

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
Кафедра информационных систем

УТВЕРЖДАЮ

Заведующий кафедрой

_____ П.П. Дьячук

подпись

« _____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 — «Информационные системы и технологии»

Применение технологий машинного обучения для трекинга людей по
видеоданным

Руководитель

подпись, дата

доцент, к.т.н.

А.В. Пятаева

Выпускник

подпись, дата

К.А. Калинин

Красноярск 2020

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных Технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ П.П. Дьячук

подпись инициалы, фамилия

« _____ » _____ 2020 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Калинину Кириллу Александровичу

Группа: КИ16-13Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Применение технологий машинного обучения для трекинга людей по видеоданным»

Утверждена приказом по университету № 6499/с от 22.05.2020 г.

Руководитель ВКР: А.В. Пятаева.

Исходные данные для ВКР: Требования к разрабатываемой системе, методические указания научного руководителя, учебные пособия.

Перечень разделов для ВКР: Введение, Машинное обучение для трекинга людей по видеоданным, Разработка системы автоматического трекинга людей, Экспериментальная часть, Заключение, Список использованных источников.

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2019.

Руководитель ВКР

подпись, дата

А.В. Пятаева

инициалы и фамилия

Задание принял к исполнению

подпись, дата

К.А. Калинин

инициалы, фамилия

« ____ » _____ 2020 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Информационные системы и технологии» содержит 55 страниц текстового документа, 14 иллюстраций, 32 использованных источника.

Актуальность.

Трекинг позволяет проводить множество исследований, таких как: изучение аномального поведения людей, идентификация каждого отдельного человека без взаимодействия с ним, авторизация на различных ресурсах, ведение статистики и т.д.

Объектом исследования является система, способная проводить трекинг людей при помощи последовательностей изображений и предсказаний нейросетевой модели.

Предметом исследования является метод трекинга, базирующийся на алгоритме глубоких признаков, методе центроидов и фильтров Калмана.

Целью работы является разработка системы автоматического трекинга людей по видеоданным, позволяющей решить проблему обнаружения и последующего трекинга людей на сцене.

Основные задачи:

- обзор высокоточного метода трекинга людей по видеоданным;
- разработка собственной системы автоматического трекинга людей по видеоданным;
- тестирование созданной автоматической системы;
- оценка качества разработанной автоматической системы, а также обсуждение ее плюсов и минусов.

Практическая значимость проекта заключается в том, что с помощью данной системы возможно автоматическое выделение людей на сцене, а также сохранение результатов трекинга и просмотр результатов трекинга.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Машинное обучение для трекинга людей по видеоданным.....	5
1.1 Процесс трекинга	5
1.2 Технологии автоматизированного трекинга людей по видеоданным.....	7
1.2.1 Предобработка обнаружений.....	7
1.2.2 Обработка обнаружений	8
1.2.3 Ассоциация обнаружений.....	10
1.3 Анализ трекинга на основе точек артикуляции.....	11
1.3.1 Анализ модуля выборки.....	11
1.3.2 Анализ модуля измерений	15
1.3.3 Анализ модуля ассоциаций.....	17
2.1 Алгоритм глубокой сортировки и нейронная сеть.....	21
2.2 Процесс трекинга людей	24
2.3 Проблемы трекинга.....	26
2.3.1 Конкретизация проблем и способов их решения	26
2.3.2 Решение проблем	28
3 Экспериментальная часть.....	39
3.1 Программная реализация. Библиотеки	39
3.1.1 Трек	40
3.1.2 Трекер.....	42
3.1.3 Визуализатор	42
3.1.4 Работа системы трекинга	43
3.1.5 Параметры запуска системы трекинга.....	46
3.2 Тестирование системы.....	47
3.3 Оценка точности	50
3.4 Примеры работы трекера	52
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	59

ВВЕДЕНИЕ

На сегодняшний день, всевозможные методы трекинга людей переживают пик своей популярности и развития ввиду возможности не только обнаружения человека, или какого бы ни было объекта, но и возможности предугадывания направления его движения. Существует множество методов трекинга людей по видеоданным. Например, отслеживание траектории движения человека при помощи распознавания его лица, либо отслеживание при помощи переносных устройств. Одним из наиболее успешных из них является трекинг при помощи технологий машинного обучения. Частным случаем машинного обучения является технология нейронных сетей.

Целью данной работы является разработка системы автоматического трекинга людей по видеоданным, позволяющей решить проблему обнаружения и последующего отслеживания людей на сцене с использованием технологий машинного обучения.

В связи с этим возникают следующие задачи:

1. Провести анализ технологий машинного обучения для трекинга по видеоданным.
2. Выбрать нейросетевую модель для решения задачи обнаружения людей на сцене.
3. Разработать метод автоматического трекинга людей по видеоданным.
4. Оценить качество работы трекинга разработанной автоматической системы.

1 Машинное обучение для трекинга людей по видеоданным

В настоящее время методами, позволяющими осуществить трекинг людей по видеоданным, с применением нейросетевых моделей являются:

- метод обнаружения артикуляций и стратегия сшивания [1];
- использование сочетания различных архитектур нейросетевых моделей, например, сверточной и рекуррентной [2];
- метод трекинга людей с использованием камер глубины [3].

Перечисленные методы позволяют решать различные практические задачи на основе данных трекинга, такие как прокторинг пользователей при осуществлении дистанционного обучения, отслеживание заболеваний пожилых людей по паттернам поведения, нарушение границ, а также обнаружение аномального поведения людей [4] и др.

1.1 Процесс трекинга

Процесс работы системы трекинга по видеоданным представляет собой отслеживание объектов, будь то человек, мяч или же автомобиль, в ряде кадров (фреймов). Алгоритм работы системы трекинга объектов интереса на основе нейронной сети по видеоданным представляет из себя определенную последовательность шагов. Для отслеживания человека на первом кадре, когда он только появился, используются различные методы по его обнаружению, после чего системой наблюдаемому присваивается уникальный идентификационный номер. Далее системой проводятся попытки найти этого же человека на сцене и назначить ему соответствующий идентификационный номер, сгенерированный на первом кадре. Это делается потому, что при смене кадра, одного и того же человека система определит как нового. Если человек покидает сцену, т.е. перестает появляться в последующих кадрах, его

идентификационный номер подлежит к удалению. При появлении нового человека на сцене, весь процесс повторяется.

Задача трекинга людей осложняется множеством факторов, таких как:

- появление внешне похожих людей;
- пешеходы в кадре могут загораживать друг друга (окклюзия);
- человек может зайти за какой-нибудь объект, а затем появиться в последующих кадрах.

Глубокое машинное обучение старается миновать вышеописанные проблемы, показывая при этом высокую эффективность трекинга. Так, при использовании двух нейросетевых моделей архитектур CNN (Convolutional Neural Network) и RNN (Recurrent Neural Network) [2] на датасете MOT16 [5], количество неверных переключений идентификаторов уменьшилось примерно на 29%, в сравнении с результатом, полученным при использовании только CNN архитектуры (рис. 1). Кроме этого, количество ложных срабатываний уменьшилось почти на 50% в сравнении с методом ранжирования [6].

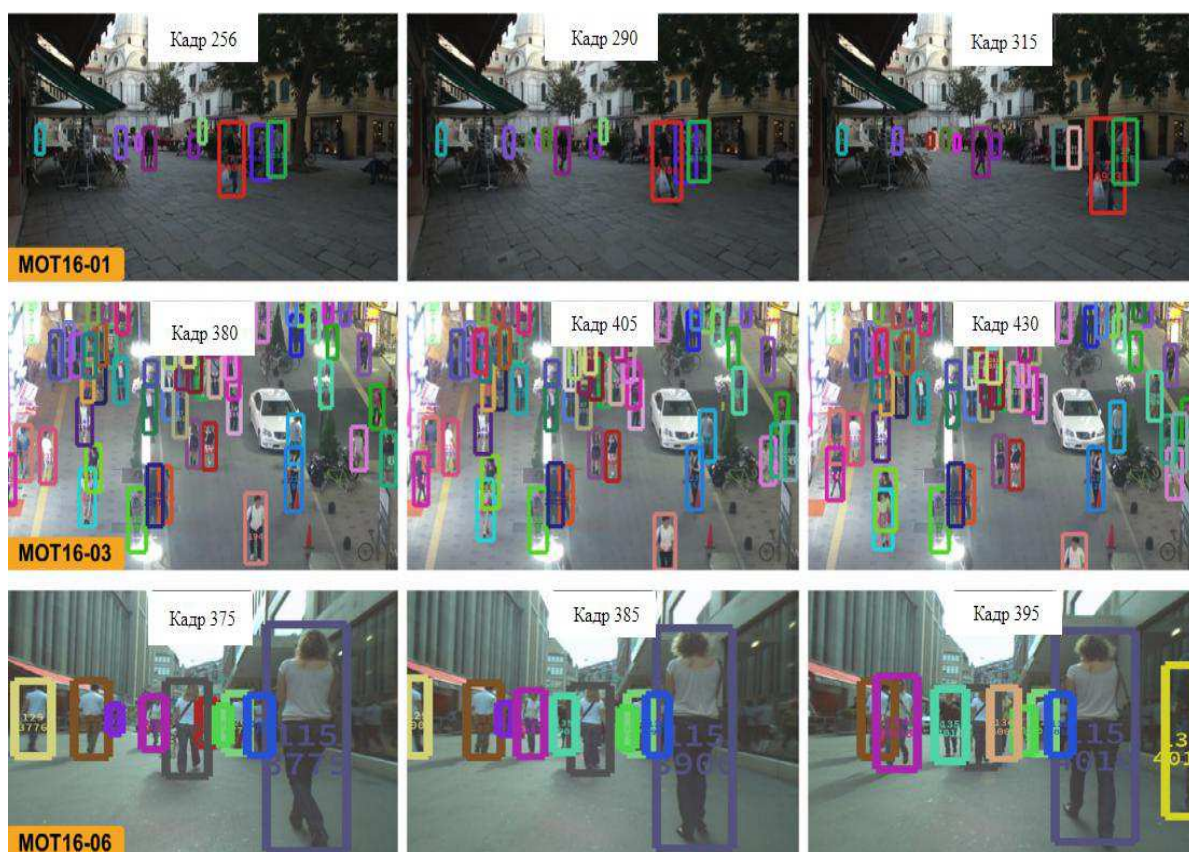


Рисунок 1 — Пример работы системы трекинга

1.2 Технологии автоматизированного трекинга людей по видеоданным

В настоящее время, основные подходы трекинга основаны на глобальной структуре DBT (Detection-Based Tracking), суть которой заключается в локализации объектов в каждом кадре и последующего связывания этих наблюдений в треки, без какой-либо маркировки [7]. Ранние методы DBT использовали фоновое моделирование [8]. Суть фонового моделирования заключается в накоплении информации о фоне сцены. Каждый пиксель изображения сравнивается с фоновым, после чего возникает ассоциация, является ли пиксель частью фона, либо он относится к движущемуся объекту. В противовес этому, современные методы трекинга базируются на попытке обнаружения объекта в каждом кадре для последующей обработки [9].

Сначала происходит процесс предобработки обнаружений, в котором предпринимаются попытки из уже существующих обнаружений выделить наиболее точные. Затем происходит обработка обнаружений, при которой выделяется каждый отдельный объект, этому объекту назначается свой идентификационный номер, система находит информацию о внешних деталях объекта для проведения трекинга. В завершении система проводит ассоциации между найденными ранее объектами с целью создания трека.

1.2.1 Предобработка обнаружений

Обнаружения представляют собой текстовый CSV файл. Формат данных этого файла следующий: номер фрейма, идентификационный номер обнаружения, координата x левой верхней части рамки обнаружения, координата y верхней левой части рамки, ширина рамки, высота рамки, точность данного обнаружения, а также три координаты пространства, если речь идет об обнаружениях в трехмерном пространстве. Все эти данные

позволяют системе выделить каждого человека на сцене. В целях данной работы требуются только первые шесть значений, остальные были проигнорированы.

Многие существующие алгоритмы трекинга используют содержащиеся в папке с последовательностями изображений обнаружения, то есть готовые данные о местоположении каждого человека на изображении. Известные наборы данных, такие как MOT [5] и KITTI [10], удовлетворяют этому описанию.

В отличие от методов, которые используют обнаружения из набора данных, Kim и соавторы [11] используют нейросетевую модель Fast R-CNN (Fast Region-based Convolutional Neural Network) для обнаружения людей на сцене [12]. Задача модели выбирать обнаружения только с высоким показателем достоверности, чтобы повысить качество работы системы. Tang и соавторы [13] используют методы подавления немаксимумов NMS (Non-Maximum Suppression) для решения проблемы со множественным обнаружением одного и того же объекта. Другие исследователи [14, 15] создают группы кластеров обнаружений, чтобы избежать создания лишних или нежелательных идентификаторов наблюдаемых объектов, и неправильных целей для трекинга.

