ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2020, № 4, с. 110–124

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

УДК 004.932.2

ОПТИМИЗАЦИЯ РАБОТЫ СИСТЕМЫ СЛЕЖЕНИЯ, ОСНОВАННОЙ НА СЕТИ КАМЕР[1]

© 2020 г. И. А. Матвеев[a,*], В. В. Чигринский[b,]**

[a] *ФИЦ ИУ РАН, Москва, Россия*

[b] *МФТИ, Долгопрудный, МО, Россия*

**e-mail: matveev@ccas.ru*

***e-mail: chigrinskiy.viktor@phystech.edu*

Поступила в редакцию 10.02.2020 г.
После доработки 29.02.2020 г.
Принята к публикации 30.03.2020 г.

Отслеживание перемещения объектов на видеопоследовательностях – важная задача компьютерного зрения, имеющая широкое практическое применение. Ключевыми моментами работы систем слежения являются обнаружение объекта и, в случае если он был обнаружен не в первый раз, его повторная идентификация. Представлена быстро и корректно работающая система слежения, использующая несколько камер и включающая в себя: детектирование и сегментацию объектов на изображении, получение дескрипторов их внешнего вида, сравнение каждого нового объекта с уже накопленными, принятие решения о повторной идентификации. Реализована базовая конфигурация системы, в которой в качестве составляющих используются лучшие на текущий момент алгоритмы детектирования и модели получения дескрипторов внешнего вида. На этой основе произведены модификации как отдельных модулей, так и всей системы в целом. Выполнен вычислительный эксперимент, количественно подтверждающий преимущество доработанной системы относительно базовой.

DOI: 10.31857/S0002338820040125

Введение. Для начала введем несколько определений попутно с кратким описанием принципа работы системы. *Мультикамерным трекингом* будем называть слежение за объектами по данным с нескольких камер видеонаблюдения. Видеопотоки с камер кадр за кадром обрабатываются *детектором* – подсистемой, которая определяет наличие и местоположение интересующих объектов на полном кадре. Результат обнаружения объекта на кадре (совокупность его характеристик) называется *детекцией*. При появлении новой детекции в работу вступает *трекер* – система, отслеживающая определенный объект до тех пор, пока он не пропадет из вида. Упорядоченный по времени набор детекций объекта называется его *треком*. После формирования трека последовательность детекций с помощью какой-либо модели (будь то нейронная сеть или классический метод) преобразуется в векторное описание объекта, которое называется его *дескриптором*. Дескрипторы обработанных объектов сохраняются в специальную базу данных, которую будем называть *галереей*, текущий обрабатываемый объект после построения дескриптора – *запросом*. При появлении нового запроса его дескриптор сравнивнивается со всеми дескрипторами из галереи, и на основе этого сравнения ему присваивается уникальное число, называемое *идентификатором* или *ID*, определяющее его принадлежность определенному классу. Например, в случае слежения за людьми идентификатор соответствует личности человека, поэтому уникальный объект будем называть *персоной*. Сравнение запроса с галереей и присвоение ID объекту запроса называется *повторной идентификацией* или, более кратко, *реидентификацией*. Расширенным вариантом реидентификации является ранжирование галереи по сходству с запросом. Таким образом, цель всей системы состоит в присвоении одинаковых ID одинаковым персонам и разных – разным в поступающих видеопотоках.

Задача мультикамерного трекинга была впервые формально рассмотрена в 1997 г. в [1]; был предложен байесовский подход, использующий информацию о внешнем виде объектов, времени съемки и положении в пространстве. Более подробно история развития этого направления на

[1]Работа выполнена при частичной финансовой поддержке РФФИ (грант № 19-07-01231).

ранних стадиях описана в [2]. В 2005 г. в [3] впервые был введен термин “реидентификация”. Годом позже реидентификация была сформулирована и рассмотрена как самостоятельная задача компьютерного зрения [4]. Со временем задача реидентификации людей стала активно набирать популярность. В 2014 г., после того как глубокие сверточные нейронные сети хорошо зарекомендовали себя в задачах классификации изображений, было предложено [5, 6] использовать их и в задаче реидентификации, что привело к заметному улучшению качества. В этом же году задача реидентификации была рассмотрена совместно с задачей детектирования людей на видео [7]. Это мотивировалось тем, что качество распознавания людей зависит не только от алгоритма построения дескрипторов, но также и от способа их детектирования. Оказывается, что качество детектора сильно влияет на качество распознавания, что подчеркнуто в [7]. В подробностях о задаче реидентификации людей, о ее развитии с 1997 до 2016 г. и о наиболее известных методах можно прочитать в обзоре [8].

В данной работе развивается и обобщается идея, предложенная в [7]. Рассмотрена не только комбинация детектирования и распознавания, но вся система в целом, начиная с покадровой обработки видеопотоков и заканчивая присвоением уникальных ID уникальным объектам. Проведено полное исследование по каждой составляющей задачи мультикамерного трекинга на примере людей и построена полная, быстро работающая и устойчивая система, решающая эту задачу. Основной целью является широкая применимость.

1. Детекторы, дескрипторы и базы данных. В этом разделе описаны известные детекторы, использованные в работе, дескрипторы признаков, применяемых для идентификации, а также базы данных, находящиеся в открытом доступе, на которых были проведены вычислительные эксперименты.

1.1. Детекторы. Известно большое количество детекторов признаков, основанных на различных эвристических подходах [9, 10]. В последние годы все большее значение в первичной обработке сигналов, в том числе в решении задач детектирования, приобретают нейронные сети. Эпоха детекторов, построенных на сверточных нейронных сетях, берет свое начало в 2012 г. с появлением *Region-based Convolutional Neural Network* (RCNN) [11]. Этот детектор быстро развивался, превращаясь последовательно в *Fast RCNN* [12], *Faster RCNN* [13], *Mask RCNN* [14] и *Keypoint RCNN* [14]. Последние два помимо того, что показывают лучший результат детектирования объектов на базе MS-COCO-2016 [15], предоставляют также информацию о сегментации и ключевых точках объекта соответственно. Несмотря на то, что скорость работы Faster RCNN, Mask RCNN и Keypoint RCNN существенно возросла в сравнении с оригинальным методом, эти детекторы не позволяют обрабатывать видео формата Full-HD в режиме реального времени. В конце 2015 г. появляется *Single Shot multibox Detector* (SSD) [16], в котором основная идея YOLO [17] – другого детектора, предложенного ранее, переносится на Faster RCNN. Это дает результаты, сравнимые с лучшими по качеству детекторами Mask RCNN и Keypoint RCNN при скорости обработки видео в 60 кадров в секунду. В табл. 1 приведены количественные сравнительные характеристики всех основных детекторов.

1.2. Дескрипторы. Основными компонентами описания объекта в задаче реидентификации являются время, местоположение и внешний вид объекта. Информация о времени и месте присутствует не во всех случаях, а наличие изображений предполагается всегда, поэтому основой описания объекта является его внешний вид, а пространственная и временная информация – вспомогательная. Для того чтобы формировать описание объекта, разрабатываются различные модели, переводящие изображение в n-мерный вектор признаков – дескриптор объекта. Расстояние между такими векторами в n-мерном евклидовом пространстве есть численная мера различия объектов. До появления глубокого обучения в задаче реидентификации в 2014 г. [5, 6] в качестве дескриптора объектов использовались гистограммы изображения в различных цветовых пространствах: RGB, HSV, YUV и др., а также такие классические дескрипторы, как HOG, LBP, SIFT, SURF и др.

