

Санкт-Петербургский государственный университет

ГАВРИЛОВА Александра Николаевна
Выпускная квалификационная работа
Система трекинга и распознавания лиц

Направление 02.04.02

«Фундаментальная информатика и информационные технологии»

ООП «Цифровые технологии и системы»

Научный руководитель:
доцент кафедры КТС,
к.ф.-м.н. Погожев Сергей
Владимирович

Рецензент: генеральный директор
ООО «ПТМК», к.ф.-м.н. Тукачев
Павел Анатольевич

Санкт-Петербург

2022

СОДЕРЖАНИЕ

Введение.....	3
Постановка задачи.....	4
Обзор литературы.....	6
Глава 1. Подготовка базы изображений	11
1.1 База изображений «CelebA».....	11
1.2 База видео «ChokePoint».....	11
1.3 База изображений «MaskTheFace»	12
1.4 База изображений «WIDER Face Training Images».....	13
Глава 2. Тестирование методов распознавания лиц	14
Глава 3. Тестирование методов детектирования лиц	17
Глава 4. Определение качества изображения.....	20
Глава 5. Тестирование методов трекинга	21
Глава 6. Объединение методов	23
Выводы	26
Заключение	27
Список использованной литературы.....	28

Введение

В настоящее время широко распространено внедрение технологий идентификации и трекинга лиц. Системы распознавания лиц используются для борьбы с преступностью, усиления безопасности на объектах, ускорения пропускного контроля. Кроме того, в настоящее время происходит внедрение системы распознавания лиц для оплаты проезда в общественном транспорте.

Данная область активно развивается и появляются новые методы. Применение наиболее эффективной комбинации методов детектирования, распознавания и трекинга при построении единой целостной системы позволит реализовать процесс идентификации людей в реальном времени и применять построенную систему для решения большого круга задач.

Постановка задачи

Дана последовательность кадров видеопотока $s^i, i = 1, 2, \dots$ фиксированного размера, не менее чем 160×160 .

Требуется обнаружить лица на изображениях, выделить их прямоугольной областью и осуществить распознавание лиц.

Необходимо реализовать трекинг лиц, что может позволить минимизировать вычислительные затраты, так как не потребуется повторно осуществлять распознавание лиц. Для каждого найденного лица на изображении должен быть поставлен в соответствие идентификатор трекинга отслеживаемого лица. Для каждого нового кадра найденные ограничивающие лица прямоугольники должны быть поставлены в соответствие уже существующим идентификаторам трекинга или должны быть созданы новые идентификаторы для отслеживания новых целей.

Предварительно необходимо получить и сохранить векторы признаков $v^j, j = 1, 2, \dots$ для всех шаблонных изображений. Распознавание должно быть осуществлено путем выделения вектора признаков v^* , последующего сравнения v^* с помощью евклидовой метрики расстояния с векторами $v^j, j = 1, 2, \dots$, хранящимися в базе. Если наименьшее из расстояний $d^* = \min(d^j) = \left[\sum_k |v_k^* - v_k^l|^2 \right]^{1/2} = d(v^*, v^l), j = 1, 2, \dots$, будет не более заданного порога, то найденный объект должен считаться соответствующим вектору v^l . Если d^* будет более заданного порога, то вектор v^* следует считать соответствующим «самозванцу».

Для улучшения качества распознавания лиц возможно проводить распознавание только на тех кадрах s^i , где лицо приближено к расположению анфас. Для этого могут быть вычислены углы Эйлера для поворота головы. Кадр потока следует считать допустимым, если угол поворота не превосходит заранее заданного допустимого угла. Допустимым

углом Эйлера при этом можно считать 15 градусов относительно положения анфас.

Поставленная задача может быть разделена на следующие этапы.

1. Подготовка базы изображений.
2. Исследование и тестирование методов распознавания лиц.
3. Исследование и тестирование методов детектирования лиц.
4. Добавление функциональности для определения угла поворота головы и прекращения обработки изображения до выполнения распознавания лица при повороте головы более чем на 15 градусов относительно положения анфас.
5. Исследование и тестирование методов трекинга лиц.
6. Выбор методов детектирования, распознавания и трекинга лиц.
7. Реализация системы распознавания и трекинга лиц.

Обзор литературы

Многие методы детектирования лиц основаны на применении нейронных сетей.

Single Shot Multibox Detector (SSD) [1], в основе которого используются сверточные слои для последовательного уменьшения размера карты признаков, является одним из методов детектирования лиц с использованием нейронных сетей. Последовательное уменьшение карты признаков позволяет находить на изображении лица разного размера. При необходимости реализации методов детектирования лиц на мобильных устройствах, возможно использовать легковесную версию SSD, основанную на MobileNet [2]. Однако, использование оригинальной версии детектора позволяет достичь лучшего качества детектирования.