1.2.2 Обработка обнаружений

После предобработки обнаружений возникает потребность в идентификации каждого объекта на сцене для построения трека. Для этой задачи существуют функции, измеряющие разницу или актуальность между наблюдениями; эти функции оказывают значительное влияние на результат отслеживания в многолюдных сценах.

Некоторые авторы [9, 13] используют выведенные вручную функции для измерения различий между обнаружениями. Другие [16, 17] рассматривают

временные параметры в рамках логистической регрессии. Здесь идет оценка разности временных интервалов для определения сходства между обнаружениями в процессе трекинга. Также существуют эксперименты [17, 18], которые успешно используют глубокие нейронные сети DNN (Deep Neural Network), в данном случае сиамские, для визуального отслеживания похожих объектов, с целью получения более точных параметров схожести.

Кроме прочих, существуют методы повторной идентификации человека. С развитием глубокого обучения, были предложены модели повторной идентификации на основе рекуррентных нейронных сетей CNN, среди которых популярным типом является сиамская модель, использующая пары изображений [19] в качестве входных данных. Tang и соавторы [17] обучает сиамскую нейронную сеть на основе архитектуры VGG-16 [20], изображенной на рис. 2, для повторной идентификации пешеходов при отслеживании нескольких человек.

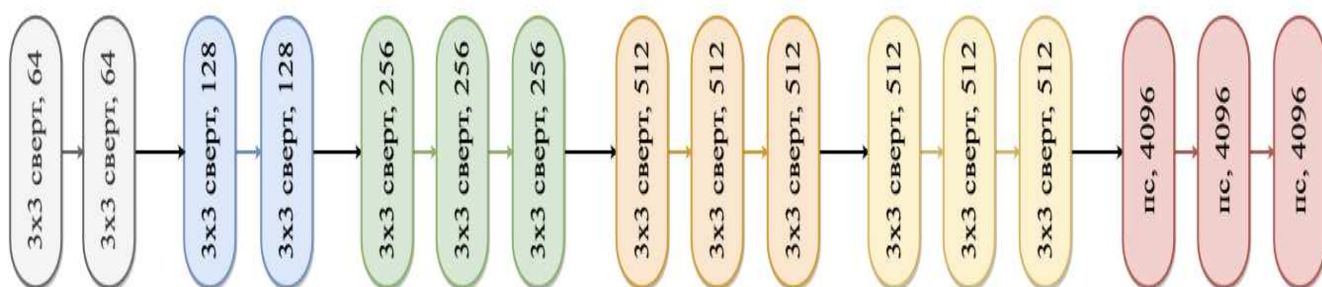


Рисунок 2 — Архитектура сверточной нейросетевой модели VGG-16

Cheng и соавторы [21] использовал функцию потери триплетов, которая принимает три изображения в качестве входных данных: первое изображение является изображением запроса и, соответственно, искомым изображением, второе изображение является положительным, то есть максимально похожим на изображение запроса, а последнее — отрицательным, то есть в меньшей мере похожее на изображение запроса. После первого сверточного слоя, то есть слоя, четыре перекрывающихся части тела разделяются для каждого

изображения и объединяются с общей частью в полностью связанном слое FC (Fully Connected).

1.2.3 Ассоциация обнаружений

Ассоциация обнаружений — процесс нахождения в текущем кадре объектов трекинга, находившихся на предыдущем. Это действие необходимо для создания трека объекта, а также переноса его уникального идентификатора. Хотя четкие обнаружения и верно подобранная метрика расстояния уменьшают сложность ассоциации, так или иначе произойдет столкновение с трудностями, которые будет возможно решить с помощью правильной стратегии ассоциации. Ранние методы ассоциации основаны на байесовской модели, которая ищет максимальную апостериорную оценку по условному распределению последовательных состояний среди всех наблюдений. Zhang и соавторы [9] используют модель сетевого потока для решения проблемы отслеживания множества людей. Их метод получил дальнейшее развитие путем изменения стратегии выбора обнаружения, метрики, структуры сети для обработки сложных факторов. Zamir и соавторы [15] представляют обобщенную проблему максимальной клики GMCP (Generalized Maximum Clique Problem). Dehghan и соавторы [14], основываясь на GMCP, получают все триплеты за один раз, чтобы избежать неправильного соответствия. Tang и соавторы [13] переводит проблему отслеживания нескольких людей в проблему наименьшего разреза. Кроме этого, существуют другие интересные алгоритмы, такие как алгоритм двух непересекающихся путей [13], динамическая графическая модель [22] и распространение меток по правилу минимакса [18] для задачи отслеживания нескольких человек.

1.3 Анализ трекинга на основе точек артикуляции

Трекинг на основе точек артикуляции представляет собой способ обнаружения и фильтрации артикуляций человеческого тела, то есть множества соединений точек на теле в один наблюдаемый объект. Данный трекер использует технологии глубокого машинного обучения для осуществления отсеивания неполных обнаружений. Если с помощью точек артикуляции не удастся создать полноценный образ человека, то это обнаружение называется неполноценным.

Внутреннее устройство трекера на основе точек артикуляции можно разделить на три модуля: модуль выборки, который реализует фильтрацию входящего потока обнаружений, модуль измерений, вычисляющий количество взаимосвязи наблюдений, а также ассоциативный модуль, соединяющий все наблюдения в одну траекторию, тем самым воссоздавая путь наблюдаемого объекта.

1.3.1 Анализ модуля выборки

Для обнаружения артикуляций используется алгоритм оценки поз [23], который позволяет локализовать точки наблюдений. Кроме того, появляется необходимость решить проблему появления кадров, на которых изображено большое количество людей в одинаковых позах, что приведет к перекрытию одного человека другим. Современные методы опираются на введение моделей, работающих по принципу сверху-вниз, или же снизу-вверх [24]. Эти модели хорошо избегают наложение объектов, что позволяет им с высокой точностью отделять даже близкие друг к другу. Кроме того, количество отслеживаемых может быть сколь угодно большим.

Современные модели нахождения артикуляций выделяют семнадцать ключевых точек на человеческом теле: нос (одна точка), глаза (2 точки), уши (2

точки), плечи (2 точки), локти (2 точки), запястья (2 точки), бедра (2 точки), колени (2 точки) и лодыжки (2 точки). На рис. 3 показано схематичное расположение данных точек на теле человека.

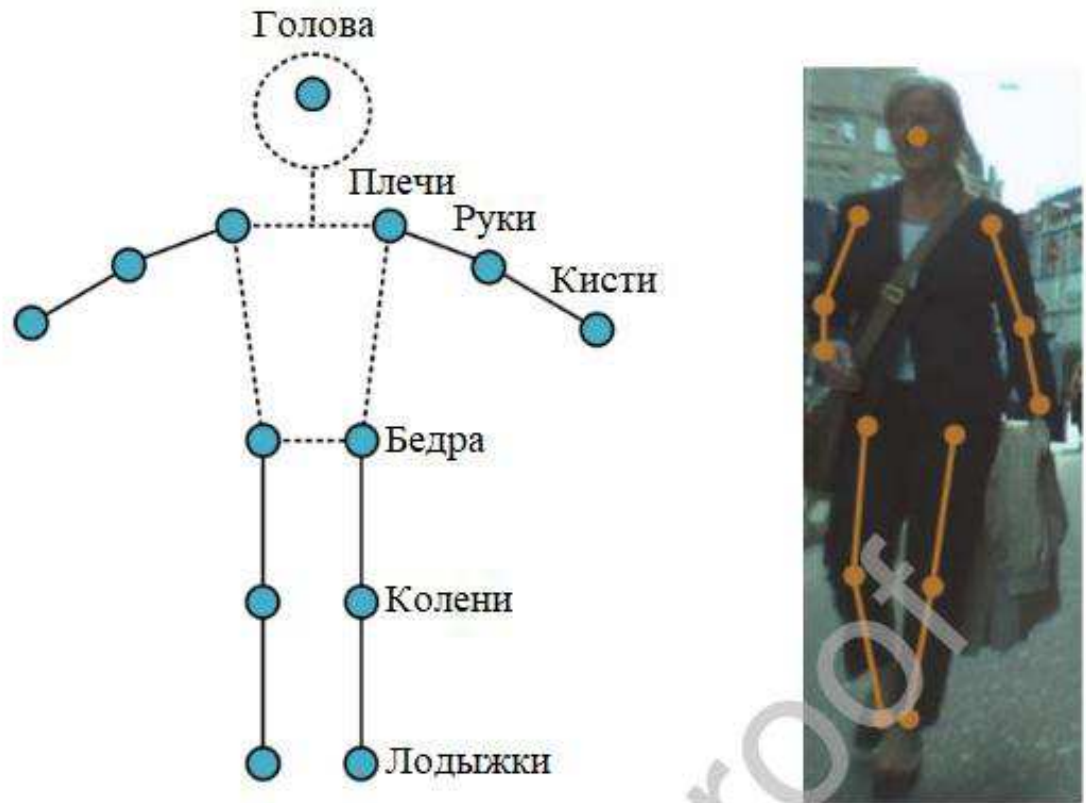


Рисунок 3 — Схематичное расположение точек артикуляции на теле человека

Имеем последовательность изображений на входе $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_n\}$ длины n . $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n\}$ — набор обнаруженных артикуляций для заданной последовательности. $\mathcal{A}_i = \{\mathcal{A}_i^1, \mathcal{A}_i^2, \dots, \mathcal{A}_i^{|\mathcal{A}_i|}\} (1 \leq i \leq n)$ представляет набор артикуляций i -го кадра, где $|\mathcal{A}_i|$ - число найденных артикуляций в кадре i и $|\mathcal{A}_i| = 0$, при $\mathcal{A}_i = \emptyset$) композиция семнадцати точек артикуляции i -го человека в кадре i .

Найденные на предыдущем шаге артикуляции используются для улучшения результата их нахождения. Набор $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n\}$ представляет обнаружения через n -ое количество кадров видео. $\mathcal{A}_i = \{\mathcal{A}_i^1, \mathcal{A}_i^2, \dots, \mathcal{A}_i^{|\mathcal{A}_i|}\} (1 \leq i \leq n)$ — обнаружения в i -ом кадре, где $|\mathcal{A}_i|$ является количеством

обнаружений в кадре \mathcal{B} и \mathcal{B}_i^a ($1 \leq i \leq \mathcal{B}_i^a$, $\mathcal{B}_i^a = 0$, при $\mathcal{B}_i = \emptyset$) — вектор, состоящий из позиции, размера и достоверности i -ой рамки вокруг наблюдения i -го кадра.



Рисунок 4 — Сравнение точности обнаружения людей методом артикуляции и обнаружений, идущих вместе с популярными наборами данных

Полученные обнаружения показаны на верхнем ряду изображений рис. 4, где видно, что некоторые обнаружения охватывают только часть человека, а некоторые люди так и не были выделены никаким образом. Голова человека в красном (левое изображение в первой строке) не находится в пределах диапазона выделяющей рамки, ограничивающего остальные части его тела, в другом случае (правое изображение в первой строке) рамка выделяет только ноги старика. На втором ряду картинок видно, что точки обнаружения артикуляции выделяют все тело наблюдаемого человека, что показывает высокую точность обнаружения по сравнению с обнаружениями, идущими вместе с набором данных. В нижнем ряду изображений видно, что современная стратегия выбора признаков исключает несколько рамок обнаружений, которые являются ошибочными обнаружениями, описанными ранее.

Из-за окклюзии, могут появляться обнаружения, в которых нет семнадцати точек артикуляции. Уравнение (1) показывает логику удаления подобных обнаружений, содержащих недостаточно точек артикуляции:

$$\Omega'_s = \{s \mid \text{count}(s) > \alpha, s \in \Omega_s\}, \quad (1)$$

где $\text{count}(s)$ означает количество точек артикуляции в s , а α — нижний предел количества точек артикуляции в найденных обнаружениях. Большинство обнаружений в Ω'_s могут выделить человека целиком. Однако, если просто отфильтровать пересечения обнаружений артикуляции и пометить их как окончательные, некоторые обнаружения вокруг одного и того же человека могут слиться в одно, что приведет к неправильной последовательности трекинга. Чтобы этого избежать, существует технология, аналогичная NMS, которую описывает уравнение (2):

$$\Omega'_s = \text{NMS}(\Omega_s, \alpha) = \{s \mid \text{count}(s) > \alpha\}, s \in \Omega'_s, s \in \Omega_s, \quad (2)$$

где $\alpha(s, \beta)$ — отношение количества точек артикуляции в s , окруженных рамкой обнаружения β , к общему количеству точек артикуляции в s , а β — нижний предел отношения, что упоминалось ранее. Функция $\alpha(s, \beta)$ в уравнении (2) помогает избежать проблему двойного трекинга, а функция $\alpha(s, \beta)$ позволяет избавиться от двойных обнаружений. Теперь мы имеем окончательный набор наблюдений β' , который показан в уравнении (3).

$$\beta' = \{\beta'_1, \beta'_2, \dots, \beta'_n\}$$

(3)

1.3.2 Анализ модуля измерений

Измерение обнаружений представляет собой двухэтапный процесс, при котором признаки каждого обнаружения извлекаются через CNN и используются для последующего вычисления расстояния.