Большой скачок качества случился после появления глубокого обучения в задаче реидентификации. Впервые формирование дескриптора не на основе эвристик, а с помощью нейронной сети предложили в работе [6]. В описанном методе изображения подвергались сверткам и MaxPooling слоям, делились на части вдоль вертикальной оси, снова подвергались сверткам и MaxPooling слоям и превращались в векторное описание объекта. Сеть обучалась давать для каждой пары изображений ответ: один и тот же объект на них или нет.

В 2016 г. была реализована идея обучить нейронную сеть классифицировать людей на изображениях [24]. В качестве классов были выбраны ID персон обучающей выборки. Во время тестирования предлагается отбросить последний слой сети, осуществляющий классификацию, и

Таблица 1. Детекторы

Название	Средняя точность на PASCAL VOC 2007, %	Средняя точность на MS COCO 2016, %	Скорость, кадров в секунду	Год
Выборочный поиск [18]	35.1	–	–	2012
RCNN [11] (AlexNet [19])	53.7	–	–	2012
RCNN (ZFNet [20])	59.2	–	–	2013
RCNN (VGGNet [21])	62.0	–	–	2014
Fast RCNN [12] (VGGNet)	70.0	–	0.5	2014
YOLO [17] (VGGNet)	66.4	–	21	2014
Faster RCNN [13] (VGGNet)	73.2	–	14	2015
SSD [16] (VGGNet)	74.3	–	59	2016
Faster RCNN (ResNet-101 [22])	–	34.9	5	2016
Mask RCNN [14] (ResNet-101)	–	38.2	5	2017
Mask RCNN (ResNeXt-101 [23])	–	39.8	2.5	2017

Таблица 2. Модели получения дескриптора

Название	Особенность	Точность на Market1501, %	Год
HGD [28]	Без учителя	32.04	2015
BoW+KISSME [29]	Метрическое обучение	44.40	2015
IDE [24] (AlexNet [19])	Обучение классификации	56.03	2016
IDE (VGGNet [21])	–	64.34	2016
IDE (ResNet [22])	–	72.54	2016
TriNet [30]	Triplet Loss [27]	84.92	2017
PCB [25]	Деление вдоль вертикали	92.40	2017
TriNet2 [31]	Triplet Loss	92.20	2018
MGN [26]	Идеи PCB и TriNet	95.70	2018

брать вектор с предпоследнего слоя в качестве дескриптора объекта. Такой подход назвали *Identity Discriminative Embedding* (IDE). Эта простая идея лежит в основе большинства современных систем.

В 2017 г. появилась статья [25], авторы который взяли за основу IDE и предложили делить признаковую карту, получающуюся на выходе ResNet [22] на N частей вдоль вертикали и на каждой из этих частей обучать свой независимый классификатор. Модель назвали *Part-based Convolutional Baseline* (PCB). Годом позже появилась *Multiple Granularity Network* (MGN) [26], в которой идея [25] была модифицирована. В этом варианте после ResNet без последнего блока сеть делится на три независимые части. Первая часть генерирует глобальные признаки объекта, вторая и третья части делят карту признаков вдоль вертикали на две и три части соответственно, формируя локальные признаки. Обучение осуществляется на каждом признаковом векторе независимо, функция потерь представляет собой сумму кроссэнтропии и Triplet Loss [27].

Сравнение всех основных моделей дано в табл. 2.

1.3. Базы данных. В табл. 3 приведена сводка по основным базам данных для задачи реидентификации. База может состоять либо из отдельных изображений (тип “И”), либо из треков, (тип “В”). Объектом в случае базы типа “И” выступает одно изображение, а в случае базы типа “В” – один трек.

В большинстве работ по реидентификации и мультикамерному трекингу последних лет для обучения и тестирования моделей чаще других используются базы данных Market1501 [29], DukeMTMC [35] и CUHK03 [6].

Таблица 3. Базы данных для задачи реидентификации

Название	Тип	Количество		Число камер	Год
		ID	объектов		
RPI field [32]	В	112	1585	12	2018
MSMT17 [33]	И	4101	126443	15	2018
AirportALERT [34]	И	9651	30243	6	2017
DukeMTMC-reID [35]	И	1812	36411	8	2017
MARS [24]	В	1259	16360	6	2016
LSPS [36]	И	11934	34574	–	2016
PRW [37]	И	933	33856	6	2016
PKU [38]	И	114	1824	2	2016
Market1501 [29]	И	1501	29419	6	2015
CASIA [39]	В	124	8181	11	2015
Shinpuhkan2014 [40]	В	24	2064	16	2014
HDA+ [41]	В	69	493	13	2014
MPR Drone [42]	И	69	2276	–	2014
iLIDS VID [43]	В	300	600	2	2014
RAiD [44]	В	43	172	4	2014
CUHK03 [6]	И	1467	14096	10	2014
CUHK02 [45]	И	1816	7264	10	2013
CUHK01 [46]	И	971	3884	2	2012
SAIVT SoftBio [47]	В	152	788	8	2012
WARD [48]	В	70	210	3	2012
V47 [49]	И	47	752	4	2011
PRID 2011 [50]	В	934	1134	2	2011
3DPeS [51]	И	193	1012	–	2011
CAVIAR4ReID [52]	И	72	1220	–	2011
GRID [53]	И	250	500	–	2009
QMUL iLIDS [54]	И	119	476	–	2009
ETHZ [55]	В	146	146	–	2007
VIPeR [56]	И	632	1264	2	2007

В 2014 г. появилась CUHK03 [6] – первая база данных для задачи реидентификации, которая является достаточно большой, чтобы обучать и тестировать на ней модели глубокого обучения. В [6] авторы наряду с публикацией базы данных предложили извлекать дескриптор человека не с помощью классических методов, а с использованием глубоких сверточных нейронных сетей, что дало существенный прирост к качеству и открыло новое направление в этой области. База состоит из изображений 1467 персон, полученных с пяти пар камер, на каждую персону приходится до пяти изображений с обеих камер пары. Помимо размеченных вручную изображений людей база также содержит изображения тех же людей, полученные с помощью DPM-детектора [57].

Спустя год, в 2015 г., появилась Market1501 [29] – самая популярная на сегодняшний день база данных для обучения и тестирования алгоритмов реидентификации. Состоит из 29419 изображений 1501 персоны, которые получены с шести непересекающихся камер, расположенных на улице. Подавляющее большинство современных методов реидентификации сравниваются именно на этой базе. Помимо изображений людей также имеются ложные срабатывания детектора, которые при тестировании могут быть специально добавлены в галерею для усложнения задачи и приближения ее к реальной ситуации. Примеры данных приведены на рис. 1.

Рис. 1. Примеры изображений базы Market1501

DukeMTMC [35] – крупнейшая по количеству треков база для мультикамерного трекинга, состоящая из видеозаписей с восьми камер, расположенных на территории кампуса, длительностью 1 ч 25 мин каждая и содержащая 6791 трек 2834 персон. Данные сопровождаются полной разметкой: детекциями всех людей на каждом кадре, принадлежностью каждой детекции определенной персоне, пространственной и временной калибровки камер между собой и с реальным миром. DukeMTMC-reID [35] – подмножество DukeMTMC, состоящее из вырезанных изображений некоторых людей, предназначенное для тестирования алгоритмов реидентификации.

2. Описание системы. Последовательность обработки данных в системе изображена на блок-схеме на рис. 2. С камер на вход поступает N видеопотоков, которые кадр за кадром обрабатываются *детектором* $\mathcal{D}$. При появлении объекта на каком-либо видео детектор дает сигнал *трекеру* $\mathcal{T}$, который с этого момента начинает отслеживать перемещение этого объекта до тех пор, пока тот не пропадет из виду (т.е. до тех пор, пока детектор выдает новые кадры). После этого формируется трек объекта и с помощью *экстрактора* $\mathcal{X}$ извлекается его дескриптор, который затем *матчером* $\mathcal{M}$ сравнивается с *галереей* $\mathbf{G}$, находящейся в специальной базе данных. На основе сравнения принимается решение: отнести объект к ложным срабатываниям детектора и проигнорировать, присвоить ID, уже имеющийся в галерее, либо присвоить новый ID. После окончания всех видеопотоков в галерее будут находиться все обнаруженные треки с метками ID.