Нейронная сеть MTCNN (Multitask Cascaded Convolutional Neural Network) [3] основана на каскадной архитектуре и состоит из трех сверточных нейронных сетей. P-Net применяется для получения кандидатов. На вход R-Net подаются результаты предыдущей нейронной сети после чего происходит удаление ошибочных кандидатов и уточнение координат оставшихся ограничивающих лица прямоугольников. Использование нейронной сети O-Net позволяет получить ключевые точки лица. В качестве результата работы MTCNN возвращаются координаты ограничивающих лица прямоугольников и координаты пяти ключевых точек лица: координаты глаз, носа и углов рта.

Существуют методы, основанные не на использовании сверточных нейронных сетей, а на применении каскада классификаторов. Примерами таких методов служат метод Viola – Jones [4], в котором применяются признаки Хаара, и метод, в котором используется каскад гистограмм направленных градиентов [5].

После детектирования лица необходимо выделять признаки и осуществлять распознавание.

В библиотеке распознавания лиц `facenet-pytorch` [6] доступна глубокая нейронная сеть, в основе которой лежит архитектура Inception [7]. В нейронной сети с такой архитектурой применяются смешенные слои: параллельно используется несколько сверточных слоев и слоев субдискретизации, а затем полученные результаты объединяются. Согласно [8] данная архитектура позволяет уменьшить число параметров сети до 20 раз. Шаблонное изображение обрабатывается нейронной сетью и формируется вектор признаков. В данной нейронной сети по умолчанию вектор признаков состоит из 512 элементов. Вектор сохраняется и в дальнейшем сравнивается с векторами признаков тестовых изображений, в процессе чего определяются значения выбранной метрики расстояния. Если минимальное из вычисленных значений удовлетворяет заданному порогу, то тестовое изображение считается соответствующим конкретному шаблонному изображению.

В основе библиотеки `face-recognition` [9] лежат модели библиотеки `Dlib` [10], в том числе на основе `ResNet`. Особенностью остаточных нейронных сетей [11] является наличие дополнительных связей между смежными слоями. Эти связи представляют тождественные отображения и не увеличивают количество параметров нейронной сети. Библиотека `face-recognition` предоставляет два типа моделей для выделения признаков: большего и меньшего размера. Модель большего размера позволяет получить более информативные векторы признаков, что помогает улучшить качество дальнейшего распознавания, но увеличивается время работы метода по сравнению с применением модели меньшего размера. Размерность итогового вектора признаков равна 128. В качестве методов детектирования лиц, библиотекой предоставляются два метода на выбор: основанный на применении сверточной нейронной сети и основанный на применении гистограмм направленных градиентов.

К методам трекинга относится алгоритм `CAMShift` (Continuously Adaptive Mean Shift) [12]. Алгоритм позволяет осуществлять трекинг на

видео, достигая 30 FPS (*англ.* frames per second, кадров в секунду). Работает с цветными изображениями в формате HSV, так как в процессе работы строит гистограммы оттенков (*англ.* hue, оттенок, цвет) для представления распределения вероятностей. Алгоритм основан на MeanShift [13], но в отличие от него, позволяет изменять размер ограничивающего объект прямоугольника, например, при приближении или отдалении объекта от камеры.

Алгоритм FairMOT [14] основан на архитектуре детектирования объектов CenterNet. В отличие от некоторых других алгоритмов, при использовании FairMOT задачи детектирования и повторная идентификация (*англ.* re-identification, re-ID) рассматриваются как равнозначные по важности. Для получения карты признаков re-ID используется сверточный слой со 128-ю ядрами свертки.

В основе метода трекинга множества объектов SORT [15] лежат фильтр Калмана и Венгерский алгоритм. Частота обновления трекинга достигает 260 Hz. Ключевым фактором, влияющим на качество работы метода, отмечен выбор метода детектирования объектов. Алгоритм DeepSORT [16] позволяет достичь лучших результатов трекинга при условии окклюзий благодаря введению более полной метрики, позволяющей хранить информацию о внешнем виде объекта. Это способствует осуществлению более длительного трекинга при различном расположении камеры и пересечениях объектов в кадре. Авторы утверждают, что метод позволяет реализовать трекинг в реальном времени, почти достигая 20 Hz, и может быть рассмотрен наравне с другими подобными методами, работающими в реальном времени. Однако, в статье [17], где также предложен метод ByteTrack, указана сравнительная таблица, в которой DeepSORT достигает только 13.5 FPS, что в несколько раз ниже, чем соответствующее значение для методов SORT (30.1 FPS), ByteTrack (29.6 FPS). Алгоритм ByteTrack [17] позволяет отслеживать перекрытые объекты, продолжая осуществлять трекинг объектов с низким значением уверенности присутствия объекта в ограничивающем

прямоугольнике при детектировании. Для принятия решения о наличии объектов в таком случае, вычисляется степень сходства с уже отслеживаемыми объектами.

В [18] предложена система распознавания и трекинга лиц. Для детектирования и трекинга используются гистограммы. Вектор признаков создается для каждого обнаруженного лица с использованием гистограмм направленных градиентов и фильтра Габора. Чтобы уменьшить размерность вектора признаков до 1536 используется метод главных компонент. Для детектирования лиц в данной системе используется метод Viola – Jones [4].