Для выявления признаков каждого обнаружения используется метод повторной идентификации. Данный метод реализован с использованием полносвязной CNN с функцией потери триплетов. Функция потери триплетов может обеспечить распознавание мало различимых объектов, помогая выделять отдельных людей даже в многолюдных сценах. Выглядит это так: β_{α} используется вместо обнаружения β_{α} для представления одного из выбранных обнаружений последовательности β' . На вход CNN приходит изображение 288 x 144 пикселя, а на выходе имеется 128-мерный вектор признаков. После чего нужно изменить размеры вектора признаков каждого обнаружения $\beta_{\alpha} \in \beta'$ и подать его на вход CNN. Вектор β_{α} является выходным вектором признаков, полученным из β_{α} .

Отслеживание нескольких человек может быть сконструировано как модель направленного ациклического графа $G = (V, E)$ где V — множество

всех вершин, а E — множество всех ребер. $\square_{\square} \in \square$ представляет обнаружение \square_{\square} в наборе \square' , где $\square_{\square\square} \in \square$ — грань, которая соединена в пару обнаружений \square_{\square} и \square_{\square} . Существует два ограничения, обеспечивающих правильность построенной модели для слежения за множеством людей. Поскольку существует не более одного обнаружения для каждой цели в кадре, каждые два обнаружения в одном и том же кадре обязательно имеют разные метки, что означает невозможность связать их одним ребром. Кроме этого, пары обнаружений в кадрах, расположенных друг рядом с другом, имеют гораздо более сильную связь, чем обнаружения, удаленные друг от друга. Рассмотрим уравнение (4)

$$\square = \{\square_{\square\square} | 0 < \square_{\square} - \square_{\square} < \square\},$$

(4)

где \square_{\square} — временная метка (номер кадра) обнаружения \square_{\square} . На временной шкале, обнаружение \square_{\square} наблюдается позже, чем \square_{\square} (то есть кадр, в котором появляется обнаружение \square_{\square} , находится после кадра, в котором появляется обнаружение \square_{\square}) и временной интервал не превышает \square . \square определяет диапазон кадров, который нужно покрыть при построении граней, которые позволят справиться с окклюзией, а также поможет избежать связывания двух обнаружений различного характера, расположенных на большом расстоянии. Евклидово расстояние $\square(\square_{\square}, \square_{\square})$ между обнаружениями \square_{\square} и \square_{\square} измеряется по формуле (5):

$$\square(\square_{\square}, \square_{\square}) = \sqrt{\sum_{i=1}^{128} (\square_{\square i} - \square_{\square i})^2},$$

(5)

где \square_{\square} — 128-мерный вектор признаков, полученный из обнаружения \square_{\square} на предыдущем шаге.

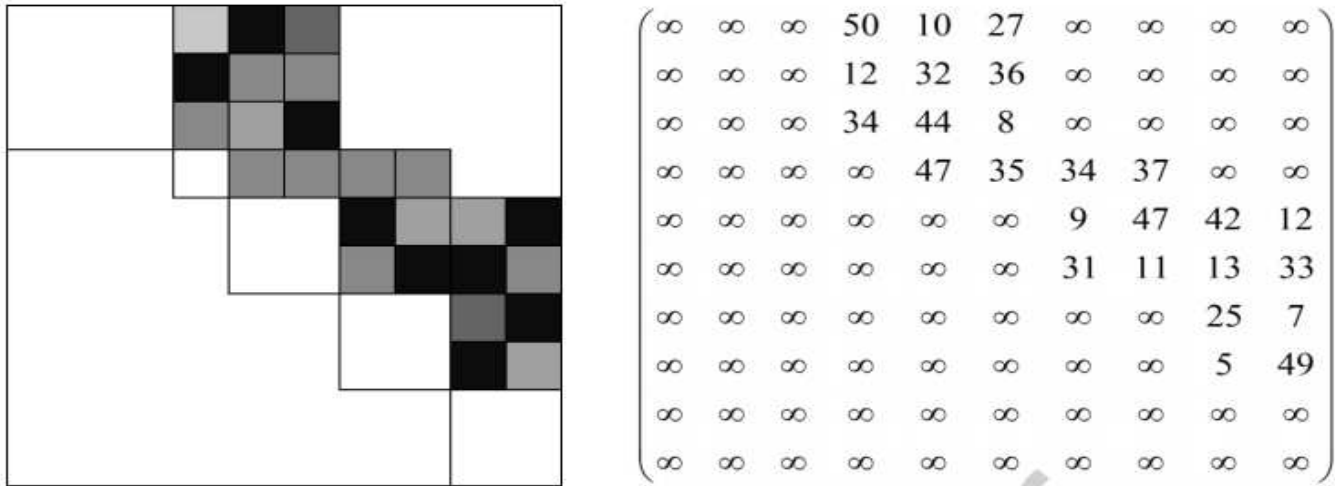


Рисунок 5 — Часть изображения вместе с его матрицей расстояний $L = l(d_i, d_j)$

На рис. 5 приведен простой пример матрицы расстояний, где $n = 2$ — количество кадров в последовательности, $\mathcal{T} = 2$ — временной интервал. На изображении слева, самый большой квадрат представляет всю матрицу, в то время как маленькие квадраты по диагонали представляют расстояния между обнаружениями в одном и том же кадре. Если клетка (i, j) окрашена в чистый белый цвет, значение матрицы $l(d_i, d_j)$ справа равно бесконечности, что означает отсутствие грани между обнаружениями d_i и d_j . Только когда клетка (i, j) затемнена, значение матрицы $l(d_i, d_j)$ не равно бесконечности, что означает существование взвешенного ребра $l(d_i, d_j)$, соединяющего обнаружения d_i и d_j . Чем больше значение $l(d_i, d_j)$, тем более вероятно соответствие обнаружения.

1.3.3 Анализ модуля ассоциаций

В данном разделе представлены минимаксная операция [18], а также стратегия сшивания. Минимаксный алгоритм минимизирует максимальное расстояние между триплетами, а стратегия сшивания связывает триплеты в траектории наблюдения.

Минимаксом называется правило принятия решений, используемое в теории принятия решений, теории игр, статистике и философии, направленное на минимизацию потерь в конкретных наихудших ситуациях. В задаче

трекинга, в отличие от традиционного алгоритма кратчайшего пути, известного как алгоритм Дейкстры, минимизирующего стоимость всего пути через минимизацию пути отдельных элементов графа, алгоритм минимакса минимизирует стоимость, определенную максимальным расстоянием на каждом пути, сохраняя при этом небольшую вариативность, которая позволяет избегать неправильных совпадений, что более разумно для отслеживания большого количества людей. Сравнение двух алгоритмов продемонстрировано на рис. 6. Числа у краев изображений — расстояния между обнаружениями. Цветами обозначены разные направления движения.

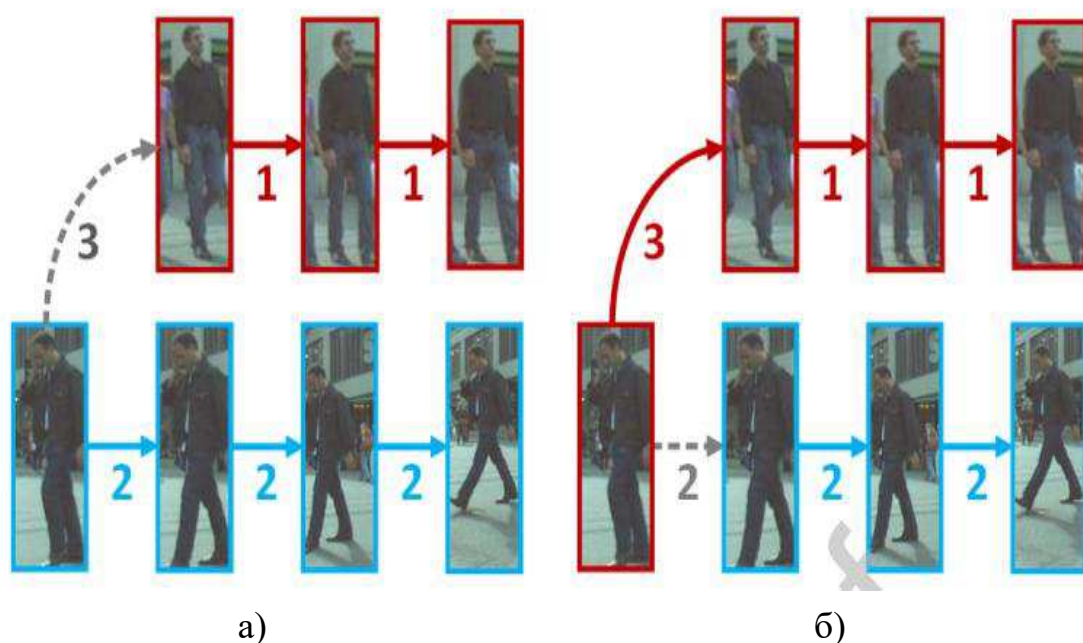


Рисунок 6 — Сравнение алгоритмов а) — минимакса и б) — Дейкстры

Если в процессе минимаксной операции несколько путей имеют высокий вес, это означает принадлежность обнаружений к нескольким объектам, но по какой-то причине соединившиеся вместе. Перед применением минимакса, ограничивающий максимальное расстояние между объектами параметр \square , используется для исключения границ объектов с расстоянием, превышающим значение этого параметра.

После применения метода минимакса остаются наборы путей, которые можно рассматривать как потенциально финальные траектории движения.

Однако, из-за осложняющих факторов, таких как вращение человека вокруг своей оси, различия в распознавании одного и того же человека могут быть серьезными, что непременно приведет к разрывам в траектории слежения. Траектории, сгенерированные после применения минимакса, называются триплетами траекторий, дабы не путать их с окончательными.

Метод сшивания необходим для соединения триплетов траекторий воедино, после чего результат можно назвать окончательной траекторией движения объекта. Положим, что $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_n\}$ представляет набор триплетов траекторий, сгенерированных методом минимакса. Для каждого триплета $\Omega_i = \{\Omega_i^1, \Omega_i^2, \dots, \Omega_i^m\} (1 \leq i \leq n)$, ищут наиболее похожий триплет $\Omega_j = \{\Omega_j^1, \Omega_j^2, \dots, \Omega_j^m\} (1 \leq j \leq n)$, причем $(i > j)$, по формуле (6), после чего соединяют их:

$$\Omega = \Omega \cup \Omega_j \quad \{\Omega_i, \Omega_j\} \text{ если } \rho(\Omega_i, \Omega_j) < \rho, \quad i < j \leq n, \quad \Omega_i \in \Omega, \quad \Omega_j \in \Omega, \quad (6)$$

где ρ — расстояние между точками артикуляции, позволяющее определить, принадлежат ли несколько обнаружений одному и тому же человеку. К сожалению, возможно соединение похожих триплетов A_j , неминуемо приводящее к неверным результатам. Чтобы этого избежать, стратегия сшивания применяется в нескольких направлениях — прямом и обратном. На рис. 7 показано выполнение прямого и обратного процессов сшивания.

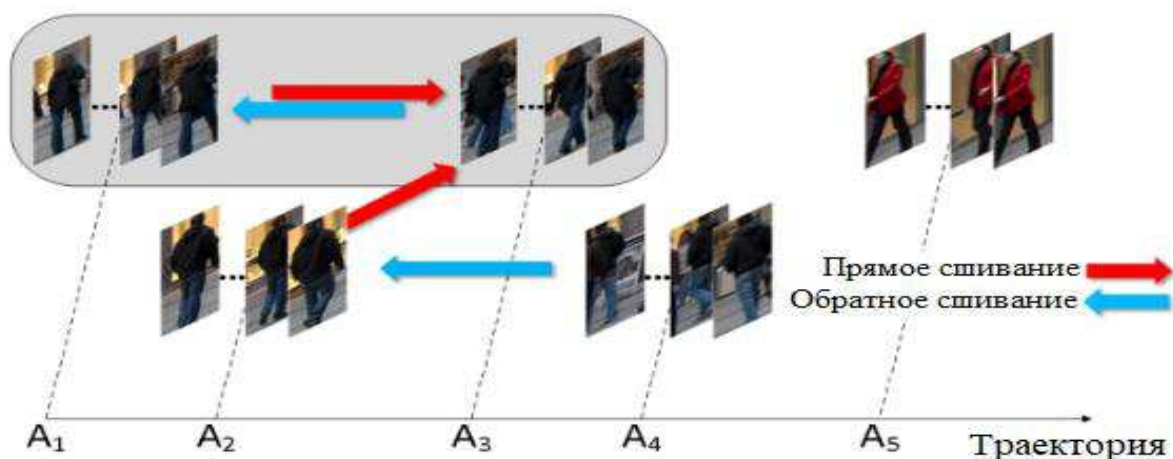


Рисунок 7 — Схематическое объяснение стратегии сшивания

Триплет A_1 представлен женщиной, делающей шаг, A_2 — мужчина, совершающий такое же действие и A_3 — женщина делает следующий шаг. Одежда этих людей на вид схожа, что привело к соединению триплетов A_1, A_2 и A_3 вместе при прямом сшивании. Если посмотреть на процесс в обратном направлении, A_3 необходимо найти более подходящий триплет, нежели A_1 и A_2 . Наконец, триплеты A_1 и A_3 соединяются методом сшивания, так как и прямой и обратный проходы алгоритма выявили их схожесть. Данный метод позволяет с большой вероятностью избежать нежелательных ситуаций объединения не похожих друг на друга триплетов.