Весь процесс реализован многопоточно, схематично это можно увидеть на рис. 3: N основных потоков – это детекторы, по одному на каждый видеопоток, они работают постоянно, обрабатывая кадр за кадром, вне зависимости от того, что происходит на видео. При детектировании объекта на каком-либо видео инициируется процесс трекера. Этот процесс, используя в дальнейшем данные детектора, ведет определенный объект до тех пор, пока детектор не перестанет его обнаруживать. В конечном счете формируется трек объекта. Для каждого нового объекта инициируется свой процесс трекера. В силу того, что используется процесс трекинга, требующий на несколько порядков меньше ресурсов по сравнению с детектором и экстрактором, количество одновременно работающих трекеров может покрывать любое физически допустимое количество одновременно присутствующих людей на видео. После формирования трека запускается процесс экстракции, извлекающий дескриптор для этого трека, а сразу после его завершения – процесс сравнения дескриптора с галереей и принятие решения насчет ID объекта. Обработка, происходящая после завершения детекции, не требует жесткой временной синхронизации с видеопотоком.

2.1. Д е т е к т о р. Опишем теперь всю систему формально по частям. Пусть $I_t^{(i)}$ – кадр под номером t i-го видеопотока, являющийся матрицей размера $W \times H \times 3$. Детектор – функция, переводящая изображение в множество детекций интересующих объектов на ней (которое может быть и пустым):

$$\mathcal{D} : I \to \{d_1, \ldots, d_k\}. \tag{2.1}$$

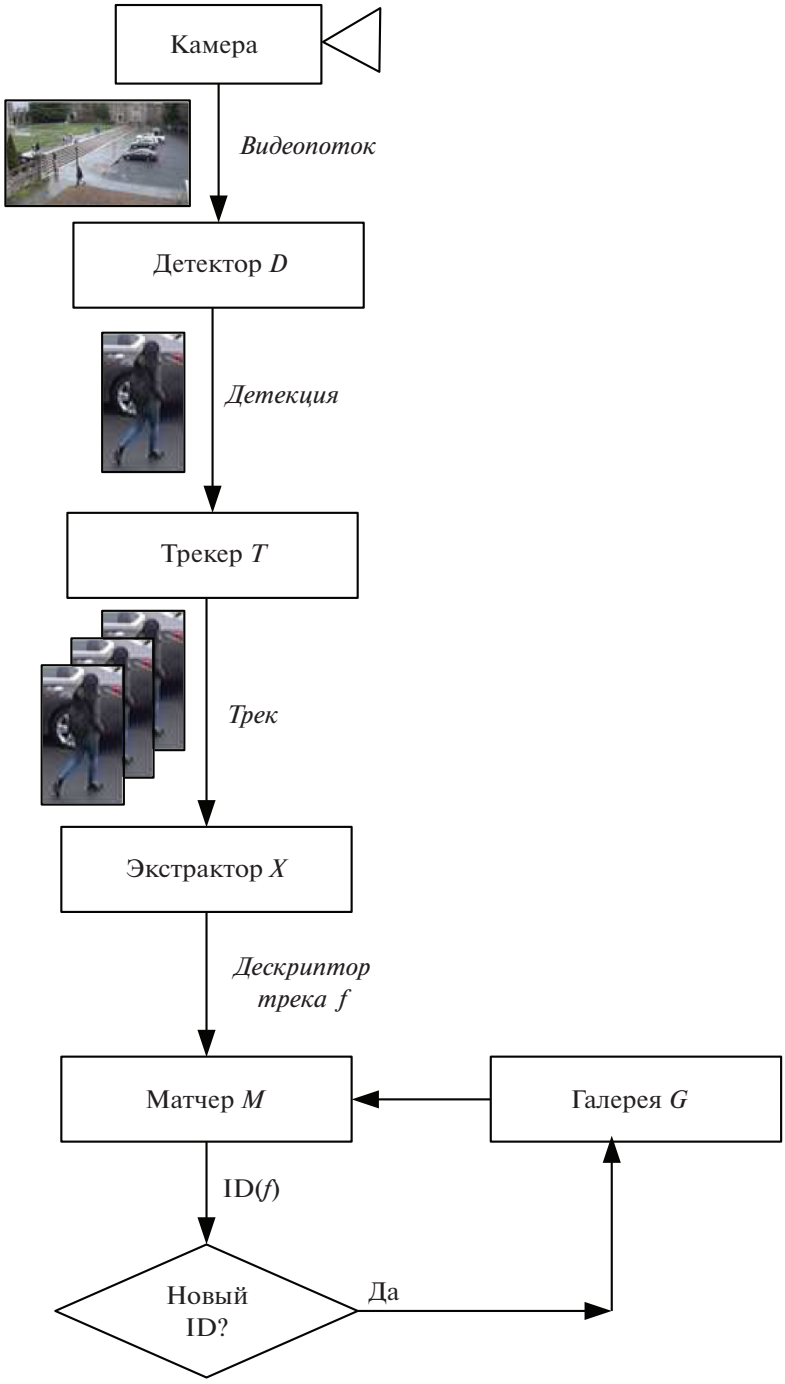

Рис. 2. Последовательность обработки данных

Каждая детекция представляет собой пару

$$d = (\beta, \beta'), \tag{2.2}$$

где $\beta = (l, t, r, b)$ – границы положения объекта на изображении (левая, верхняя, правая и нижняя соответственно), а β' – дополнительная информация, выдаваемая детектором.

В работе рассмотрены и реализованы четыре вида детектора: $\mathfrak{D}_{\text{SSD}}$ (SSD), $\mathfrak{D}_{\text{Faster}}$ (Faster-RCNN), $\mathfrak{D}_{\text{Mask}}$ (Mask-RCNN) и $\mathfrak{D}_{\text{KP}}$ (Keypoint-RCNN). Каждый из них описан в разд. 1.1.