В [19] описана система распознавания лиц на видеопотоке с применением нечетких методов. В системе присутствуют отдельные блоки, между которыми разделяются задачи детектирования лица, построения модели, трекинга, принятия решений путем вычисления меры сходства с разными классами лиц. Перед распознаванием лица происходит детектирование с использованием метода Viola – Jones. После детектирования определяется цвет кожи лица и 6 ключевых точек глаз и рта. Каждое лицо характеризуется шестью параметрами: цветом кожи, расстоянием между глазами, шириной каждого глаза, расстояниями от рта до каждого глаза.

В системе [20] не используется трекинг лиц. Система состоит из трех модулей: для получения кадра, для детектирования лиц и для распознавания лиц. Для детектирования лиц используется нейронная сеть MTCNN [3]. Затем происходит распознавание лиц путем выделения и сравнения векторов признаков размерности 128. Полученный вектор признаков затем передается SVM для получения результатов распознавания лица.

Главной целью при разработке системы распознавания лиц [21] была оптимизация скорости работы для осуществления распознавания в реальном времени. Для этого используются техники уменьшения размера изображения и применяется трекинг, чтобы уменьшить количество запусков метода распознавания лиц. Для детектирования лиц применяется метод

AdaBoost [22]. При трекинге происходит поиск двух ближайших ограничивающих лица прямоугольников на текущем и предыдущем кадрах. Расстояние между двумя ограничивающими лица прямоугольниками вычисляется по формуле

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Распознавание лица происходит только после того, как оно было детектировано на заданном количестве кадров. В качестве метода распознавания лиц используется алгоритм Eigenface.

Глава 1. Подготовка базы изображений

1.1 База изображений «CelebA»

В процессе работы были подготовлены изображения из базы изображений «CelebA» [23]. В данной базе содержится 202 599 обезличенных фотографий 10 177 субъектов. Было оставлено по 5 изображений для 2 256 субъекта, что в сумме составило 11 280 изображений. При подготовке изображений учитывались отсутствие ошибки разметки и возможность определения признаков лица. Были убраны ошибочно размеченные изображения, сильно размытые и с практически полностью закрытыми лицами. 3 изображения для каждого субъекта являлись шаблонными, а 2 других являлись тестовыми. Примеры изображений из базы «CelebA» [23] представлены на рис. 1.



Рис. 1. Примеры изображений из базы «CelebA» [23]

1.2 База видео «ChokePoint»

Для тестирования на последовательности кадров также было необходимо выбрать базу. В данном случае использовалась база «ChokePoint» [24]. Размер кадра 800×600. Аннотации данной базы представлены в виде xml-документов и содержат координаты глаз субъектов. Была написана программа на языке программирования Python 3 для обработки xml-документов и преобразования координат глаз в координаты ограничивающих лица прямоугольников, чтобы привести аннотации к

единому формату. Координаты ограничивающих лица прямоугольников были получены из координат глаз по формулам

$$w = x_r - x_l + 1,$$

$$x'_l = x_l - \left\lfloor \frac{w}{2} \right\rfloor,$$

$$x'_r = x_r + \left\lfloor \frac{w}{2} \right\rfloor,$$

$$y'_l = \max(y_l, y_r) - w,$$

$$y'_r = \min(y_l, y_r) + \left\lfloor \frac{3w}{2} \right\rfloor,$$

где x_l, x_r, y_l, y_r – координаты глаз на изображении, x'_l, x'_r, y'_l, y'_r – полученные координаты левого верхнего и правого нижнего углов ограничивающего лица прямоугольника.

Корректность получившихся ограничивающих лица прямоугольников была проверена для всех используемых кадров из базы «ChokePoint» [24], а также дополнительно на 2072 изображениях из базы «The Images of Groups» [25].

1.3 База изображений «MaskTheFace»

На момент проведения исследования актуальным также является распознавание лиц в масках. Для этого была использована база изображений «MaskTheFace». Примеры изображений представлены на рис. 2.



Рис. 2. Примеры изображений из базы «MaskTheFace»

1.4 База изображений «WIDER Face Training Images»

Для тестирования методов детектирования лиц была выбрана база изображений «WIDER Face Training Images» [26]. Помимо изображений в базе находятся текстовые файлы с аннотацией. В аннотации содержится информация о количестве лиц на изображении, координаты левого верхнего угла, ширина и высота ограничивающего прямоугольника для каждого объекта и дополнительные параметры, например, выражение лица, поза, освещенность и размытость. Данная база была выбрана, так как она содержит изображения, на которых количество людей, их расположение и условия освещенности не являются постоянными. Поэтому данная база подходит для тестирования работы методов детектирования в реальных условиях. В тестировании использовались 1000 изображений из данной базы. На рис. 3 представлены примеры изображений.



Рис. 3. Примеры изображений из базы «WIDER Face Training Images» [26]

Глава 2. Тестирование методов распознавания лиц

Было осуществлено сравнение двух нейронных сетей для распознавания лиц из библиотек face-recognition [9] и facenet-pytorch [6].