2 Разработка системы автоматического трекинга людей по видеоданным

Предполагается общий сценарий отслеживания, когда камера не откалибрована и нет информации о движении отдельного человека. Данные обстоятельства усложняют трекинг, однако, на сегодняшний день, они являются обычным явлением [25].

Основная практическая задача системы автоматического трекинга состоит в отображении вокруг людей ограничивающей рамки, а также их уникальных идентификаторов, а также сопровождение этой рамкой людей в движении. Для этих целей необходимо использовать метод, позволяющий запоминать внешние очертания людей, за которыми ведется наблюдение. В

настоящей работе применена нейросетевая модель, которая реализует алгоритм глубокой сортировки.

2.1 Алгоритм глубокой сортировки и нейронная сеть

Остаточная сверточная нейронная сеть — RNN (Residual Neural Network), используемая в работе, реализует алгоритм глубокой сортировки. Это означает, что она вычисляет расстояние и скорость перемещения людей, а также особенности их внешности. Нейронная сеть осуществляет это при помощи вычисления глубоких признаков для каждой рамки наблюдения, а также нахождения сходства между глубокими признаками, влияющими на логику слежения.

Вычисление глубоких признаков в работе возлагается на нейросетевую модель. Подход вычисления глубоких признаков мотивирован необходимостью осуществлять нахождение и изучение внешних признаков людей в режиме реального времени, а также принимать решения в сложных динамических ситуациях, например в случаях появления окклюзии, когда один человек перекрывает собой другого. Пример сложной сцены представлен на рис. 8.

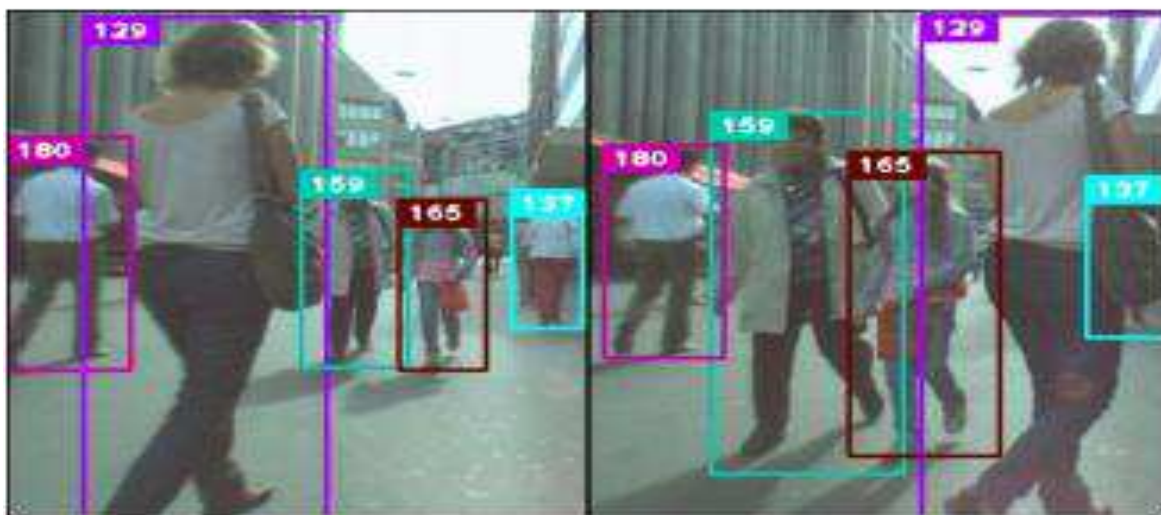


Рисунок 8 — Отслеживание людей на сложных сценах с окклюзией

На сцене, люди с идентификаторами 159 и 165 были временно перекрыты женщиной с идентификатором 129. В последующих кадрах видно положительное влияние глубоких признаков на процесс ассоциации наблюдаемых — идентификаторы были переназначены верно.

Используемая в работе остаточная CNN была обучена на крупных наборах данных, таких как MARS [26] и Market1501 [27]. Оба датасета в сумме содержат более миллиона изображений 1261 уникальных пешеходов, что делает эту модель подходящей для оценки глубоких метрик людей на сцене. Архитектура данной модели представлена в табл. 1.

Таблица 1 — архитектура нейросетевой модели для вычисления глубоких признаков

Название слоя	Размерность/Страйд	Размерность выхода
Сверточный	3 x 3/1	32 x 128 x 64

Окончание таблицы 1

Сверточный	3 x 3/1	32 x 128 x 64
Слой пулинга по максимуму	3 x 3/2	32 x 64 x 32
Остаточный	3 x 3/1	32 x 64 x 32
Остаточный	3 x 3/1	32 x 64 x 32
Остаточный	3 x 3/2	64 x 32 x 16
Остаточный	3 x 3/1	64 x 32 x 16
Остаточный	3 x 3/2	128 x 16 x 8
Остаточный	3 x 3/1	128 x 16 x 8
Полносвязный	—	128

Слой пакетной нормализации, а также L2-нормализации	—	128
---	---	-----

Модель обладает следующими характеристиками: это широкая нейронная сеть с двумя сверточными слоями, одним пулиновым слоем, за которым следует шесть остаточных, после которых следует один полносвязный слой, а также заключительный слой пакетной (батч) и L2 нормализациями.

Сверточные слои необходимы для выполнения операции свертки на изображении. Результатом свертки будут являться изображения, на которых изображены небольшие части исходных данных каждой картинке. Эти данные в дальнейшем используются для улучшения точности отображения рамки наблюдения поверх изображений.

Слой, использующий функцию максимума (или слой Max Pool), используется для сжатия исходных изображений, полученных при помощи операции свертки, а также для ускорения вычислений, вследствие ограниченности вычислительных ресурсов.

Особенностью использованной сверточной нейронной сети является применение остаточных слоев. Дело в том, что во время обучения сверточных нейронных сетей могут возникнуть проблемы затухающих градиентов (производных от ошибки, распространяемых по сети в обратном порядке для корректировки весов), когда, проходя через большое количество слоев, значения этих градиентов становятся незначительно малыми, либо же вовсе исчезают, что снижает качество обученной модели. Остаточные слои обнаруживают затухание и позволяют пропустить несколько слоев ради сохранения имеющегося результата, позволяя создавать очень глубокие нейронные сети высокой эффективности.

Основной принцип работы нейросетевой модели заключается в нахождении при помощи обнаружений, которые находятся в массиве с последовательностью изображений людей, как это показано на рис. 9.

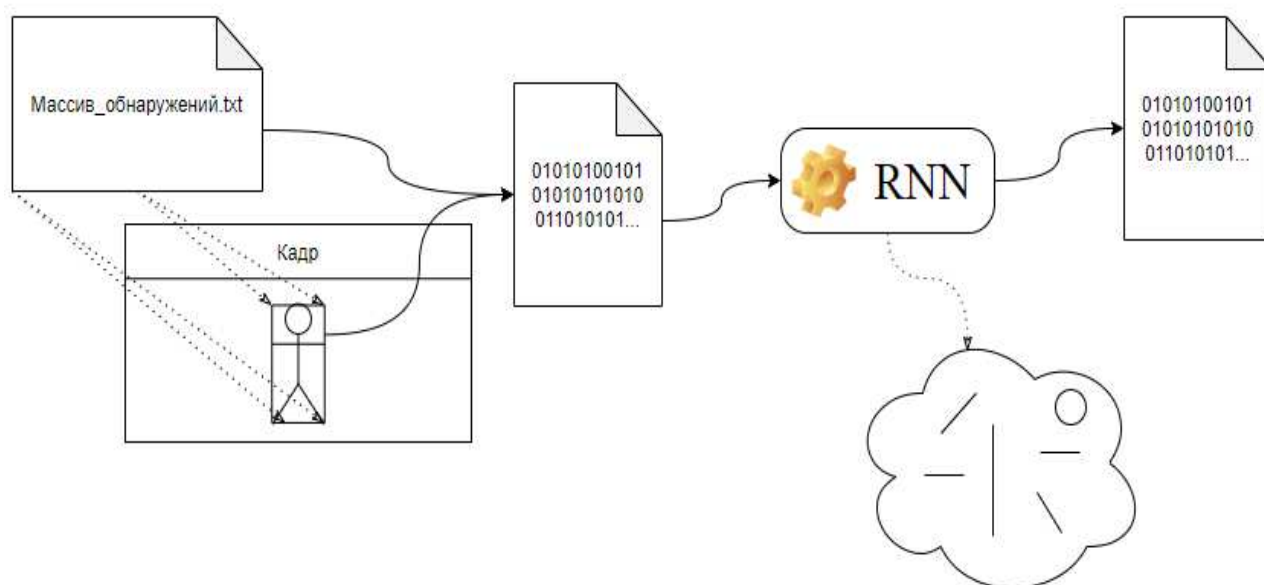


Рисунок 9 — Принцип работы RNN

В полносвязном слое вычисляется 128-мерный выходной вектор признаков, записанный в файл с расширением *.npy — массив данных NumPy. Последние слои нейросетевой модели являются слоями батч-нормализации и L2 нормализации. Они подгоняют результат для удобного вычисления косинусного подобия — основного преимущества данной нейронной сети для задачи извлечения глубоких признаков. В общей сложности сеть имеет более $2^{8000000}$ параметров.

2.2 Процесс трекинга людей

Первым делом системе необходимо сгенерировать файл наблюдений, в котором будет запечатлено местоположение каждого человека на сцене, а также соответствующий ему вектор признаков, полученный при помощи нейросетевой модели. Результатом генерации будет файл, названный в честь папки, из которой брались необходимые элементы. Этот файл будет иметь расширение *.npy. Далее запускается трекер. Трекер использует

сгенерированный ранее файл и начинает сопоставление данных, взятых из файла с расширением *.пру с каждой картинкой последовательности так, что в окне просмотра происходит наложение результатов трекинга на исходные картинки. Описание этого процесса представлено на рис. 10.

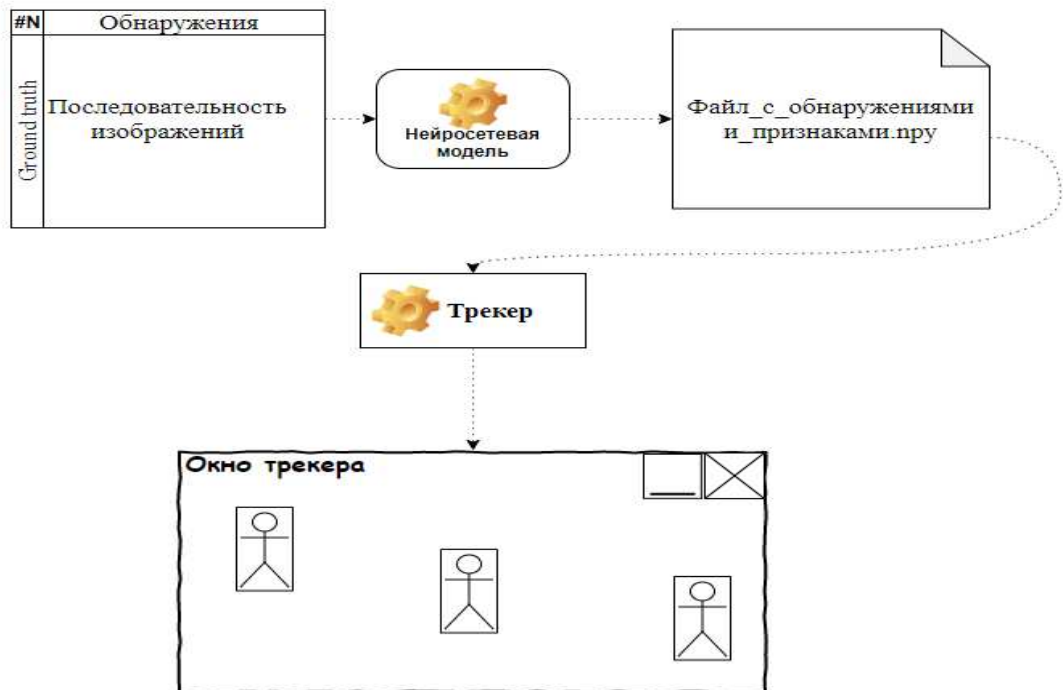


Рисунок 10 — Диаграмма активности работы системы автоматического трекинга

Для каждого k - наблюдения система подсчитывает количество кадров с момента последнего успешного сопоставления измерений \square_{\square} . Счетчик увеличивается в момент прохождения наблюдения через фильтр Калмана и сбрасывается в нуль, если сопоставление прошло успешно. Наблюдения, превышающие максимальное время жизни, считаются неактуальными и удаляются из сета наблюдений. Новые предположения о передвижении создаются для каждого обнаружения, которое не может быть ассоциировано с существующей траекторией движения. Эти новые предположения относятся к неподтвержденным в течение первых трех кадров. В промежутке между этими кадрами ожидается подтверждение их актуальности через сопоставление с предыдущими. Те предположения о перемещении, которые превысили свой жизненный лимит, удаляются.

