица равнялась 128 для всех сетей, кроме FR pytorch, и 256 для случая FR pytorch.

Таблица 3.

Сравнение качества распознавания лиц различным составом конвейеров

Конвейер	Точность, 1 чел	Точность, 2 чел	Средняя точность	AUC	FPS
Openface	0.9240	0.9418	0.9329	0.9789	2.9
MTCNN, face align, keras FaceNet	0.9493	0.9940	0.9717	0.9797	9.8
MTCNN, face align, FR pytorch	0.9413	0.9913	0.9663	0.9768	13.5
MTCNN, face align, Dlib FR	0.9310	0.9727	0.9518	0.9791	14.9
Dlib cnn/predictor, keras FaceNet	0.9680	0.9963	0.9822	0.9894	9.1
Dlib cnn/predictor, FR pytorch	0.9663	0.9857	0.9760	0.9876	12.1
Dlib cnn/predictor, Dlib FR	0.9680	0.9900	0.9790	0.9924	12.8
LFW landmarks, keras FaceNet	0.9903	0.9930	0.9917	0.9996	25.0
LFW landmarks, FR pytorch	0.9913	0.9980	0.9947	0.9994	56.5
LFW landmarks, Dlib FR	0.8703	0.8803	0.8753	0.9434	122.4

Конвейер Openface показал наименьшую точность и скорость распознавания, поддержка GPU у конвейера не реализована. Эксперименты с различными методами предобработки для Openface не проводились. Для остальных сетей распознавания лиц использовались сочетания MTCNN face detector и face align, Dlib cnn detector и shape predictor, а также выделение и трансформация лиц по разметке LFW landmarks. В последнем случае скорость работы конвейера практически отражает скорость работы распознающего модуля, так как трансформация изображения по известным параметрам занимает незначительную часть времени. Можно заметить, что последний метод предобработки привел к существенному увеличению качества распознавания в случае двух сетей, но для решения Dlib качество существенно упало. Стоит отметить, что использование известных разметочных данных позволяет проводить сравнение систем распознавания лиц в одинаковых условиях, однако в реальной жизни на качество распознавания сильное влияние будут иметь методы выделения и предобработки лиц. Можно рекомендовать использовать модуль распознавания Dlib с модулями предвари-

тельной обработки, предоставляемыми этой же библиотекой. Модули Dlib cnn detector и shape predictor дают заметное улучшение качества распознавания для всех сетей. При этом использование Dlib cnn detector на компьютере без GPU нецелесообразно из-за резкого падения скорости обработки изображений.

Влияние исходных данных

Было проведено исследование влияния исходных данных на качество распознавания. Для этого различными устройствами в одинаковых условиях освещения производилась съемка наборов из пяти фотографий одного лица, различающихся поворотами головы. Устройства съемки - встроенная веб-камера ноутбука, выносная веб-камера «Сануон 413g», фронтальная камера смартфона Nexus 5. Рассчитывалось косинусное расстояние (*cosine distance*)¹³ между всеми изображениями из полученных наборов данных. Результаты сравнения приведены в таблице 4. Применены следующие обозначения наборов данных: а) встроенная веб-камера, б) Сануон 413g, в) Nexus 5, г) Nexus 5, лицо с очками. Для распознавания был использован конвейер MTCNN, face align, keras FaceNet.

Таблица 4.

Сравнение качества распознавания лиц в зависимости от исходных данных

Сравниваемые наборы данных	Среднее значение	Стандартное отклонение	Минимальное значение	Максимальное значение
а – а	0.061	0.019	0.030	0.095
б – б	0.086	0.049	0.035	0.189
в – в	0.068	0.032	0.025	0.125
а – б	0.095	0.032	0.051	0.164
а – в	0.101	0.019	0.075	0.149
б – в	0.137	0.035	0.073	0.224
в – г	0.227	0.033	0.179	0.291

Приведенное сравнение показывает явное ухудшение качества распознавания при использовании фотографий, полученных с разных устройств. Однако наблюдаемое повышение косинусной дистанции при сравнении таких наборов данных не критично для успешного распознавания. Появление на фото дополнительных атрибутов, например, очков, имеет явно большее влияние на рассчитываемую величину косинусного расстояния. Тем не менее, все значения из таблицы №4 находятся ниже оптимального порога, полученного при тестах на наборе данных LFW и составляющего значение 0.61.

Заключение

Полученные результаты показывают, что публикуемые решения с открытым исходным кодом позволяют с малыми затратами создать конвейер распознавания лиц с высоким качеством распознавания. Выбор модулей для создания конвейера определяется возможностями используемого аппаратного обеспечения и требуе-

¹³ Cosine similarity [Электронный ресурс]. 2020. URL: https://en.wikipedia.org/wiki/Cosine_similarity

мой производительностью системы распознавания. При этом качество распознавания различных конвейеров оказывается сравнимым. Применение графического ускорителя для расчетов оказывает сильное влияние на достижимую производительность, и наличие специализированного ускорителя в составе системы распознавания представляется практически обязательным. Использование входных данных, полученных с различных устройств, может оказывать некоторое влияние на качество распознавания, но данное влияние не является критичным.

Литература

1. Brownlee, Jason. How to Develop a Face Recognition System Using FaceNet in Keras [Электронный ресурс]. 2019. URL: <https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/>
2. Schroff, Florian, Kalenichenko, Dmitry, Philbin, James. FaceNet: A Unified Embedding for Face Recognition and Clustering [Электронный ресурс]. 2015. URL: <https://arxiv.org/abs/1503.03832>
3. Chen, Sheng, Liu, Yang, Gao, Xiang, Han, Zhen. MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices Clustering [Электронный ресурс]. 2018. URL: <https://arxiv.org/abs/1804.07573>
4. He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian. Deep Residual Learning for Image Recognition Library [Электронный ресурс]. 2015. URL: <https://arxiv.org/abs/1512.03385>
5. Gupta, Vikas. Face Detection – OpenCV, Dlib and Deep Learning (C++ / Python) [Электронный ресурс]. 2018. URL: <https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/>

References

1. Brownlee, Jason. How to Develop a Face Recognition System Using FaceNet in Keras [Electronic resource]. 2019. URL: <https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/> (date of access: 5.06.2020)
2. Schroff, Florian, Kalenichenko, Dmitry, Philbin, James. FaceNet: A Unified Embedding for Face Recognition and Clustering [Electronic resource]. 2015. URL: <https://arxiv.org/abs/1503.03832> (date of access 5.06.2020)
3. Chen, Sheng, Liu, Yang, Gao, Xiang, Han, Zhen. MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices Clustering [Electronic resource]. 2018. URL: <https://arxiv.org/abs/1804.07573>. (date of access: 5.06.2020)
4. He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian. Deep Residual Learning for Image Recognition Library [Electronic resource]. 2015. URL: <https://arxiv.org/abs/1512.03385> (date of access: 5.06.2020)
5. Gupta, Vikas. Face Detection – OpenCV, Dlib and Deep Learning (C++ / Python) [Electronic resource]. 2018. URL: <https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/> (date of access: 5.06.2020)