В основе face-recognition [9] использованы модели библиотеки Dlib [10], в том числе модель для распознавания лиц на основе ResNet. Библиотекой на выбор предоставляются две модели для выделения признаков: большего и меньшего размера. В экспериментах использовалась большая нейронная сеть для получения вектора признаков, так как нейронная сеть меньшего размера при исследовании не показала достаточно хороших результатов. Меньшая модель работает быстрее, но точность распознавания лиц значительно хуже. Так как по результатам дальнейших исследований было выяснено, что модель большего размера позволяет обрабатывать кадры в реальном времени, то модель меньшего размера не использовалась в дальнейших экспериментах. Размерность получаемого вектора признаков равна 128.

Библиотека facenet-pytorch [6] использует InceptionResnetV1, обученную на базе изображений VGGFace2. Размерность вектора признаков равна 512. В результате обработки изображения нейронной сетью формируется вектор признаков. Вектор должен быть сохранен и в дальнейшем использован для сравнения с векторами признаков тестовых изображений. Если вычисленное значение евклидовой метрики расстояния удовлетворяет заданному порогу и минимально среди всех таких расстояний, то тестовое изображение считается соответствующим конкретному шаблонному изображению.

Программа для тестирования была написана на Python 3. Тестирование было проведено на GPU с использованием подготовленных изображений из базы «CelebA» [23]. Код программы представлен в репозитории [27].

Сначала были сохранены векторы признаков шаблонных изображений (6768 изображений) и подобраны параметры порога евклидовой меры

сходства. Для библиотеки face-recognition значение порога по умолчанию было выбрано равным 0,6 (рекомендованный порог в документации [9]). Для библиотеки facenet-pytorch рекомендуемое значение порога не было указано в документации, поэтому было проведено тестирование на изображениях при изменении значения порога от 0.55 до 0.95 с шагом 0.01. Было выбрано значение 0,84, как соответствующее наибольшей доле верных ответов. Во время тестирования определялось отношение количества верных положительных ответов каждой нейронной сети (true positive) к общему количеству изображений в тестовой выборке (4512 изображений). Найденное лицо на изображении считалось соответствующим шаблонному изображению, если евклидова мера сходства между двумя векторами признаков не превосходила допустимый порог, и при этом являлась минимальной для всех попарных сравнений.

После этого был составлен набор изображений для проверки верных отрицательных случаев (true negative). Набор изображений состоял из фотографий субъектов, для которых отсутствовали шаблонные изображения в базе. Всего таких изображений было 1461.

Также было измерено время обработки одного изображения и оценена возможность работы в реальном времени.

Результаты тестирования представлены в табл. 1.

Таблица 1. Результаты тестирования

Характеристика	face-recognition	facenet-pytorch
Размерность вектора признаков	128	512
Доля верных положительных ответов	0,84	0,85
Доля верных отрицательных ответов	0,15	0,94
Среднее время обработки одного изображения, сек.	0,02	0,05

Аналогичные результаты, с точностью до двух знаков после запятой, были получены при тестировании на базах «ChokePoint» и «MaskTheFace».

Для библиотеки face-recognition [9] при заданном значении порога 0,6 доля верных ответов на изображениях «самозванцев» составила 0,15. Поэтому было проведено тестирование при других значениях параметра. Параметр изменялся в диапазоне от 0,45 до 0,6 с шагом 0,05.

Результаты тестирования представлены в табл. 2.

Таблица 2. Результаты тестирования при изменении порога face-recognition

Значение порога	Доля верных срабатываний	
	Доля верных отрицательных ответов	Доля верных положительных ответов
0,6	0,15	0,84
0,55	0,71	0,81
0,5	0,93	0,69
0,45	0,98	0,49

Обе библиотеки предоставляют возможность распознавания лиц в реальном времени. По результатам тестирования для дальнейшего рассмотрения была оставлена библиотека facenet-pytorch [6], так как значение доли верных отрицательных ответов при тестировании данной библиотеки значительно превосходило количество верных отрицательных результатов библиотеки face-recognition [9] и значение доли верных положительных результатов тоже оказалось больше.

Глава 3. Тестирование методов детектирования лиц

Были изучены различные методы детектирования лиц. В результате исследования, для дальнейшего рассмотрения были оставлены 2 метода, так как найденная информация про сравнения данных методов оказалась противоречивой, и для выбора наиболее подходящего метода требовалось дополнительное тестирование.

Одним из методов являлся Single Shot Multibox Detector (SSD) [1]. На рис. 4 представлена предложенная авторами архитектура. Рассматривалась реализация из модуля DNN (Deep neural networks) библиотеки OpenCV [28]. Метод основан на остаточной нейронной сети ResNet-10. В качестве результата возвращаются координаты прямоугольных областей, найденных детектором, и числовая характеристика, определяющая вероятность присутствия лица в найденной прямоугольной области. Далее, если числовая характеристика превосходит заданный порог, то лицо считается присутствующим в найденной детектором прямоугольной области.