2.3 Проблемы трекинга

Процесс трекинга сопряжен с различными затрудняющими факторами. Эти факторы влияют на границы рамок наблюдений, скорость обработки последовательностей изображений, связывание уникального идентификатора каждого человека на всем протяжении трекинга.

2.3.1 Конкретизация проблем и способов их решения

В работе были выявлены следующие значительные проблемы: плохая калибровка камер наблюдения, отсутствие информации о передвижении, сложность ассоциации, возникновения окклюзий, а также многократное распознавание. Проблему плохой калибровки способна решить описанная в 2.1 нейросетевая модель. Другие же проблемы было решено рассмотреть более детально.

Многократное распознавание. При обнаружении различного рода объектов, произойдет столкновение с одной очень важной проблемой распознавания — многократный отклик модели распознавания. Дело в том, что границы распознавания образов нечеткие, что приводит к наложению результатов распознаваний одного на другой. В рамках экспериментальных исследований эта проблема может проявить себя во время создания целевого файла распознавания, так как большое количество людей в кадре будет также генерировать огромное количество рамок (границ распознавания). Рис. 11 наглядно демонстрирует эту проблему: изображение слева было подвержено многократному отклику модели распознавания, что увеличило количество рамок наблюдения, справа представлен результат, к которому необходимо стремиться.

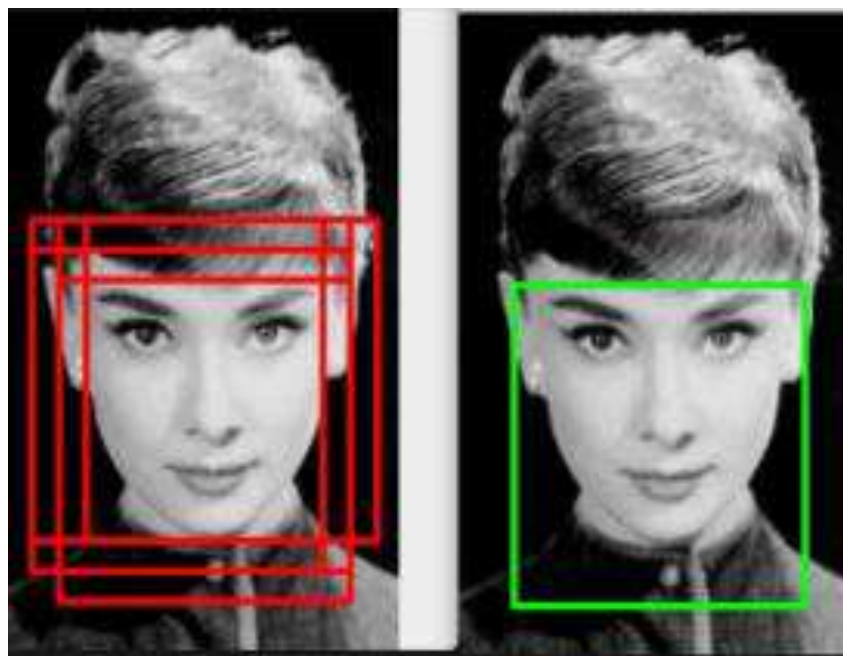


Рисунок 11 — Пример проблемы многократного распознавания

Если взять во внимание практическую сторону вопроса, то сам факт многократного распознавания одного объекта в кадре не является проблемой, а наоборот подтверждает эффективность модели. Проблемой в этом случае является распознавание объекта в той области, где он отсутствует. Для решения проблемы многократного распознавания в работе использован метод подавления немаксимумов или NMS (Non-Maximum Suppression).

Отсутствие информации о передвижении. Процесс трекинга включает в себя наблюдение за объектом, поведение которого непредсказуемо. Это вызывает трудность, суть которой в невозможности плавного следования рамки наблюдения за человеком, отчего каждая рамка будет мигать. Таким образом, возникает необходимость заранее узнать приблизительное направление движения, чтобы трекинг получился как можно плавнее. Из-за отсутствия данных о передвижении людей на сцене, было принято решение использовать фильтр Калмана. Основное его преимущество заключается в хороших прогнозах, основанных на использовании неточных входных данных.

Окклюзии. В процессе трекинга, человек может на некоторое время покинуть сцену. Это может произойти по причине окклюзии, то есть один человек может полностью или частично закрыть в кадре другого. Окклюзия

проявляется в моменты, когда один человек загораживает другого. Это проблема, которая не только помешает отобразить рамку наблюдения, но и заставит систему выдать новый уникальный идентификатор, так как системе неизвестно, что наблюдаемый человек присутствовал на сцене ранее. Для борьбы с окклюзией было принято решение использовать такие методы, как центроиды рамок наблюдений, фильтры Калмана, а также каскад сопоставлений.

Ассоциации. Решая проблему отсутствия информации о передвижении, была выявлена следующая проблема: предсказания положения человека на сцене при помощи фильтра Калмана точны недостаточно для того, чтобы не замечать ошибочное расположение рамок наблюдений. Исходя из этого было принято решение использовать центроиды рамок наблюдений и каскад сопоставлений, имеющие не самую трудную реализацию, а также позволяющие решать проблему ассоциаций в режиме реального времени.

2.3.2 Решение проблем

Проблемы трекинга людей по видеоданным и способы их решения представлены на рис. 12.

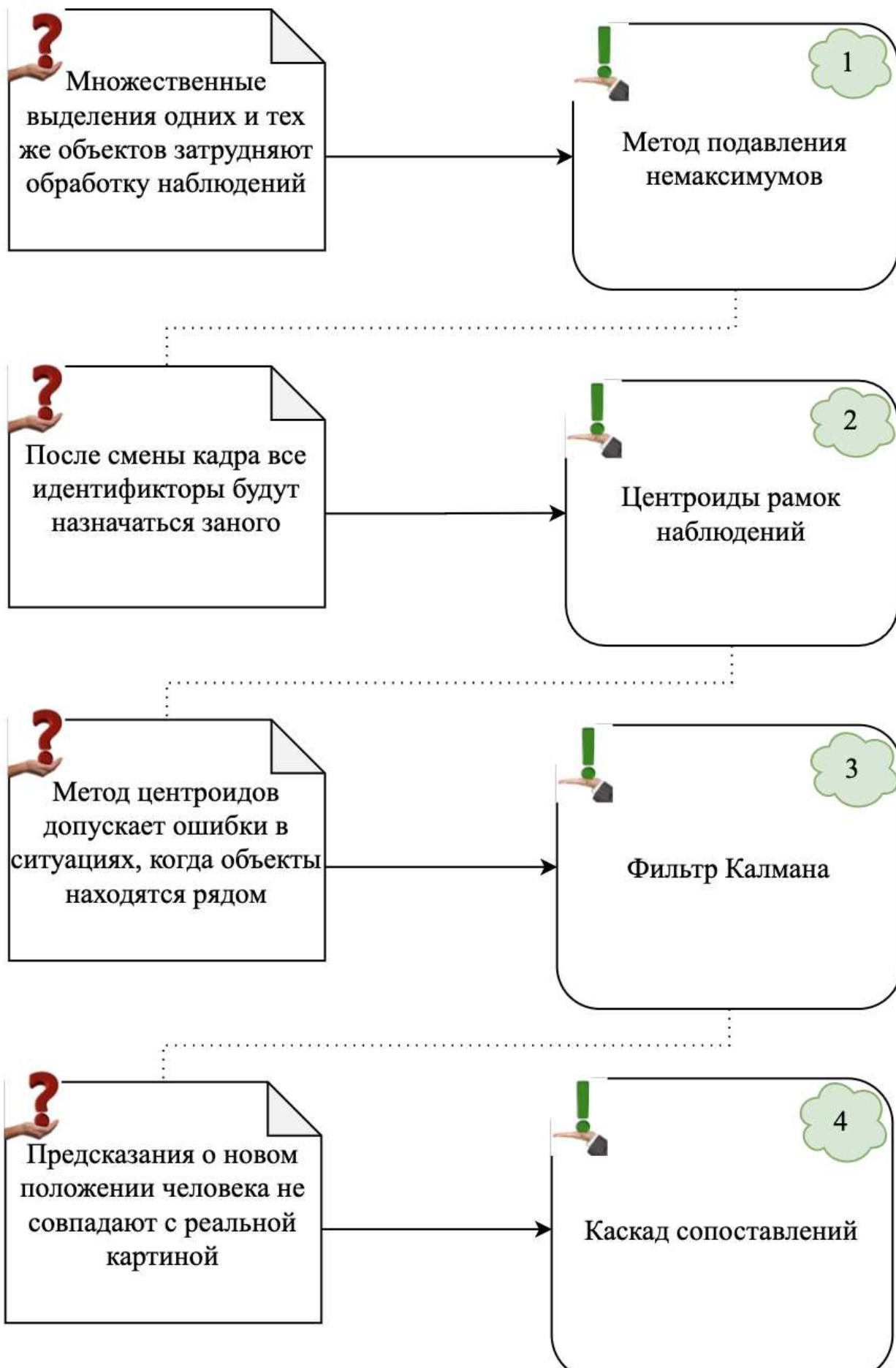


Рисунок 12 — Проблемы и способы их решения

Для решения вышеописанных проблем, возникающих при трекинге людей, использовались следующие методы:

- метод подавления немаксимумов (NMS — Non-Maximum Suppression);
- центроиды рамок наблюдений;
- фильтр Калмана;
- каскад сопоставлений.

Стоит отметить, что некоторые методы решали несколько проблем, или же являлись дополнениями к другим. Так, например, использование фильтра Калмана позволяет уточнить результаты работы метода центроидов, который работает неверно в случаях близкого расположения людей на сцене.

2.3.2.1 Метод подавления немаксимумов

Одним из методов предотвращения наложения рамок распознавания объектов, является оператор Кэнни (метод подавления немаксимумов) [32]. Основные принципы этого метода:

- отношение полезной информации к шуму должно быть как можно больше;
- определение границ объекта должно быть как можно точнее;
- если граница была определена, она должна существовать в единственном экземпляре.

Основа метода заключается в поиске максимальной области покрытия найденного отрезка изображения, а также выведение из множества наложенных друг на друга выделяющих рамок в одну большую. Однако, при всех своих плюсах, метод подавления немаксимумов требует значительных вычислительных затрат, так как необходимо обработать более тысячи изображений за короткий промежуток времени. Поэтому, для решения этой проблемы использована усовершенствованная реализация метода Кэнни на языке Python. Реализация классической версии метода требовала

дополнительного внутреннего цикла `for` для вычисления размера наибольшей выделяющей рамки и вычисления отношения перекрывающихся областей. Для ускорения алгоритма этот цикл был заменен на векторизованный код. Функции `pr.maximum` и `pr.minimum` позволяют совершить все рутинные вычисления над массивом рамок выделения за одну операцию, уменьшая количество необходимых вычислений для нахождения максимальных и минимальных значений.

Следующая проблема, которые необходимо решить — проблема повторной идентификации при появлении окклюзий. Нахождение решения данной проблемы — важная задача. В обычной ситуации, человек, который был перекрыт, получит новый идентификатор в системе и его трек будет просчитываться заново. Решить проблему повторной идентификации сможет метод центроидов.

2.3.2.2 Метод центроидов рамок наблюдений

Для обеспечения наблюдения за объектами интереса в реальных приложениях слежения за объектами необходимо заранее получить информацию о нахождении каждой рамки наблюдения, поэтому мы должны «подружить» нейросетевую модель, создающую прогнозируемую область наблюдаемого человека, а также трекера, который будет «вести» наблюдаемые объекты до их выхода из кадра. Предположим, что имеется информация об ограничивающих рамках для всех людей в кадре. В процессе трекинга, получая информацию о рамке наблюдения в кадре 1, затем идентифицируя наблюдаемого при помощи назначения ID каждому наблюдаемому, необходимо решить каким образом передавать этот ID для последующих кадров.