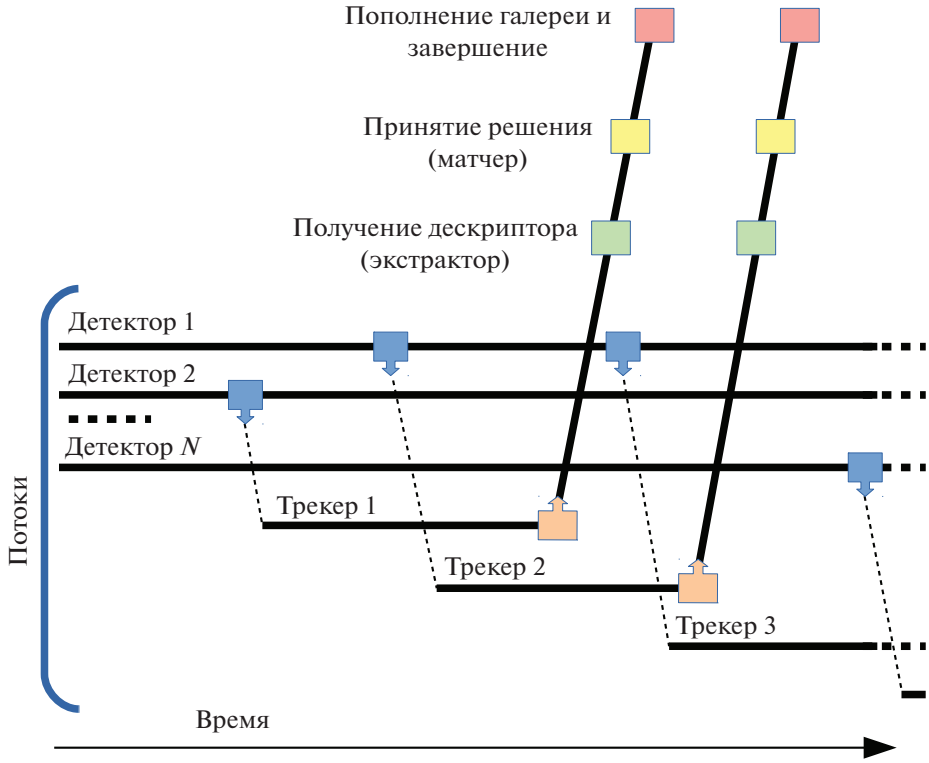

Рис. 3. Процесс работы системы

Для $\mathcal{D}_{\text{SSD}}$ и $\mathcal{D}_{\text{Faster}}$ $\beta' = \beta$, поскольку кроме координат ограничивающего объект прямоугольника эти детекторы ничего не выдают. Для $\mathcal{D}_{Mask}$ $\beta' = \mathbf{M}$ – бинарная маска сегментации объекта в этой области, имеющая размер $(r - l) \times (b - t)$. Для $\mathcal{D}_{\text{KP}}$ $\beta' = \{(\xi^{(1)}, \eta^{(1)}, \chi^{(1)}), \ldots, (\xi^{(17)}, \eta^{(17)}, \chi^{(17)})\}$ – упорядоченный набор 17 ключевых точек тела человека, представленных в виде x-координаты ξ_i, y-координаты η_i и величиной χ_i, которая равна 1, если i-я ключевая точка видна на изображении, и 0 – иначе.

2.2. Т р е к е р. Трекер представляет собой функцию, которая в качестве аргумента принимает пару детекций с последовательных кадров d_t, d_{t+1}, вычисляет для них специальную величину $\mathcal{T}'(\beta'_t, \beta'_{t+1})$ и, если она больше порога $\mathcal{T}_{\text{IoU}} > 0$, дает сигнал о продолжении трека новой детекцией, иначе – нет:

$$\mathcal{T}(d_t, d_{t+1}) = [\mathcal{T}'(\beta'_t, \beta'_{t+1}) > \mathcal{T}_{\text{IoU}}], \tag{2.3}$$

где $[\cdot]$ – функция, принимающая значение 1, если условие в скобках верно, 0 – иначе. Трек завершается при непоявлении новых детекций на протяжении какого-то количества кадров Δ.

Базовый трекер $\mathcal{T}_{\text{Box}}$ продолжает трек новой детекцией, если величина отношения площадей пересечения и объединения ограничивающих объекты прямоугольников новой и старой детекций (назовем ее *Intersection over Union*, IoU) превышает порог $\mathcal{T}_{\text{IoU}}$:

$$\mathcal{T}'_{\text{Box}}(\beta'_t, \beta'_{t+1}) = \text{IoU}_{\text{Box}}(\beta'_t, \beta'_{t+1}) = \frac{S_i^{(\text{Box})}}{S_u^{(\text{Box})}}, \tag{2.4}$$

где $S_i^{(\text{Box})}$ – площадь пересечения ограничивающих прямоугольников:

$$S_i^{(\text{Box})} = \max(0, \min(r_t, r_{t+1}) - \max(l_t, l_{t+1})) \times \tag{2.5}$$

$$\times \max(0, \min(b_t, b_{t+1}) - \max(t_t, t_{t+1})), \tag{2.6}$$

а $S_u^{(\mathrm{Box})}$ – площадь их объединения:

$$S_u^{(\mathrm{Box})} = S_t^{(\mathrm{Box})} + S_{t+1}^{(\mathrm{Box})} - S_i^{(\mathrm{Box})}, \tag{2.7}$$

$S_t^{(\mathrm{Box})}, S_{t+1}^{(\mathrm{Box})}$ – площади самих прямоугольников:

$$S_t^{(\mathrm{Box})} = (r_t - l_t)(b_t - t_t), \tag{2.8}$$

$$S_{t+1}^{(\mathrm{Box})} = (r_{t+1} - l_{t+1})(b_{t+1} - t_{t+1}). \tag{2.9}$$

Такой трекер может использоваться в комбинации с любым детектором.

2.3. Дескриптор. После завершения работы трекера трек объекта представляет собой упорядоченный набор исходных изображений $I_t, \ldots, I_{t+\delta-1}$ и соответствующих им детекций $d_t, \ldots, d_{t+\delta-1}$. На каждом исходном изображении по координатам β детекции вырезается область интереса и масштабируется к определенному размеру $w \times h$, который требуется экстрактору. Экстрактор вычисляет набор признаков – дескриптор, что можно записать как

$$\mathbf{F}(t) = \mathscr{X}(I_t, d_t). \tag{2.10}$$

В качестве общего дескриптора объекта используется усредненный по всем изображениям трека:

$$\mathbf{f} = \frac{1}{\delta} \sum_{\tau=t}^{t+\delta-1} \mathbf{F}(\tau). \tag{2.11}$$

В работе рассмотрены и реализованы две модели извлечения дескриптора: $\mathscr{X}_{\mathrm{PCB}}$ (**PCB**) и $\mathscr{X}_{\mathrm{MGN}}$ (**MGN**).

2.4. Принятие решения. В ходе экспериментов было обнаружено, что дескрипторы с низкой нормой чаще других соответствуют ложным срабатываниям детектора. Поэтому, если значение нормы $\|\mathbf{f}\|$ ниже порога T_{norm}, принимается решение отнести трек к ложным срабатываниям детектора. Иначе дескриптор делится на эту норму и сравнивается с другими, уже сохраненными в галерею **G** нормализованными дескрипторами обработанных ранее объектов. Сравнение осуществляется вычислением евклидова расстояния дескриптора запроса и каждого из дескрипторов галереи. Если расстояние до какого-либо объекта галереи меньше порогового значения T_{dist}, объекту присваивается ID самого близкого сохраненного в галерее объекта. В противном случае считается, что объект встречен впервые и он добавляется в галерею с новым уникальным ID. Весь процесс принятия решения формально выглядит следующим образом:

$$\mathrm{ID}(\mathbf{f}) = \begin{cases} -1, & \text{если } \|\mathbf{f}\| < T_{\mathrm{norm}}; \\ \mathrm{ID}\left(\arg\min\limits_{\mathbf{g}\in\mathbf{G}} \|\mathbf{f} - \mathbf{g}\|\right) & \text{иначе, если } \min\limits_{\mathbf{g}\in\mathbf{G}} \|\mathbf{f} - \mathbf{g}\| < T_{\mathrm{dist}}; \\ |\mathbf{G}| + 1 & \text{иначе;} \end{cases} \tag{2.12}$$

где $|G|$ – число объектов в галерее.

3. Вычислительные эксперименты. В этом разделе описаны вычислительные эксперименты, проведенные при разработке, обучении и моделировании функционирования системы. Также приведены модификации системы: метрики качества, трекера и матчера.

3.1. Тестовые данные. В качестве тестовых данных были выбраны видео базы DukeMTMC. У каждой из восьми камер есть собственная информация о времени записи, которое измеряется в кадрах. Начало записи каждого видео соответствует первому кадру соответствующей камеры. Длительности записи (в кадрах) для видео с каждой из восьми камер:

$$T_1 = 359580, \quad T_2 = 360720, \quad T_3 = 355380, \quad T_4 = 374850,$$
$$T_5 = 366390, \quad T_6 = 344400, \quad T_7 = 337680, \quad T_8 = 353220.$$

Частота записи для каждой из камер постоянна и равна 60 кадрам в секунду. Тестовые данные представляют собой восемь видео, начинающиеся с кадра номер 47720 и заканчивающиеся кадром номер 356648. Длительность каждого видео составляет 1 ч 15 мин.