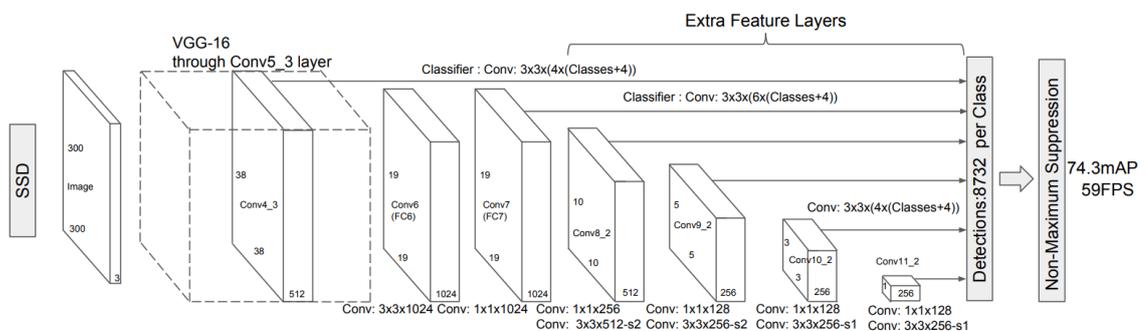


Рис. 4. Архитектура SSD [1]

В качестве второго метода была выбрана нейронная сеть MTCNN (Multitask Cascaded Convolutional Neural Network) [3]. На рис. 5 представлена предложенная авторами архитектура. При тестировании использовалась реализация библиотеки facenet-pytorch [6].

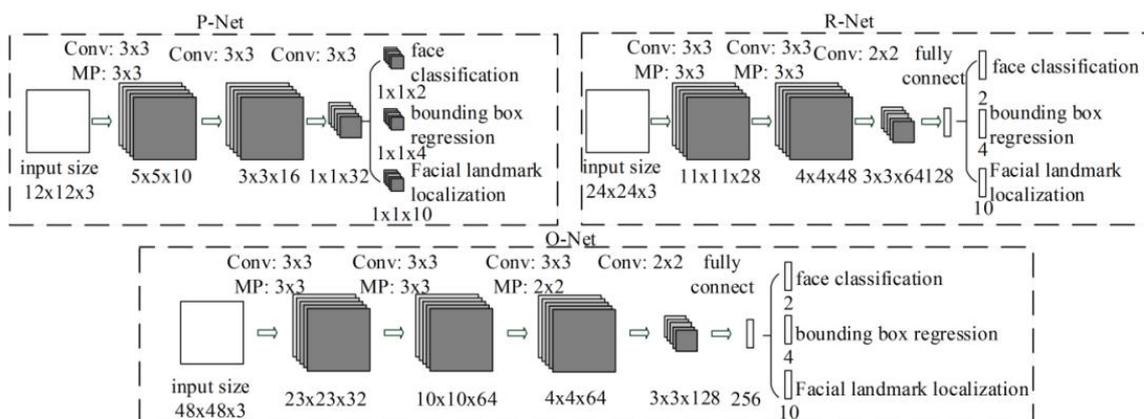


Рис. 5. Архитектура MTCNN [3]

Тестирование проводилось на 1000 изображениях из базы «WIDER Face Training Images» [26]. Программа была написана на языке программирования Python 3.

Для каждого изображения был получен набор ограничивающих лицо прямоугольников. Прямоугольник считался определенным верно, если в аннотации к изображению существовал ограничивающий лицо прямоугольник, который имел с ним значение меры сходства Intersection over Union превосходящее 0,45. В противном случае прямоугольник считался найденным неверно. Формула для вычисления меры сходства:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

где A, B – прямоугольные области, ограничивающие объекты на изображении.

Для сравнения использовались значения метрики точность (precision), и время работы программы.

$$precision = \frac{TP}{TP + FP}$$

где TP – количество верно определенных прямоугольных областей на изображении, FP – количество неверно определенных прямоугольных областей на изображении.

Принимаемые значения могут изменяться от 0 до 1. При этом чем выше значение метрики, тем лучше.

Результаты тестирования представлены в табл. 3.

Таблица 3. Результаты тестирования

Характеристика	SSD	MTCNN
Точность (precision)	0,56	0,9
Время обработки 1000 изображений, сек.	826,98	595,24

По результатам вычислений метрики и скорости работы программы, была выбрана MTCNN. Кроме того, MTCNN при дальнейшем тестировании было ускорено на GPU, что позволило до четырех раз увеличить скорость детектирования. Еще одним преимуществом MTCNN является возможность выделения ключевых точек на лице, однако, это не являлось ключевым критерием при выборе данного метода.

Глава 4. Определение качества изображения

Предварительный отбор кадров видеопотока для дальнейшего распознавания может позволить улучшить качество распознавания и увеличить скорость работы системы. В качестве характеристики качества изображения может быть взят угол поворота лица относительно положения анфас. Для этого возможно использовать углы Эйлера.