Присвоение идентификаторов производится при помощи центроидов рамок наблюдений. В сравнении с другими аналогичными методами, такими как метод наименьших квадратов [28] или использование сиамских нейронных

сетей [17, 18], метод использования центроидов рамок наблюдений позволяет достичь приемлемой эффективности без значительных вычислительных затрат. Центроиды вычисляются для каждого прямоугольника, характеризующего границы распознавания объекта, в первом кадре. При переходе к следующему кадру видеопоследовательности происходит вычисление новых центроидов рамок наблюдений людей для каждого человека, присутствующего в кадре. При этом новые центроиды сравниваются с уже имеющимися центроидами с использованием меры относительных расстояний от местоположения предыдущих, таким образом передавая ID наблюдаемого от одного кадра к другому. Основное предположение состоит в том, что центроиды от кадра к кадру будут двигаться на небольшое расстояние. Такой подход работает очень эффективно, если центроиды находятся на разном расстоянии друг от друга. Однако, метод не сработает, если люди в кадре находятся на небольшом друг от друга расстоянии, что приведет к энтропии переназначения ID объектов внутри последующих кадров. Для решения этой проблемы в работе использован фильтр Калмана. Фильтр Калмана использует неточные метрики для определения последующего состояния объекта, исходя из предыдущего.

2.3.2.3 Фильтры Калмана

Фильтры Калмана являются надстройкой над методом на основе центроидов, которая позволит трекеру не путать уникальный идентификатор людей на последующих кадрах. Эти фильтры позволяют моделировать трекинг на основе положения и скорости, прогнозируя возможное местонахождение человека. Моделирование происходит при помощи гауссианы, или функции плотности нормального распределения Гаусса. Когда функция получает на вход параметры, она использует вероятности для присвоения нового значения, либо обновления существующего. Этот подход может работать в режиме реального времени, что представляет высокую ценность для осуществления трекинга.

Также метод Кэнни может быть использован для выполнения вспомогательных операций в любой области, где существует определенная информация о какой-либо динамической системе, а также сделать предположение, что система будет делать дальше. Кроме того, он учитывает неопределенность, вносимую окружающей средой, при помощи корреляций различных значимых величин. Программная реализация метода Кэнни не требует много памяти — необходима только информация о последнем состоянии.

Для использования фильтра Калмана предполагается, что переменные величины являются случайными и имеют нормальное распределение, заданное по формуле (6):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (6)$$

Каждая переменная имеет среднее значение μ , являющееся центром случайного распределения (и его наиболее вероятным состоянием), и дисперсию, которая выражена неопределенностью.

Важно отметить, что на момент применения фильтров, неизвестно, какое из состояний «реально», на что функции прогнозирования все равно. Она отработает при любых значениях состояния системы и создаст новый прогноз. Функция прогнозирования берет каждую точку первоначальной оценки и перемещает ее в новое предсказанное место, куда система будет перемещаться, если эта первоначальная оценка была действительной.

Существуют изменения, которые не связаны с самим состоянием объекта — внешние силы. Например, если система моделирует движение поезда, оператор поезда может нажать на дроссель, тем самым ускорив его. Если известна дополнительная информация о том, что происходит вокруг, возможно добавить эту информацию в вектор, обработать его и внедрить в прогноз в качестве поправки. Зная влияние внешних и внутренних сил, может показаться,

что проблема решена. К сожалению, существуют такие внешние факторы, о которых ничего не известно. Более того, эти факторы могут повлиять на систему в случайный промежуток времени. Например, если при отслеживании квадрокоптера, происходит удар о встречный поток ветра. Их влиянием можно было бы пренебречь, однако нет возможности предугадать степень их влияния, а значит, с ними придется считаться. Таким образом, новая оценка предсказывается благодаря предыдущей, с некоторой неопределенностью, добавляемой окружающей средой.

Помимо измерения влияния внешних сил, необходимо иметь прибор, позволяющий измерить положение и скорость. Не имеет разницы то, какие параметры они измеряют. Каждый датчик косвенно говорит о текущем состоянии параметра. Единицы измерения и масштаб датчиков могут не совпадать с соответствующими параметрами отслеживаемого состояния. Фильтры Калмана отлично подходят для работы с зашумленными показаниями датчиков, так как датчики сами по себе являются ненадежными, и каждое состояние в первоначальной оценке может привести к некоторому разбросу.

Итак, создается ситуация, в которой имеется несколько Гауссовых капель: одна описывает среднее значение преобразованного показания, а вторая — фактическое показание датчика, которое было считано. Далее необходимо согласовать предположение о будущих показаниях при помощи предсказанного состояния, с другим предположением, основанным на показаниях датчика, которые наблюдаются. Область наложения, получившаяся после перемножения гауссовых капель, является областью наиболее вероятных состояний. Данная область является наиболее вероятной оценкой, по сравнению с предыдущими.

Теперь, когда можно выяснить примерное направление движения человека, встает вопрос о том, каким образом проводить ассоциацию между предсказаниями фильтра Калмана и реальным положением каждого человека в кадре.

2.3.2.4 Каскад сопоставлений

Проблема ассоциации предположений возникает в тот момент, когда приходится ассоциировать состояние наблюдения, предсказанное при помощи фильтра Калмана и только что появившееся. Обычно эту проблему решают при помощи венгерского алгоритма. Для этой задачи информацию о движении и внешнем виде интегрировалась посредством комбинации двух соответствующих метрик.

Для объединения информации о движении я использую квадрат расстояния Махаланобиса между предсказаниями фильтров Калмана и только что поступившими:

$$\sigma_{\square, \square}^{(I)}(\square, \square) = (\square_{\square} - \square_{\square})^T \square_{\square}^{-I} (\square_{\square} - \square_{\square}), \quad (7)$$

где $(\square_{\square}, \square_{\square})$ — пространство измерений проекции распределения \square -го наблюдения, а \square_{\square} — определение \square -й рамки наблюдения. Расстояние Махаланобиса учитывает неопределенность оценки состояния, измеряя количество стандартных отклонений обнаружения, находящихся далеко от среднего его расположения. Помимо всего прочего, возможно исключить маловероятные ассоциации, установив пороговое значение расстояния Махаланобиса с 95% доверительным интервалом, вычисленным при помощи обратного распределения \square^2 . Обозначим это как индикатор

$$\sigma_{\square, \square}^{(I)} = I[\sigma_{\square, \square}^{(I)}(\square, \square) \leq \sigma_{\square, \square}^{(I)}], \quad (8)$$

который равен 1, если связь между \square -м путем и \square -м обнаружением допустима. Для четырехмерного пространства измерений соответствующий порог распределения Махаланобиса равен $\square^{(1)} = 9,4877$.

Но расстояние Махаланобиса как метрика может давать неверный результат в случаях движения камеры наблюдения, поэтому было решено ввести еще одну метрику. Для каждой рамки обнаружения я вычисляю определитель \square_{\square} , где $\|\square_{\square}\| = 1$. Далее сохраняем галерею $\square_{\square} = \{\square_{\square}^{(\square)}\}_{k=1}^{L_k}$ из последних $\square_{\square} = 100$ связанных дескрипторов для каждого наблюдения k . Затем, вторая метрика измеряет наименьшее косинусное расстояние между \square -м наблюдением и \square -м обнаружением в момент появления:

$$\square^{(2)}(\square, \square) = \min_{\square_{\square}^{(\square)} \in \square_{\square}} \{1 - \square_{\square}^T \square_{\square}^{(\square)}\}, \quad (9)$$

Итак, снова вводится двоичная переменная для утверждения возможной ассоциативности согласной этой метрике

$$\square_{\square, \square}^{(2)} = I[\square^{(2)}(\square, \square) \leq \square^{(2)}], \quad (10)$$

после чего находится подходящий порог для этого показателя в отдельном обучающем наборе данных. На практике я применяю предварительно обученную CNN для вычисления дескрипторов появления рамок обнаружения. Архитектура этой сверточной нейронной сети описана далее.

В совокупности обе метрики дополняют друг друга, обслуживая различные аспекты проблемы назначения. С одной стороны, расстояние Махаланобиса предоставляет информацию о возможных местоположениях объектов на основе движения, которые особенно полезны для краткосрочных

прогнозов. С другой стороны, косинусное расстояние учитывает информацию о внешнем виде, которая особенно полезна для восстановления личности после длительных окклюзий, когда движение менее различимо. Чтобы построить проблему ассоциации, объединим обе метрики, используя взвешенную сумму

$$\square_{\square, \square} = \square \square^{(1)}(\square, \square) + (1 - \square) \square^{(2)}(\square, \square),$$

(11)

где ассоциация является допустимой, если она находится в области гейтинга обеих метрик:

$$\square_{\square, \square} = \prod_{m=1}^2 \square_{\square, \square}^{(\square)}$$

(12)

Влияние каждой метрики на совокупную стоимость ассоциации можно контролировать с помощью гиперпараметра λ . В ходе экспериментов было обнаружено, что установка $\lambda = 0$ является разумным выбором, когда камера имеет существенное движение. В этом параметре в термине стоимости ассоциации используется только информация о внешнем виде.

Тем не менее, ворота Махаланобиса по-прежнему используются для игнорирования невыполнимых назначений, основанных на возможных местоположениях объектов, определяемых фильтром Калмана.

Вместо того, чтобы решать задачу сопоставления измерения и пути, было решено ввести каскад, решающий несколько подзадач. Чтобы мотивировать данный подход рассмотрим ситуацию: когда объект подвергается окклюзии продолжительный промежуток времени, каждая итерация прохождения через фильтр Калмана увеличивает энтропию возможного места нахождения этого объекта. Необходимо учитывать эту энтропию во время использования ассоциативной метрики. С другой стороны, когда два разных обнаружения конкурируют за назначение целевого ID, расстояние Махаланобиса способствует еще большей неопределенности, так как быстро смещает значение

стандартного отклонения любого обнаружения по отношению к среднему значению прогнозируемого направления.

Такое поведение не является желательным, потому что приводит к плохим предсказаниям направления движения объекта. Поэтому было решено ввести каскад сопоставлений, который отдает приоритет более часто наблюдаемым объектам.

Алгоритм работы каскада сопоставлений следующий:

- Есть входные данные: массив индексов треков, массив индексов обнаружений и максимальный возраст среди всех треков;
- Вычисляется матрица стоимости $C = [c_{ij}]$ по формуле 7;
- Вычисляется матрица ассоциаций $V = [v_{ij}]$ по формуле 8;
- Инициализируется пустой набор совпадений μ ;
- Инициализируется набор не совпавших обнаружений ν , равный массиву индексов обнаружений;
- Далее совершается итерация по возрасту дорожек, начиная с наименьшего и до максимального возраста среди всех дорожек: выбирается подмножество дорожек, которые не были связаны с обнаружением в последних n кадрах;
- Рассчитывается линейное расстояние между треками подмножества и ненайденными обнаружениями из ν ;
- Среди элементов матрицы стоимости C находится минимальный элемент $[c_{ij}]$;
- После этого набор совпадений μ дополняется теми совпадениями, произведение соответствующих значений матрицы ассоциации V и минимального элемента матрицы стоимости C которых больше нуля;
- Из набора не совпавших обнаружений ν убираются элементы, попавшие в набор совпадений μ .

После решения всех теоретических проблемы, пришло время перейти к программной реализации.

3 Экспериментальная часть

В главе представлены основные составляющие трекера, рассмотрены тестовые наборы данных, на которых производилось тестирование трекера, выполнена оценка точности результатов трекинга, представлены примеры работы трекера.

3.1 Программная реализация. Библиотеки

Использование библиотек при разработке системы автоматического трекинга людей позволяет уменьшить объем набираемого кода, позволяет взаимодействовать с операционной системой пользователя, а также использовать векторизованный код, то есть все операции над массивами будут выполняться для каждого значения в этом массиве. Для разрабатываемой системы автоматического трекинга людей по видеоданным были выбраны следующие библиотеки Python:

- `argparse` — библиотека для работы с параметрами запуска;
- `os` — библиотека для взаимодействия программы с инструментами операционной системы, на которой ведется запуск программы;
- `cv2` — библиотека для работы с изображениями;
- `numpy` — библиотека для работы с числовыми массивами данных;
- `colorsys` — библиотека для создания уникальных цветовых оттенков;
- `time` — библиотека для работы с временными данными;
- `sklearn.utils.linear_assignment_` — библиотека для линейного сопоставления векторов признаков;
- `scipy.linalg` — библиотека, представляющая собой модуль линейной алгебры.

Встраивание библиотек в систему выполняется путем написания ключевого слова `import`. Если библиотека имеется в наличии дистрибутива

Python, то больше никаких действий производить не требуется, если нет, то при запуске программы возникает ошибка, в которой будет сказано, что импортируемый модуль не найден. Такую проблему можно решить при помощи менеджера пакетов Python — PyPI (сокращенно pip). Сначала его потребуется скачать с официального сайта [29]. Затем, в консоли разработчика необходимо ввести команду `pip install [имя пакета]`. После установки попробовать снова запустить программу.

После установки и подключения необходимых библиотек, были созданы основные компоненты (классы) трекера: Track, Tracker и Visualizator.

3.1.1 Трек

В программной реализации под треком будем понимать местоположение отслеживаемого объекта на изображении. Каждый трек может иметь одно из трех состояний:

- появившийся,
- подтвержденный,
- подлежащий удалению.