В [35] предлагается следующее разбиение:

Trainval (обучение и валидация): 0–50 мин (кадры с 47720 по 227540);

Test-hard (сложный тест): 50–60 мин (кадры с 227541 по 263503);

Test-easy (простой тест): 60–85 мин (кадры с 263504 по 356648).

Оно используется и здесь. Объекты из Trainval участвовали в обучении модели дескриптора, объекты из Test-easy и Test-hard – в оценке качества работы алгоритма на соответствующем промежутке времени в простом и сложном сценариях. Простой и сложный сценарии отличаются плотностью объектов на видео, в сложном сценарии она больше.

3.2. Разметка. Разметка представляет собой информацию о всех детекциях и принадлежности объектов на этих детекциях определенному ID. Разметка по каждому объекту x – это множество:

$$\mathrm{ann}(x) = (c, y, \beta), \tag{3.1}$$

где c – идентификатор камеры (число от одного до восьми в данном случае), y – ID объекта, $\beta = (l, t, r, b)$ – границы положения объекта на изображении (левая, верхняя, правая и нижняя соответственно)

3.3. Метрика качества. Пусть $X = \{x_1, \ldots, x_N\}$ – множество всех объектов тестовой выборки, $Y = \{y_1, \ldots, y_N\}$ – множество соответствующих ID из $\mathrm{ann}(\cdot)$ для каждого объекта, а $B = \{\beta_1, \ldots, \beta_N\}$ – локализации объектов на соответствующих кадрах. Пусть также $\Xi(\Theta) = \{\xi_1, \ldots, \xi_M\}$ – выход системы, множество детекций с ID, присвоенными им системой с параметрами Θ, при этом Ξ не содержит те детекции, которые на этапе принятия решения были отнесены к ложным срабатываниям детектора. Информация о принадлежности каждого объекта определенному треку, а также информация о сегментации объекта на изображении используются только во время обработки данных, но не при оценке качества. Множество $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_M\}$ – ID, предсказанные системой; $\hat{B} = \{\hat{\beta}_1, \ldots, \hat{\beta}_M\}$ – локализации объектов, полученные на выходе детектора.

Сперва каждому объекту $\xi_i \in \Xi$ на основе Y, B и $\hat{\beta}_i$ присваивается истинное значение ID $\overline{y}_i$ по следующему правилу:

$$\overline{y}_i = \begin{cases} -1, & \text{если } \max\limits_{j=1,N} \mathcal{T}'(\hat{\beta}_i, \beta_j) < T_{\mathrm{IoU}}; \\ y_k, k = \arg\max\limits_{j=1,N} \mathcal{T}'(\hat{\beta}_i, \beta_j) & \text{иначе.} \end{cases} \tag{3.2}$$

Затем каждому ξ_i присваивается значение точности:

$$p_i = \begin{cases} 0, & \text{если } \overline{y}_i = -1; \\ |\{\xi_j \,|\, \overline{y}_j = \overline{y}_i \text{ и } \hat{y}_j = \hat{y}_i\}| / |\{\xi_j \mid \hat{y}_j = \hat{y}_i\}| & \text{иначе} \end{cases} \tag{3.3}$$

и полноты

$$r_i = \begin{cases} 0, & \text{если } \overline{y}_i = -1; \\ |\{\xi_j \,|\, \overline{y}_j = \overline{y}_i \text{ и } \hat{y}_j = \hat{y}_i\}| / |\{\xi_j \mid \overline{y}_j = \overline{y}_i\}| & \text{иначе.} \end{cases} \tag{3.4}$$

Затем всему множеству Ξ присваиваются значения точности:

$$p(\Xi) = \sum_{i=1}^{M} p_i \tag{3.5}$$

и полноты

$$r(\Xi) = \sum_{i=1}^{M} r_i. \tag{3.6}$$

Зафиксировав все параметры, кроме $\tau = T_{\mathrm{dist}}$, рассмотрим эти значения точности и полноты как функции от $\tau \in [0, 2]$:

$$p = p(\tau) = p(\Xi) = p(\Xi(\Theta)) = p(\Xi(\Theta(\tau))), \tag{3.7}$$

$$r = r(\tau) = r(\Xi) = r(\Xi(\Theta)) = r(\Xi(\Theta(\tau))). \tag{3.8}$$

Таблица 4. Результаты для различных детекторов и экстракторов

Детектор	Экстрактор	AP, %
$\mathcal{D}_{\text{SSD}}$	$\mathcal{X}_{\text{PCB}}$	43.86
$\mathcal{D}_{\text{Faster}}$	$\mathcal{X}_{\text{PCB}}$	57.92
$\mathcal{D}_{\text{Mask}}$	$\mathcal{X}_{\text{PCB}}$	59.90
$\mathcal{D}_{\text{KP}}$	$\mathcal{X}_{\text{PCB}}$	60.11
$\mathcal{D}_{\text{SSD}}$	$\mathcal{X}_{\text{MGN}}$	46.23
$\mathcal{D}_{\text{Faster}}$	$\mathcal{X}_{\text{MGN}}$	58.56
$\mathcal{D}_{\text{Mask}}$	$\mathcal{X}_{\text{MGN}}$	61.01
$\mathcal{D}_{KP}$	$\mathcal{X}_{\text{MGN}}$	**61.15**

Затем получим значение целевой метрики системы:

$$\text{AP} = \int_0^1 p(\tau)dr(\tau) = \int_0^2 p(\tau)\frac{\partial r}{\partial \tau}(\tau)d\tau \tag{3.9}$$

как значение площади под кривой $\{(p(\tau), r(\tau)) \mid \tau \in [0,1]\}$, параметризованной одним параметром $\tau = T_{\text{dist}}$.

3.4. Базовый эксперимент. Положим $T_{\text{IoU}} = 0.3$, $\Delta = 60$ и $T_{\text{norm}} = 0.5$ как эвристики. В ходе базового эксперимента проведено сравнение рассматриваемых в работе моделей детектора и экстрактора. В качестве обучающей выборки для модели дескриптора взята обучающая выборка базы DukeMTMC. Информация о времени не использовалась. Результаты на тестовой выборке приведены в табл. 4.

3.5. Модификация трекера. Базовый трекер $\mathcal{T}_{\text{Box}}$ учитывает лишь информацию о координатах левого верхнего и правого нижнего углов положения объекта на изображении: $\beta' = \beta$. Однако детекторы $\mathcal{D}_{\text{Mask}}$ и $\mathcal{D}_{\text{KP}}$ помимо этих координат позволяют получить дополнительные данные – бинарную маску сегментации объекта $\mathbf{M}$ и координаты 17 ключевых точек человека с метками видимости $(\xi^{(k)}, \eta^{(k)}, \chi^{(k)})$ соответственно. Разработаны два новых трекера, применяющих эту информацию: $\mathcal{T}_{\text{Mask}}$ и $\mathcal{T}_{\text{KP}}$.

Трекер $\mathcal{T}_{\text{Mask}}$ работает аналогично $\mathcal{T}_{\text{Box}}$ с одним отличием – величина IoU вычисляется не по ограничивающим прямоугольникам, а по маскам сегментации объектов:

$$\mathcal{T}'_{\text{Mask}}(\beta'_t, \beta'_{t+1}) = \text{IoU}_{\text{Mask}}(\beta'_t, \beta'_{t+1}) = \frac{S_i^{(\text{Mask})}}{S_u^{(\text{Mask})}}, \tag{3.10}$$

где $S_i^{(\text{Mask})}$, $S_u^{(\text{Mask})}$ – площади объединения и пересечения масок сегментаций объектов. Такой трекер может работать только в комбинации с детектором $\mathcal{D}_{\text{Mask}}$, поскольку ему необходимы маски сегментации, которые способен выдавать только метод Mask-RCNN.