Была написана программа для определения углов Эйлера. Для этого были использованы методы библиотеки OpenCV [28]: solvePnP, Rodrigues, RQDecomp3x3. Угол поворота головы не должен был превосходить 15 градусов относительно положения анфас.

Угол поворота вычислялся на основе 5 ключевых точек лица, которые могут быть получены после применения метода детектирования лиц MTCNN [3].

На рис. 6 представлен пример результата работы MTCNN.



Рис. 6. Пример ограничивающего лицо прямоугольника и пяти ключевых точек, полученные как результат работы MTCNN [3]

Глава 5. Тестирование методов трекинга

В результате обзора существующих методов трекинга, для дальнейшего рассмотрения были оставлены метод SORT и метод ByteTrack.

В основе метода трекинга SORT [15] лежат фильтр Калмана и Венгерский алгоритм. Ключевым фактором, влияющим на качество работы метода, является выбор метода детектирования лиц. В данной работе в качестве метода детектирования лиц применялась нейронная сеть MTCNN [3].

Алгоритм ByteTrack [17] позволяет отслеживать перекрытые лица, продолжая осуществлять трекинг при низком значении уверенности присутствия лица в ограничивающем прямоугольнике после детектирования. Для принятия решения о наличии лица в этом случае, вычисляется степень сходства с уже отслеживаемыми лицами, которая определяется мерой сходства Intersection over Union.

В часто используемых метриках качества работы методов трекинга (например, MOTA) большинство используемых показателей зависят от метода детектирования лиц. Также в них используется значение количества изменений идентификатора трекинга для одного и того же отслеживаемого объекта. Так как для обоих методов трекинга использовался один и тот же ранее выбранный метод детектирования MTCNN [3], то при тестировании измерялось только количество изменений идентификаторов. Кроме того, измерялось время работы программы на одном и том же потоке кадров при использовании методов SORT и ByteTrack.

Была подготовлена тестовая среда на языке программирования Python 3. Для тестирования использовалась база «ChokePoint» [24].

Результаты тестирования представлены в табл. 4.

Таблица 4. Результаты тестирования

Характеристика	SORT	ByteTrack
Количество ложных изменений идентификатора	5	3
Среднее время обработки кадра тестового потока кадров, сек.	0.01659	0.01659

Общее время обработки тестового потока кадров и среднее время обработки кадра одинаковое с точностью до 5 знаков запятой. Оба метода трекинга имеют достаточную скорость работы, чтобы обрабатывать видео. Количество ложных изменений идентификатора трекинга отслеживаемого лица оказалось больше у метода SORT. По результатам тестирования был выбран метод ByteTrack.

Глава 6. Объединение методов

Выбранные методы были объединены в систему трекинга и распознавания лиц.

В качестве метода детектирования была выбрана нейронная сеть MTCNN, которая кроме ограничивающих лица прямоугольников возвращает координаты пяти ключевых точек лица.

Была добавлена функциональность для определения угла поворота лица. Добавлен дополнительный параметр, который определяет, стоит ли учитывать угол поворота лица. Если угол поворота лица нужно учитывать, то он определяется с помощью методов библиотеки OpenCV. В этом случае распознавание лица не проводится при угле отклонения от положения анфас более чем на 15 градусов.

Методом трекинга был выбран метод ByteTrack. При работе трекинга, найденным детектором лицам присваиваются идентификаторы трекинга, что позволяет производить распознавание только ранее не распознанных лиц.

Для распознавания лиц используется библиотека facenet-pytorch. Сначала необходимо получить и сохранить все векторы признаков шаблонных изображений. При передаче тестового изображения, обрезанного так, чтобы в нем содержалась только лицо, сначала определяется вектор признаков тестового изображения, затем он сравнивается с сохраненными шаблонными векторами признаков с помощью евклидовой меры сходства. Если минимальное значение евклидовой меры сходства не превосходит порог 0,84 (определен при тестировании в главе 2), то вектор признаков тестового изображения считается соответствующим шаблонному вектору с данным значением меры сходства. Если все вычисленные значения меры сходства оказались больше заданного порога, то считается, что не найдено соответствующего шаблонного изображения.

На рис. 7 представлены примеры работы системы на потоке кадров.