Кроме того, каждый трек характеризуется четырехмерным вектором, состоящим из координат (x, y) центра ограничивающей рамки, ее высоты, а также коэффициентом отношения длины и ширины этой рамки, рис. 13.



Рисунок 13 — На рисунке трек выделен красным цветом

Трек имеет следующие характеристики:

- средний вектор начального распределения,
- ковариационную матрицу распределения,
- уникальный идентификатор,
- количество последовательных обнаружений до подтверждения трека (если трек не подтверждается в течение этого количества, его состояние устанавливается в надлежащее к удалению),
- время жизни (если в течение некоторого количества сопоставлений трек не был сопоставлен с действительным положением наблюдаемого объекта, он будет переведен в состояние надлежащего к удалению),
- вектор признаков обнаружения, при помощи которого осуществляется этот трек (если не является пустым, его данные заносятся в кэш признаков),
- количество обновлений измерения,
- количество кадров с момента появления трека,

- количество фреймов с момента последнего обновления измерения.

Треки имеют широкий инструментарий для работы с ними при помощи алгоритмов обработки. Например, трек может вернуть свой четырехмерный вектор в формате ограничивающей рамки: координаты (x, y) , с учетом начала координат в верхнем левом углу, ширину и высоту рамки. Помимо этого, есть возможность вернуть вектор в формате минимального и максимального значения x и y — крайних углов ограничивающей рамки. На основании информации о треке можно при помощи фильтра Калмана предсказать будущее положение объекта наблюдения, а также обновить предыдущее предположение. Треки сами следят за своим сроком жизни, поэтому, если количество фреймов жизни обнаружения подошло к концу, либо трек не был подтвержден до определенного момента, этот трек изменяет свое состояние на надлежащий к удалению.

3.1.2 Трекер

Трекер является связующим звеном между треками, различными метриками и фильтрами Калмана. На вход он получает метрику ближайшего соседа, максимальное расстояние IoU (Interception over Union), максимальное число пропущенных промахов сопоставления перед удалением трека и количество кадров, на протяжении которых трек будет оставаться в фазе инициализации. Трекер двигает состояние жизни трека вперед, вызывая необходимую функцию каждый раз перед обновлением состояния треков. При обновлении состояния треков, трекер запускает каскад сопоставлений, после чего обновляет каждый трек последовательности. В конце происходит обновление метрики расстояния и обновление треков завершается.

3.1.3 Визуализатор

Данный модуль показывает конечный результат трекинга при помощи библиотеки OpenCV (cv2) [30]. Из полученной в самом начале работы программы информации, визуализатор узнает размеры изображений, которые надлежит отобразить, вычисляет отношение ширины и высоты этих изображений, а также ID первого и последнего изображений. После череды дополнительных манипуляций, визуализатор отображает окно вывода, в котором начинает отображать результат трекинга. Создается впечатление, что идет передача видеопотока, однако на самом деле перед зрителем происходит быстрая смена изображений. Процесс отображения можно прекратить, нажав клавишу Escape, приостановить или возобновить клавишей Space.

После создания основных компонентов, были реализованы методы, описанные в главах 2.3.2.1-2.3.2.4. Последовательность работы созданной системы автоматического трекинга людей описана далее в 3.1.4.

3.1.4 Работа системы трекинга

Работа системы автоматического трекинга людей по последовательности изображений состоит из нескольких этапов.

1. На начальном этапе идет сбор данных о последовательности, на протяжении которой будет осуществляться трекинг людей. В список начальных данных входят:

- полный путь к последовательности,
- название каждого фрагмента последовательности,
- файл с обнаружениями и связанными с ними векторами глубоких признаков в формате MOTChallenge,
- информация о размерах изображений (длина и ширина),
- индекс первого фрейма и индекс последнего фрейма.

2. На последующем этапе идет просчет метрики ближайшего соседа, которой является ближайшее расстояние к любому наблюдаемому объекту. В

качестве метрики была взята метрика косинусного подобия, рассчитываемого по формуле (13):

$$\text{similarity} = \frac{A \times B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}, \quad (13)$$

где A и B — 128-мерные векторы признаков, полученные в результате работы нейросетевой модели, описанной в 2.1.

Ближайшее расстояние задается вручную во время запуска программы, либо используется стандартное значение 0.2.

3. Завершающим этапом является инициализация трекера, результатом работы которого будет являться объединение последовательностей изображений в видеоряд. Именно благодаря трекеру на каждое изображение последовательности будет производиться наложение результатов предсказания места нахождения людей в кадре в виде ограничивающей каждого человека рамки.

После инициализации трекера выполняется запуск отображающего окна, в котором в реальном времени идет трекинг, осуществляемый при помощи методов, рассмотренных в 2.3.2. Весь процесс схематично представлен на рис. 14.

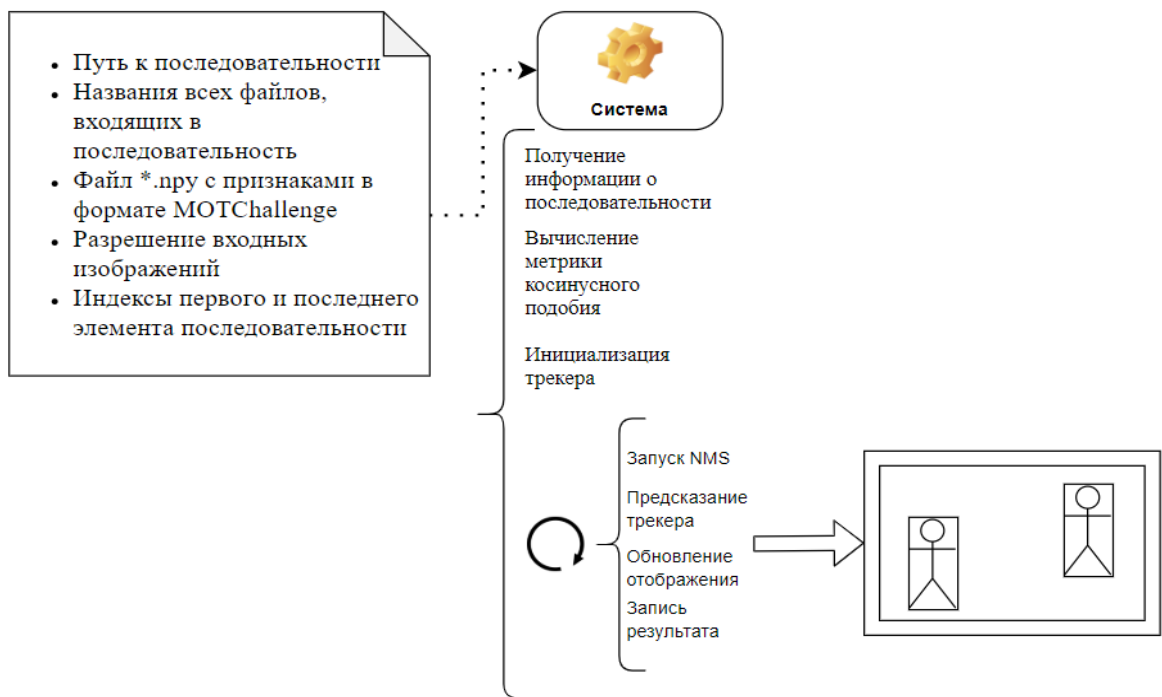


Рисунок 14 — Схема работы алгоритма трекинга

Каждый фрейм обрабатывается следующим образом: из информации, полученной на начальном этапе, создается список обнаружений, после чего идет отсеивание обнаружений, мера достоверности которых выше заданного значения; затем применяется метод подавления немаксимумов, описанный в 2.3.2.1, задача которого заключается в отсеивании накладывающихся друг на друга обнаружений и создание одной наибольшей площади; затем трекер рассчитывает новое положение наблюдаемого объекта, делая предсказание и обновляя положение обнаружений; модуль визуализации отображает рамки обнаружений, а также рамки предсказанного положения каждого отслеживаемого объекта; в конце концов, каждый трек проверяется на достоверность обнаружений и на время спустя первое обновление: если трек подтвержден, а время обновления не превышает 1, трек записывается в финальный список, который затем можно вывести в виде текстового файла.

Для использования системы автоматического трекинга людей по видеоданным необходимо рассмотреть параметры запуска, которые следует указать для корректной работы системы.

3.1.5 Параметры запуска системы трекинга

Запуск программы осуществляется путем задания некоторых опциональных и обязательных параметров. Обязательные параметры — параметры, запуск программы без которых невозможен: `sequence_dir` и `detection_file`. Необязательными (опциональными) параметрами являются: `output_file`, `min_confidence`, `nms_max_overlap`, `max_cosine_distance`, `display`.

Параметры `sequence_dir` и `detection_file` необходимы для указания программе абсолютного пути к папке с последовательностью изображений для обнаружения, а также указания абсолютного пути к файлу обнаружений, полученному при работе нейросетевой модели.

Если запуск программы осуществлялся с параметром `display` равным `True`, пользователь увидит окно программы, в котором будет наглядно видно процесс трекинга, случае `display=False`, трекинг будет проведен в фоновом режиме. В любом случае будет запущен `visualizer`, который начнет обработку кадров.

Если запуск программы осуществлялся с параметром `output_file`, то программа откроет текстовый файл, название которого будет указано в параметрах, и запишет в него результаты трекинга в формате MOTChallenge.

Параметр `min_confidence` ограничивает порог достоверности обнаружения. Если достоверность обнаружения меньше заданного (от 0 до 1), данное обнаружение не будет обработано программой. Если не указать при запуске, будет выставлено стандартное значение — 0.8.

Параметр `nms_max_overlap` выставляет порог для метода подавления немаксимумов. Если не указан, принимает стандартное значение 1.0.

Параметр `max_cosine_distance` является ограничителем метрики косинусного расстояния (при появлении объекта). Если запуск производился без указания значения данного параметра, он автоматически присвоит себе значение 0.2.

3.2 Тестирование системы

Тестирование системы автоматического трекинга людей по видеоданным производилось на наборе данных MOT16, который содержит 14 наборов видеоряда, состоящих из картинок. Набор данных предполагает 7 тренировочных сетов и 7 проверочных. На разработанной системе были протестированы все 14 наборов. MOT16 создан из различного рода видеофрагментов, отснятых статически и динамически. Отслеживание и оценка выполняются в координатах изображения. Все последовательности изображений, аннотированные с высокой точностью, строго следуя протоколу об аннотациях.

В таблице 1 приведено краткое описание каждой последовательности изображений из датасета MOT16.

Таблица 1 — описание последовательностей MOT16

Пример сцены	Название	Разрешение	Количество рамок обнаружений
--------------	----------	------------	------------------------------

Продолжение таблицы 1

	MOT16-01	1920 x 1080	6395
	MOT16-02	1920 x 1080	17833
	MOT16-03	1920 x 1080	104556
	MOT16-04	1920 x 1080	47557
	MOT16-05	640 x 480	6818

Продолжение таблицы 1

	MOT16-06	640 x 480	11538
	MOT16-07	1920 x 1080	16322
	MOT16-08	1920 x 1080	16737
	MOT16-09	1920 x 1080	5257
	MOT16-10	1920 x 1080	12318

Окончание таблицы 1

	MOT16-11	1920 x 1080	9174
	MOT16-12	1920 x 1080	8295
	MOT16-13	1920 x 1080	11450
	MOT16-14	1920 x 1080	18483

3.3 Оценка точности

Для оценки точности методов трекинга использовались следующие метрики:

- MOTA (Multi-object tracking accuracy) — точность трекинга нескольких объектов. Совокупность точности по показателям FP, FN и ID.
- MOTP (Multi-object tracking precision) — точность расположения трекинга нескольких объектов. Мера, отражающая точность расположения ограничивающей рамки объекта трекером по сравнению с его действительным местоположением.
- MT (Mostly tracked) — процент треков, имеющих одну и ту же рамку наблюдения на протяжении хотя бы 80% времени своего существования.
- ML (Mostly lost) — процент ground-truth треков, которые отслеживались до 20% времени своего существования.
- ID (Identity switches) — количество раз, когда уникальный идентификатор трека менялся.
- FM (Fragmentation) — количество раз, когда трек прерывался из-за пропущенного обнаружения.
- FP (False positives) — мера отношения ложных срабатываний к общему числу срабатываний.
- FN (False negatives) — мера отношения количества не осуществленных срабатываний к общему числу срабатываний.