Трекер $\mathcal{T}_{\text{KP}}$ продолжает трек, если среднее смещение видимых ключевых точек, полученных с помощью $\mathcal{D}_{\text{KP}}$, не превышает $-T_{\Delta_{\text{KP}}}$:

$$\mathcal{T}'_{\text{KP}}(\beta'_t, \beta'_{t+1}) = \begin{cases} -\dfrac{\sum_{k=1}^{17} \chi_t^{(k)} \chi_{t+1}^{(k)} \left\| \mathbf{x}_{t+1}^{(k)} - \mathbf{x}_t^{(k)} \right\|}{\sum_{k=1}^{17} \chi_t^{(k)} \chi_{t+1}^{(k)}}, & \text{если} \quad \sum_{k=1}^{17} \chi_t^{(k)} \chi_{t+1}^{(k)} > 0, \\ -\infty & \text{иначе,} \end{cases} \tag{3.11}$$

где $\mathbf{x}_t^{(k)} = (\xi_t^{(k)}, \eta_t^{(k)})^{\text{T}}$. Такой трекер может использоваться только в комбинации с детектором $\mathcal{D}_{\text{KP}}$.

Таблица 5. Результаты для различных детекторов и трекеров

Детектор	Трекер	AP, %
$\mathcal{D}_{\text{Mask}}$	$\mathcal{T}_{\text{Box}}$	61.01
$\mathcal{D}_{\text{Mask}}$	$\mathcal{T}_{\text{Mask}}$	65.23
$\mathcal{D}_{\text{KP}}$	$\mathcal{T}_{\text{Box}}$	61.15
$\mathcal{D}_{\text{KP}}$	$\mathcal{T}_{\text{KP}}$	**67.88**

Таблица 6. Результаты для различных моделей и обучающих данных

Экстрактор	Обучающая выборка	AP, %
$\mathcal{X}_{\text{PCB}}$	**DukeMTMC**	65.93
$\mathcal{X}_{\text{PCB}}$	**MDB**	68.43
$\mathcal{X}_{\text{MGN}}$	**DukeMTMC**	67.88
$\mathcal{X}_{\text{MGN}}$	**MDB**	**71.03**

Таблица 7. Результаты для различных моделей и обучающих данных

Детектор	**Трекер**	**Экстрактор**	**Обучающая выборка**	**Время**	AP, %
$\mathcal{D}_{\text{Faster}}$	$\mathcal{T}_{\text{Box}}$	$\mathcal{X}_{\text{PSB}}$	**DukeMTMC**	–	57.92
$\mathcal{D}_{\text{Faster}}$	$\mathcal{T}_{\text{Box}}$	$\mathcal{X}_{\text{MGN}}$	**DukeMTMC**	–	58.56
$\mathcal{D}_{\text{KP}}$	$\mathcal{T}_{\text{Box}}$	$\mathcal{X}_{\text{MGN}}$	**DukeMTMC**	–	61.15
$\mathcal{D}_{\text{KP}}$	$\mathcal{T}_{\text{KP}}$	$\mathcal{X}_{\text{MGN}}$	**DukeMTMC**	–	67.88
$\mathcal{D}_{\text{KP}}$	$\mathcal{T}_{\text{KP}}$	$\mathcal{X}_{\text{MGN}}$	**MDB**	–	71.03
$\mathcal{D}_{\text{KP}}$	$\mathcal{T}_{\text{KP}}$	$\mathcal{X}_{\text{MGN}}$	**MDB**	–	**81.01**

Зафиксируем $T_{\text{IoU}}^{(\text{Mask})} = 0.3$ и $T_{\Delta_{\text{KP}}} = -20$ в качестве эвристик, величины $\Delta = 60$ и $T_{\text{norm}} = 0.5$ оставим неизменными с базового эксперимента. Зафиксируем экстрактор $\mathcal{X}_{\text{MGN}}$, обученный на DukeMTMC. Для каждого из детекторов $\mathcal{D}_{\text{Mask}}$, $\mathcal{D}_{\text{KP}}$ сравним его комбинацию с базовым трекером и модифицированным. Результаты даны в табл. 5.

3.6. Расширение обучающей выборки. Во всех экспериментах ранее использовались модели дескриптора PCB и MGN, веса которых обучены на тренировочной выборке DukeMTMC, однако в открытом доступе есть и другие размеченные базы. Предполагается, что добавление к DukeMTMC смеси из остальных баз данных разнообразит обучающую выборку и улучшит качество модели. Проверим это предположение: зафиксируем комбинацию $\mathcal{D}_{\text{KP}} + \mathcal{T}_{\text{KP}}$, постоянные $T_{\Delta_{\text{KP}}} = -20$, $\Delta = 60$ и $T_{\text{norm}} = 0.5$. Сравним модели PCB и MGN, обученные на DukeMTMC и MDB – смеси всех описанных баз. Результаты приведены в табл. 6.

3.7. Использование информации о времени. В большинстве сценариев слежения за людьми в общественных местах, во-первых, предполагается, что они не будут долго задерживаться в одном месте, во-вторых, имеется информация о времени съемки. Поэтому в работе предложена и реализована следующая модификация модуля принятия решения: в качестве запроса рассматривается не только дескриптор объекта $\mathbf{f}$, но и время t появления объекта на камере, то есть пара $(\mathbf{f}, t)$; в галерею информация о времени тоже сохраняется. Соответственно, вместо расстояния $\|\mathbf{f} - \mathbf{g}\| \in [0, 2]$ берется сходство $\text{Sim}(\mathbf{f}, \mathbf{g}) = (1 - \|\mathbf{f} - \mathbf{g}\|)/2 \in [0, 1]$; сходство пары объектов домножается на затухающую по времени экспоненту: $\text{Sim}'((\mathbf{f}, t), (\mathbf{g}, \tau)) = \text{Sim}(\mathbf{f}, \mathbf{g})e^{-\lambda|t-\tau|}$. Идея заключается в искуственном уменьшении сходства объектов, разнесенных по времени, причем чем сильнее они разнесены, тем сильнее уменьшается сходство. Параметр λ подобран

так, чтобы сходство объектов, отстоящих друг от друго по времени на 15 мин, уменьшалось вдвое: $\lambda = \ln 2/54000 \approx 10^{-5}$ (если время измерять в кадрах при частоте записи 60 кадров в секунду).

Сравнение результатов для конфигурации $\mathcal{D}_{\text{KP}} + \mathcal{T}_{\text{KP}} + \mathcal{X}_{\text{MGN}}$ с обучающей выборкой **MDB** без описанной модификации дает $AP = 71.03\%$, с модификацией получено $AP = 81.01\%$.

3.8. Результаты. Основное развитие системы от базовой конфигурации до итоговой приведено в табл. 7. Таким образом за счет модификации каждой части системы, интеграции частей между собой, а также использования общих идей удалось повысить точность на 20% относительно базовой конфигурации.

Заключение. Рассмотрена задача слежения за объектами по данным сети камер видеонаблюдения. Проведено полное исследование каждой составляющей этой задачи: детектирование объектов на изображении, повторная идентификация объектов при их неоднократном появлении в поле зрения нескольких камер, сравнение новых объектов с уже накопленными в базе данных и принятие решения о принадлежности каждого объекта какому-либо классу. В каждой подзадаче исследованы и реализованы лучшие по качеству на текущий момент методы. Выполнена интеграция всех используемых методов друг с другом в единую систему слежения, а также модификация всех частей этой системы по отдельности и в общем.