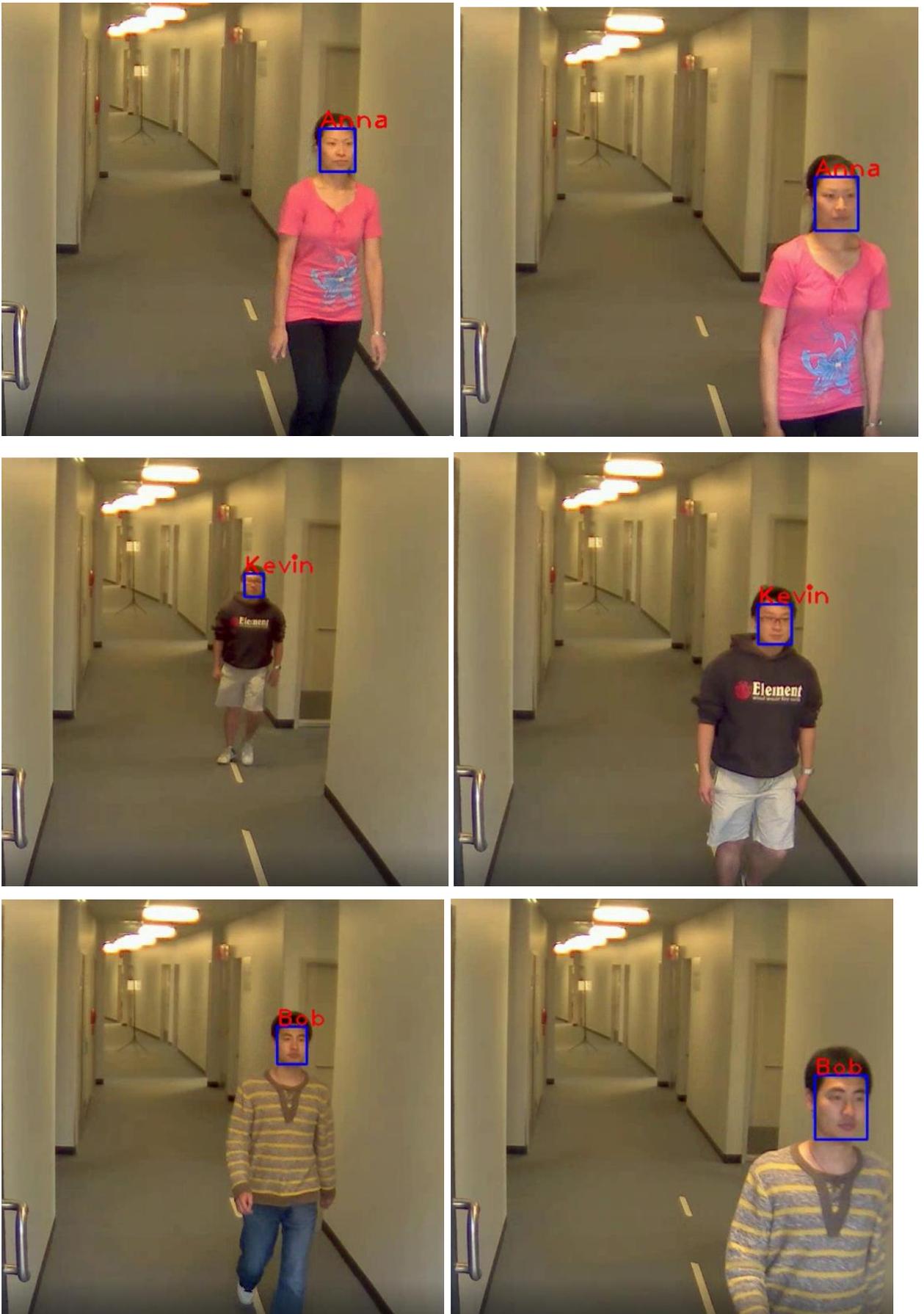


Рис. 7. Примеры работы полученной системы на потоке кадров

Было измерено среднее время обработки кадра тестового потока базы «ChokePoint» при использовании метода трекинга и без его использования.

Результаты тестирования представлены в табл. 5.

Таблица 5. Время работы с трекингом и без трекинга

Характеристика	Без трекинга	С трекингом
Среднее время обработки кадра тестового потока кадров, сек.	0,03787	0,02945

При добавлении трекинга среднее время обработки видео потока уменьшилось в 1,3 раза благодаря уменьшению количества раз применения метода распознавания лиц.

Выводы

По полученным результатам можно сделать вывод, что применение трекинга позволяет увеличить скорость работы системы распознавания лиц, так как количество вызовов метода распознавания уменьшается. Скорость работы системы можно увеличить при использовании GPU. Однако, при недоступности GPU, используемые методы позволяют производить вычисления на CPU.

База изображений «CelebA» изначально содержит неудачные изображения и требует дополнительной обработки: не верная разметка, полностью закрытые лица, соответствие одного субъекта нескольким идентификаторам, что может привести к некорректным результатам тестирования.

Библиотека face-recognition не подходит в тех случаях, когда важно определять «самозванцев», однако, использование библиотеки facenet-pytorch показало хорошие результаты при тестировании на таких изображениях.

Метод детектирования лиц MTCNN позволяет получать не только координаты ограничивающих лица прямоугольников, но и координаты пяти ключевых точек лица, которые могут быть использованы, например, при определении угла поворота головы.