Таблица 2 — сравнение характеристик различных трекеров

Метод	Режим работы	MOT A	MOTP	MT	ML	ID	FM	FP	FN
LMP_p	BATCH	71,0	80,2	46,9%	21,9%	434	587	7880	44564
NOMTws DP16	BATCH	62,2	79,6	32,5%	31,1%	406	642	5119	63352
EAMTT	ONLINE	52,5	78,8	19,0%	34,9%	910	1321	4407	81223
POI	ONLINE	66,1	79,5	34,0%	20,8%	805	3093	5061	55914
SORT	ONLINE	59,8	79,6	25,4%	22,7%	1423	1835	8698	63245

Предложенный	ONLINE	61,4	79,1	32,8%	18,2%	781	2008	12852	56668
--------------	--------	------	------	-------	-------	-----	------	-------	-------

Разработанная система показывает конкурентный результат по показателю MOTP, что означает высокую точность отображения рамки вокруг человека. Из этого следует, что примененные методы решения проблем повторной идентификации, такие как центры рамок обнаружений и фильтры Калмана, а также метод подавления немаксимумов оказали значительное влияние на качество трекинга. Показатель MT в 32.8% указывает на то, что каждый третий человек в кадре отслеживался непрерывно почти все время. Кроме того, по показателю ID, равному 781, видна высокая эффективность фильтра Калмана (у метода SORT этот показатель почти в два раза больше).



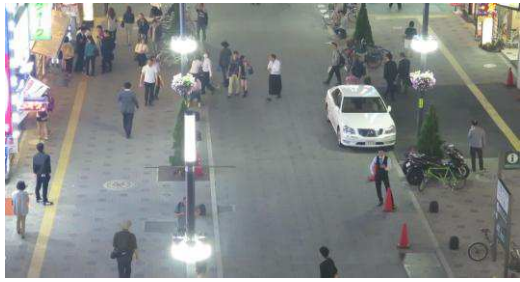
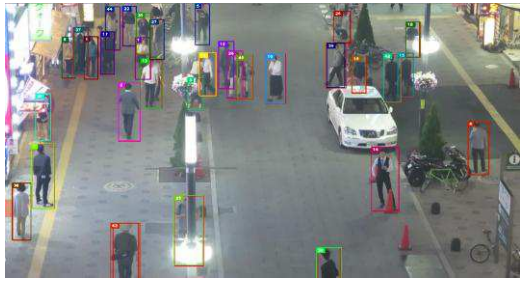



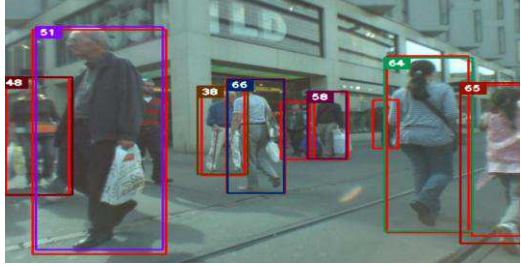
3.4 Примеры работы трекера



Таблица 3 — пример работы системы автоматического трекинга людей по видеоданным на датасете MOT16

Датасет (кадр)	Изначальный кадр	Обработанный системой кадр
MOT16-01 (446)		

<p>MOT16-02 (32)</p>		
--------------------------	---	--

Продолжение таблицы 3

<p>MOT16-03 (128)</p>		
<p>MOT16-04 (94)</p>		
<p>MOT16-05 (325)</p>		
<p>MOT16-06 (128)</p>		

<p>MOT16-07 (305)</p>		
<p>MOT16-08 (118)</p>		

Окончание таблицы 3

<p>MOT16-09 (447)</p>		
<p>MOT16-10 (208)</p>		
<p>MOT16-11 (234)</p>		
<p>MOT16-12 (303)</p>		



На датасете MOT16 трекинг выглядит сносно, однако, чтобы наглядно продемонстрировать проблему использования центроидов рамок наблюдений, система была дополнительно проверена на более позднем датасете MOT20. Дело в том, что наборы данных в MOT16 имели коэффициент плотности от 8,1 до 69,7. Эти показатели соответствуют довольно небольшой плотности людей на сцене. В MOT20 коэффициент плотности находится в пределах от 62,1 до 226,6. Результаты трекинга представлены в таблице 4.

Таблица 4 — пример работы системы автоматического трекинга людей по видеоданным на датасете MOT20 [31]

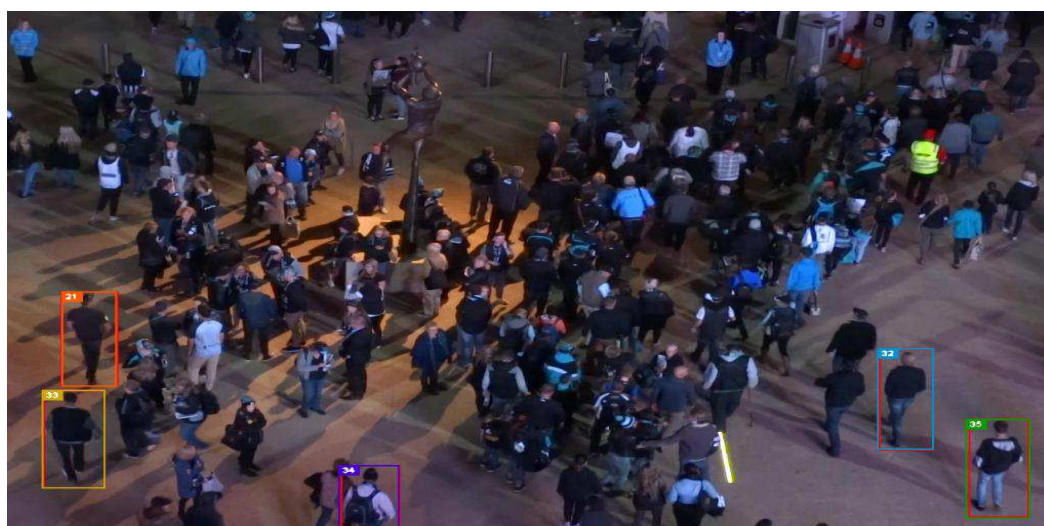
Название датасета	Пример работы системы
-------------------	-----------------------

<p>MOT20-01</p>	
<p>MOT20-02</p>	

Окончание таблицы 4

<p>MOT20-03</p>	
-----------------	--

МОН20-05



Из результатов трекинга видно, что расчет расстояния между центроидами рамок идет очень затруднительно ввиду большой кучности людей. Приходится вычислять расстояния между каждым отдельным человеком и остальными. Ситуацию осложняет то, что сцена является динамической. Таким образом, использование трекера для слишком массовых сцен является затруднительным.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы были изучены основные этапы трекинга, проанализирован трекинг на основе точек артикуляции, разработана система автоматического трекинга людей по видеоданным. В процессе разработки системы были обнаружены и устранены проблемы трекинга. Система была протестирована и сравнена с другими известными методами трекинга.

Результаты экспериментальных исследований могут быть использованы для расширения возможностей трекера, например, возможно измерять аномальное поведение людей, идентифицировать их личность, выявлять различные заболевания.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Multiple People Tracking with Articulation Detection and Stitching Strategy [Электронный ресурс]. Режим доступа: <https://sci-hub.tw/https://www.sciencedirect.com/science/article/abs/pii/S0925231219317400>.
2. A dual CNN–RNN for multiple people tracking [Электронный ресурс]. Режим доступа: <https://sci-hub.tw/https://www.sciencedirect.com/science/article/abs/pii/S0925231219311270>.
3. People tracking in an environment with multiple depth cameras: A skeleton-based pairwise trajectory matching scheme [Электронный ресурс]. Режим доступа: <https://sci-hub.tw/https://www.sciencedirect.com/science/article/pii/S1047-32031500228X>.
4. An automatic data mining method to detect abnormal human behaviour using physical activity measurements [Электронный ресурс]. Режим доступа: <https://sci-hub.tw/https://www.sciencedirect.com/science/article/abs/pii/S1574119214001606>.
5. MOT Challenge - MOT16 [Электронный ресурс]. Режим доступа: <https://motchallenge.net/data/MOT16/>.
6. Fusion of Head and Full-Body Detectors for Multi-Object Tracking [Электронный ресурс]. Режим доступа: http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w29/Henschel_Fusion_of_Head_CVPR_2018_paper.pdf.
7. A Detection-Based Multiple Object Tracking Method [Электронный ресурс]. Режим доступа: <http://www.iitg.ac.in/amitsethi/publications/04.04TrackingICIP.pdf>.
8. Background modeling and foreground extraction in video data using spatio-temporal region persistence features [Электронный ресурс]. Режим доступа: <https://sci-hub.tw/https://www.sciencedirect.com/science/article/pii/S0045790619300084>.

9. Global Data Association for Multi-Object Tracking Using Network Flows [Электронный ресурс]. Режим доступа: http://vision.cse.psu.edu/courses/Tracking/vlpr12/lzhang_cvpr08global.pdf.
10. The KITTI Vision Benchmark Suite [Электронный ресурс]. Режим доступа: <http://www.cvlibs.net/datasets/kitti/>.
11. CDT: Cooperative Detection and Tracking for Tracing Multiple Objects in Video Sequences [Электронный ресурс]. Режим доступа: https://sci-hub.tw/https://doi.org/10.1007/978-3-319-46466-4_51.
12. Fast R-CNN [Электронный ресурс]. Режим доступа: <https://arxiv.org/pdf/1504.08083.pdf>.
13. Subgraph Decomposition for Multi-Target Tracking [Электронный ресурс]. Режим доступа: https://www.is.mpg.de/uploads_file/attachment/attachment-355/tangCVPR15.pdf.
14. GMMCP Tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking [Электронный ресурс]. Режим доступа: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Dehghan_GMMCP_Tracker_Globally_2015_CVPR_paper.pdf.
15. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs [Электронный ресурс]. Режим доступа: https://cs.stanford.edu/~amirz/index_files/GMCP-Tracker_ECCV12.pdf.
16. Multi-person Tracking by Multicut and Deep Matching [Электронный ресурс]. Режим доступа: https://www.is.mpg.de/uploads_file/attachment/attachment-353/siyu_eccvw.pdf.
17. Multiple People Tracking by Lifted Multicut and Person Re-identification [Электронный ресурс]. Режим доступа: http://openaccess.thecvf.com/content_cvpr_2017/papers/Tang_Multiple_People_Tracking_CVPR_2017_paper.pdf.
18. Multi-Object Tracking with Quadruplet Convolutional Neural Networks [Электронный ресурс]. Режим доступа: http://openaccess.thecvf.com/content_cvpr

r_2017/papers/Son_Multi-Object_Tracking_With_CVPR_2017_paper.pdf.

19. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples [Электронный ресурс]. Режим доступа: <https://arxiv.org/pdf/1604.02426.pdf>.

20. Very deep convolutional networks for large-scale image recognition [Электронный ресурс]. Режим доступа: <https://arxiv.org/pdf/1409.1556.pdf>.

21. Person Re-Identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function [Электронный ресурс]. Режим доступа: https://www.cv-foundation.org/openaccess/content_cvpr_2016/app/S06-37.pdf.

22. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor [Электронный ресурс]. Режим доступа: http://www.nec-labs.com/uploads/images/Department-Images/MediaAnalytics/papers/iccv15_onlinetracking.pdf.

23. ArtTrack: Articulated Multi-person Tracking in the Wild [Электронный ресурс]. Режим доступа: <https://arxiv.org/pdf/1612.01465.pdf>.

24. PoseTrack: Joint Multi-Person Pose Estimation and Tracking [Электронный ресурс]. Режим доступа: <https://arxiv.org/pdf/1611.07727.pdf>.

25. MOT16: A Benchmark for Multi-Object Tracking [Электронный ресурс]. Режим доступа: <https://arxiv.org/pdf/1603.00831.pdf>.

26. MARS: A Video Benchmark for Large-Scale Person Re-Identification [Электронный ресурс]. Режим доступа: https://www.researchgate.net/publication/308277502_MARS_A_Video_Benchmark_for_Large-Scale_Person_Re-Identification.

27. Scalable Person Re-identification: A Benchmark [Электронный ресурс]. Режим доступа: <https://sci-hub.tw/https://ieeexplore.ieee.org/document/7410490>.

28. Learning Discriminative Appearance-Based Models Using Partial Least Squares [Электронный ресурс]. Режим доступа: <http://users.umiacs.umd.edu/~lsd/papers/paperSibgrapi09.pdf>.

29. The Python Package Index (PyPI) is a repository of software for the Python programming language [Электронный ресурс]. Режим доступа: <https://pypi.org/>.
30. OpenCV [Электронный ресурс]. Режим доступа: <https://opencv.org/>.
31. MOT Challenge - MOT20 [Электронный ресурс]. Режим доступа: <https://motchallenge.net/data/MOT20/>.
32. Non-Maximum Suppression for Object Detection in Python [Электронный ресурс]. Режим доступа: <https://www.pyimagesearch.com/2014/11/17/non-maximum-suppression-object-detection-python/>.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
Кафедра информационных систем

УТВЕРЖДАЮ

Заведующий кафедрой

_____ П.П. Дьячук


подпись

« _____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 — «Информационные системы и технологии»

Применение технологий машинного обучения для трекинга людей по
видеоданным

Руководитель  26.06.2020
подпись, дата

доцент, к.т.н.

А.В. Пятаева

Выпускник  26.06.2020
подпись, дата

К.А. Калинин

Красноярск 2020