Поставлен вычислительный эксперимент на видео с восьми камер, расположенных на территории студенческого городка, длительностью 1 ч 25 мин. В качестве объектов рассматривались люди, целью было присвоить каждому человеку корректный ID. Сравнение результатов разработанной системы с базовым вариантом количественно подтверждает преимущества предлагаемых идей.

СПИСОК ЛИТЕРАТУРЫ

1. *Huang T., Russell S.J.* Object Identification in a Bayesian Context // Proc. 15th Intern. Joint Conf. Artificial Intelligence. Nagoya, Japan: ACM, 1997. P. 1276–1282.
2. *Wang X.* Intelligent Multi-Camera Video Surveillance: A Review // Pattern Recognition Letters. 2013. V. 34. 1. P.3–19.
https://doi.org/10.1016/j.patrec.2012.07.005
3. *Zajdel W., Zivkovic Z., Krose B.* Keeping Track of Humans: Have I Seen This Person Before? // Proc. 2005 IEEE Intern. Conf. Robotics and Automation. Barcelona, Spain: IEEE, 2005. P. 2081–2086.
https://doi.org/10.1109/ROBOT.2005.1570420.
4. *Gheissari N., Sebastian T.B., Hartley R.I.* Person Reidentification Using Spatiotemporal Appearance // Proc. 2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition. NY, USA: IEEE, 2006. P. 1528–1535.
https://doi.org/10.1109/CVPR.2006.223.
5. *Yi D., Lei Zh., Liao Sh. et al.* Deep Metric Learning for Person Re-identification // Proc. 22nd Intern. Conf. Pattern Recognition. Stockholm, Sweden: IEEE, 2014. P. 34–39.
https://doi.org/10.1109/ICPR.2014.16.
6. *Li W., Zhao R., Xiao T. et al.* DeepReID: Deep Filter Pairing Neural Network for Person Re-identification // Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE, 2014. P. 152–159.
https://doi.org/10.1109/CVPR.2014.27.
7. *Xu Y., Ma B., Huang R. et al.* Person Search in a Scene by Jointly Modeling People Commonness and Person Uniqueness // Proc. ACM Intern. Conf. Multimedia. Orlando, FL, USA: ACM, 2014. P. 937–940.
https://doi.org/10.1145/2647868.2654965.
8. *Zheng L., Yi Y., Hauptmann A.G.* Person Re-identification: Past, Present and Future // URL:
https://arxiv.org/pdf/1610.02984.pdf.
9. *Себряков Г.Г., Сошников В.Н., Кикин И.С., Ишутин А.А.* Алгоритм распознавания объектов на оптико-электронных изображениях наземных сцен, основанный на оценке ковариационных матриц градиентных функций поля яркости // Вестник компьютерных и информационных технологий. 2013. V. 109. № 7. С. 14–19.
10. *Визильтер Ю.В., Желтов С.Ю.* Использование проективных морфологий в задачах обнаружения и идентификации объектов на изображениях // Изв. РАН. ТиСУ. 2009. № 2. С. 125–138.
11. *Girshick R.B., Donahue J., Darell T. et al.* Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation // Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE, 2014. P. 580–587.
https://doi.org/10.1109/CVPR.2014.81.

12. *Girshick R.B.* Fast R-CNN // Proc. 2015 IEEE Intern. Conf. Computer Vision. Santiago, Chile: IEEE, 2015. P. 1440–1448.
https://doi.org/10.1109/ICCV.2015.169.
13. *Ren S., He K., Girshick R.B. et al.* Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. V. 39. № 6. P. 91–99.
https://doi.org/10.1109/TPAMI.2016.2577031
14. *He K., Gkioxari G., Dollár P. et al.* Mask R-CNN // Proc. 2017 IEEE Intern. Conf. Computer Vision. Venice, Italy: IEEE, 2017. P. 2980–2988.
https://doi.org/10.1109/ICCV.2017.322.
15. *Lin T.-Y., Maire M., Belongie S. et al.* Microsoft COCO: Common Objects in Context // URL:
https://arxiv.org/pdf/1405.0312.pdf.
16. *Liu W., Anguelov D., Erhan D. et al.* SSD: Single Shot MultiBox Detector // Proc. 2016 European Conf. Computer Vision. Amsterdam, The Netherlands: Springer, 2016. P. 21–37.
https://doi.org/10.1007/978-3-319-46448-0_2.
17. *Redmon J., Divvala S., Girshick R.B. et al.* You Only Look Once: Unified, Real-Time Object Detection // Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016. P. 779–788.
https://doi.org/10.1109/CVPR.2016.91.
18. *Uijlings J., Van de Sande K.E.A., Gevers T. et al.* Selective Search for Object Recognition // Intern. J. Computer Vision. 2013. V. 104. № 2. P. 154–171.
https://doi.org/10.1007/s11263-013-0620-5
19. *Krizhevsky A., Sutskever I., Hinton G. E.* ImageNet Classification with Deep Convolutional Neural Networks // Communications ACM. 2017. V. 60. 6. P. 84–90.
https://doi.org/10.1145/3065386
20. *Zeiler M.D., Fergus R.* Visualizing and Understanding Convolutional Networks // Proc. 2014 European Conf. Computer Vision. Zurich, Switzerland: Springer, 2014. P. 818–833.
https://doi.org/10.1007/978-3-319-10590-1_53.
21. *Simonyan K., Zisserman A.* Very Deep Convolutional Networks for Large-Scale Image Recognition // URL:
https://arxiv.org/pdf/1409.1556.pdf.
22. *He K., Zhang X., Sen S. et al.* Deep Residual Learning for Image Recognition // Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016. P. 770–778.
https://doi.org/10.1109/CVPR.2016.90.
23. *Xie S., Girshick R. B., Dollár P. et al.* Aggregated Residual Transformations for Deep Neural Networks // Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017. P. 1492–1500.
https://doi.org/10.1109/CVPR.2017.634.
24. *Zheng L., Bie Z., Sun Y. et al.* MARS: A Video Benchmark for Large-Scale Person Re-identification // Proc. 2016 European Conf. Computer Vision. Amsterdam, The Netherlands: Springer, 2016. P. 868–884.
https://doi.org/10.1007/978-3-319-46466-4_52.
25. *Sun Y., Zheng L., Yang Y. et al.* Beyond Part Models: Person Retrieval with Refined Part Pooling (and a Strong Convolutional Baseline) // Proc. 2018 European Conf. Computer Vision. Munich, Germany: Springer, 2018. P. 501–518.
https://doi.org/10.1007/978-3-030-01225-0_30.
26. *Wang G., Yuan Y., Chen X. et al.* Learning Discriminative Features with Multiple Granularities for Person Re-identification // Proc. 26th ACM Intern. Conf. Multimedia. Seoul, Republic of Korea: ACM, 2018. P. 274–282.
https://doi.org/10.1145/3240508.3240552.
27. *Hoffer E., Ailon N.* Deep Metric Learning Using Triplet Network // Proc. 2015 Intern. Workshop on Similarity-Based Pattern Analysis and Recognition. Copenhagen, Denmark: Springer, 2015. P. 84–92.
https://doi.org/10.1007/978-3-319-24261-3_7.
28. *Matsukawa T., Okabe T., Suzuki E. et al.* Hierarchical Gaussian Descriptor for Person Re-identification // Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016. P. 1363–1372.
https://doi.org/10.1109/CVPR.2016.152.
29. *Zheng L., Shen L., Tian L. et al.* Scalable Person Re-identification: A Benchmark // Proc. 2015 IEEE Intern. Conf. Computer Vision. Santiago, Chile: IEEE, 2015. P. 1116–1124.
https://doi.org/10.1109/ICCV.2015.133.
30. *Hermans A., Beyer L., Leibe B.* In Defense of the Triplet Loss for Person Re-identification // URL:
https://arxiv.org/pdf/1703.07737.pdf.
31. *Almaz'an J., Gajic B., Murray N. et al.* Re-ID Done Right: Towards Good Practices for Person Re-identification // URL: https://arxiv.org/pdf/1801.05339.pdf.
32. *Zheng M., Karanam S., Radke R.J.* Measuring the Temporal Behavior of Real-World Person Re-identification // URL: https://arxiv.org/pdf/1808.05499.pdf.