Заключение

В результате выпускной квалификационной работы:

- подготовлена база изображений для тестирования методов, которая содержит как статические изображения, так и поток кадров;
- рассмотрены существующие методы детектирования, распознавания и трекинга лиц;
- подготовлена тестовая среда с использованием языка программирования Python 3;
- проведено тестирование двух библиотек распознавания лиц на базе изображений и выбрана библиотека facenet-pytorch, использование которой помогло достигнуть более качественных результатов;
- проведено сравнение качества детектирования для двух методов: SSD в реализации библиотеки OpenCV и MTCNN в реализации библиотеки facenet-pytorch и по результатам тестирования на 1000 изображениях из базы «WIDER Face Training Images» для дальнейшего рассмотрения выбрана нейронная сеть MTCNN;
- написана программа на языке программирования Python 3 для определения угла поворота лица;
- проведено тестирование методов трекинга SORT и ByteTrack на потоке кадров и метод ByteTrack был выбран для дальнейшего применения в рамках системы трекинга и распознавания лиц;
- выбранные методы объединены в общую систему и измерено среднее время обработки потока кадров при использовании метода трекинга и без его использования.

Список использованной литературы

1. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. SSD: Single Shot Multibox Detector // European conference on computer vision. 2016. P. 21–37.
2. Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // arXiv: 1704.04861. 2017. P. 1–9.
3. Zhang Z., Li Z., Qiao Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks // IEEE Signal Processing Letters. Vol. 23, No 10. 2016. P. 1499-1503. doi: 10.1109/LSP.2016.2603342.
4. Viola P., Jones M. J. Robust real-time face detection // International Journal of Computer Vision. 2004. Vol. 57. No 2. P. 137–154.
5. Zhu Q., Yeh M.-C., Cheng K.-T., Avidan S. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients // 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). 2006. P. 1491-1498.
6. Репозиторий библиотеки facenet-pytorch [Электронный ресурс]: URL: <https://github.com/timesler/facenet-pytorch> (дата обращения: 20.05.2022).
7. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going deeper with convolutions // CoRR, abs/1409.4842. 2014.
8. Schroff F., Kalenichenko D., Philbin J. FaceNet: A Unified Embedding for Face Recognition and Clustering // arXiv: 503.03832. 2015. P. 1-10.
9. Техническая документация библиотеки face_recognition [Электронный ресурс]: URL: https://face-recognition.readthedocs.io/en/latest/face_recognition.html (дата обращения: 20.05.2022).
10. Техническая документация библиотеки Dlib [Электронный ресурс]: URL: <http://dlib.net/> (дата обращения: 20.05.2022).

11. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778.
12. Bradski G. R. Computer Vision Face Tracking for Use in a Perceptual User Interface // Intel Technology Journal, 2nd Quarter. 1998.
13. Fukunaga K. Introduction to Statistical Pattern Recognition // Academic Press, Boston. 1990.
14. Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. // arXiv: 2004.01888. 2020.
15. Bewley A., Ge Z., Ott L., Ramos F., Upcroft B. Simple online and realtime tracking // 2016 IEEE international conference on image processing (ICIP), IEEE. 2016. P. 3464– 3468.
16. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric // 2017 IEEE international conference on image processing (ICIP), IEEE, 2017. P. 3645–3649.
17. Zhang Y., Sun P., Jiang Y., Yu D., Yuan Z., Luo P., Liu W., Wang X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box // arXiv: 2110.06864. 2021.
18. Nagendra Shruthi, Baskaran R., Abirami S. Video-Based Face Recognition and Face-Tracking using Sparse Representation Based Categorization // Procedia Computer Science. 2015. Vol. 54. P. 746–755.
19. Hatimi H., M. Fakir M., Chabi M. Face Recognition Using a Fuzzy Approach and a Multi-agent System from Video Sequences // 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV). 2016. P. 442–447.
20. Eiamsaard, K., Bamrunghai P., Jitpakdeebodin S. Smart Inventory Access Monitoring System (SIAMS) using Embedded System with Face Recognition // 2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE). 2021. P. 1–4.

21. Cheong W. L., Char C. M., Lim Y. C., Lim S., Khor S. W. Building a computation savings real-time face detection and recognition system // 2010 2nd International Conference on Signal Processing Systems. 2010. Vol. 1. P. 815–819.
22. Freund Y., Schapire R. E. A decision-theoretic generalization of on-line learning and an application to boosting // Computational Learning Theory: Eurocolt'95. 1995. P. 23–37.
23. Liu, Ziwei and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou. Deep Learning Face Attributes in the Wild // Proceeding of International Conference on Computer Vision. 2015.
24. Wong Y., Chen S., Mau S., Sanderson C., Lovell B.C. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition // IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops. 2011. P. 81-88.
25. Gallagher A. C., Chen T. Understanding images of groups of people // 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009. P. 256-263.
26. Yang S., Luo P., Loy C. C., Tang X. WIDER FACE: A Face Detection Benchmark // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 5525–5533.
27. Репозиторий с исходным кодом программ [Электронный ресурс]: URL: <https://github.com/vsister/Tracking-and-Recognition> (дата обращения: 25.05.2022).
28. Техническая документация библиотеки OpenCV [Электронный ресурс]: URL: <https://opencv.org/> (дата обращения: 20.05.2022).