33. *Wei L., Zhang S., Gao W. et al.* Person Transfer GAN to Bridge Domain Gap for Person Re-identification // Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018. P. 79–88.
https://doi.org/10.1109/CVPR.2018.00016.

34. *Karanam S., Gou M., Wu Z. et al.* A Systematic Evaluation and Benchmark for Person Re-identification: Features, Metrics and Datasets // IEEE Trans. Pattern Analysis and Machine Intelligence. 2019. V. 41. № 3. P. 523–536.
https://doi.org/10.1109/TPAMI.2018.2807450.

35. *Ristani E., Solera F., Zou R.S. et al.* Performance Measures and a Data Set for Multi-target, Multi-camera Tracking // Proc. 2016 European Conf. Computer Vision. Amsterdam, The Netherlands: Springer, 2016. P. 17–35.
https://doi.org/10.1007/978-3-319-48881-3_2

36. *Xiao T., Li S., Wang B. et al.* Joint Detection and Identification Feature Learning for Person Search // Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017. P. 3346–3355.
https://doi.org/10.1109/CVPR.2017.360.

37. *Zheng L., Zhang H., Sun S. et al.* Person Re-identification in the Wild // URL:
https://arxiv.org/pdf/1604.02531.pdf.

38. *Ma L., Liu H., Hu L. et al.* Orientation Driven Bag of Appearances for Person Re-identification // URL:
https://arxiv.org/pdf/1605.02464.pdf.

39. *Zheng S., Zhang J., Huang K. et al.* Robust View Transformation Model for Gait Recognition // Proc. 2011 18th IEEE Intern. Conf. Image Processing. Brussels, Belgium: IEEE, 2011. P. 2073–2076.
https://doi.org/10.1109/ICIP.2011.6115889.

40. *Kawanishi Y., Wu Y., Mukunoki M. et al.* Shinpuhkan2014: A Multi-camera Pedestrian Dataset for Tracking People across Multiple Cameras // URL:
http://www.mm.media.kyoto-u.ac.jp/wp-content/uploads/2014/05/shinpuhkan2014dataset.pdf.

41. *Figueira D., Taiana M., Nambiar A.M. et al.* The HDA+ Data Set for Research on Fully Automated Re-identification Systems // Proc. 2014 European Conf. Computer Vision. Zurich, Switzerland: Springer, 2014. P. 241–255.
https://doi.org/10.1007/978-3-319-16199-0_17.

42. *Layne R., Hospedales T.M., Gong S.* Investigating Open-World Person Re-identification Using a Drone // Proc. 2014 European Conf. Computer Vision. Zurich, Switzerland: Springer, 2014. P. 225–240.
https://doi.org/10.1007/978-3-319-16199-0_16.

43. *Wang T., Gong S., Zhu X. et al.* Person Re-identification by Discriminative Selection in Video Ranking // IEEE Trans. Pattern Analysis and Machine Intelligence. 2016. V. 38. 12. P. 2501–2514.
https://doi.org/10.1109/TPAMI.2016.2522418.

44. *Das A., Chakraborty A., Roy-Chowdhury A.K.* Consistent Re-identification in a Camera Network // Proc. 2014 European Conf. Computer Vision. Zurich, Switzerland: Springer, 2014. P. 330–345.
https://doi.org/10.1007/978-3-319-10605-2_22.

45. *Li W., Wang X.* Locally Aligned Feature Transforms Across Views // Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition. Portland, OR, USA: IEEE, 2013. P. 3594–3601.
https://doi.org/10.1109/CVPR.2013.461.

46. *Li W., Zhao R., Wang X.* Human Reidentification with Transferred Metric Learning // Proc. 2012 Asian Conf. Computer Vision. Daejeon, Korea: Springer, 2012. P. 31–44.
https://doi.org/10.1007/978-3-642-37331-2_3.

47. *Bialkowski A., Denman S., Sridharan S. et al.* A Database for Person Re-identification in Multi-camera Surveillance Networks // Proc. 2012 Intern. Conf. Digital Image Computing Techniques and Applications. Fremantle, WA, Australia: IEEE, 2012. P. 1–8.
https://doi.org/10.1109/DICTA.2012.6411689.

48. *Martinel N., Micheloni C.* Re-identify People in Wide Area Camera Network // Proc. 2012 IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops. Providence, RI, USA: IEEE, 2012. P. 31–36.
https://doi.org/10.1109/CVPRW.2012.6239203.

49. *Wang S., Lewandowski M., Annesley J. et al.* Re-identification of Pedestrians with Variable Occlusion and Scale // Proc. 2011 IEEE Intern. Conf. Computer Vision Workshops. Barcelona, Spain: IEEE, 2011. P. 1876–1882.
https://doi.org/10.1109/ICCVW.2011.6130477.

50. *Hirzer M., Beleznai C., Roth P.M. et al.* Person Re-identification by Descriptive and Discriminative Classification // Proc. 2011 Scandinavian Conf. Image Analysis. Ystad, Sweden: Springer, 2011. P. 91–102.
https://doi.org/10.1007/978-3-642-21227-7_9.

51. *Baltieri D., Vezzani R., Cucchiara R.* 3DPeS: 3D People Dataset for Surveillance and Forensics // Proc. 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding. Scottsdale, AZ, USA: ACM, 2011. P. 59–64.
https://doi.org/10.1145/2072572.2072590.

52. *Cheng D.S., Cristani M., Stoppa M. et al.* Custom Pictorial Structures for Re-identification // Proc. 2011 British Machine Vision Conference. Dundee, UK: BMVA Press, 2011. P. 1–11.
https://doi.org/10.5244/C.25.68.

53. *Loy C.C., Liu C., Gong S.* Person Re-identification by Manifold Ranking // Proc. 2013 IEEE Intern. Conf. Image Processing. Melbourne, Australia: IEEE, 2013. P. 3567–3571.
https://doi.org/10.1109/ICIP.2013.6738736.

54. *Zheng W., Gong S., Xiang T.* Associating Groups of People // Proc. 2009 British Machine Vision Conference. London, UK: BMVA Press, 2009. P. 1–11.
https://doi.org/10.5244/C.23.23.

55. *Schwartz W.R., Davis L.S.* Learning Discriminative Appearance-Based Models Using Partial Least Squares // Proc. 2009 XXII Brazilian Sympos. Computer Graphics and Image Processing. Rio de Janiero, Brazil: IEEE, 2009. P. 322–329.
https://doi.org/10.1109/SIBGRAPI.2009.42.

56. *Gray D., Tao H.* Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features // Proc. 2008 European Conf. Computer Vision. Marseille, France: Springer, 2008. P. 262–275.
https://doi.org/10.1007/978-3-540-88682-2_21.

57. *Felzenszwalb P.F., Girshick R.B., McAllester D.A. et al.* Visual Object Detection with Deformable Part Models // Communications ACM. 2013. V. 56. 9. P. 97–105. DOI: http://dx.doi.org/. justification=raggedright,singlelinecheck=off
https://doi.org/10.1145/2500468.2494532.