

УДК 004.931

ЭКСПЕРИМЕНТАЛЬНЫЙ ПРОТОТИП ОТКРЫТОЙ СИСТЕМЫ ПОВТОРНОЙ ИДЕНТИФИКАЦИИ ЛЮДЕЙ ПРИ МНОГОКАМЕРНОМ ВИДЕОНАБЛЮДЕНИИ

С.А. ИГНАТЬЕВА, Н.А. ТОМАШЕВИЧ, А.А. ГОЛУБЕНОК, Р.П. БОГУШ

*Полоцкий государственный университет имени Евфросинии Полоцкой, Республика Беларусь**Поступила в редакцию 20 марта 2023*

Аннотация. Рассмотрен алгоритм реидентификации людей в распределенной системе видеонаблюдения с использованием сверточных нейронных сетей. Разработан экспериментальный прототип системы реидентификации, позволяющий на кадрах с камер видеонаблюдения выполнять повторную идентификацию людей, формировать набор изображений, упорядоченных в соответствии с идентификаторами для каждого обнаруженного человека. Для обнаружения людей использовалась СНС YOLOv5, для реидентификации применяется DenseNet-121. Выполнена оценка точности работы системы реидентификации в метриках *precision* (точность) и *recall* (полнота).

Ключевые слова: реидентификация, сверточные нейронные сети, информационная система.

Введение

Одной из актуальных задач компьютерного зрения является повторная идентификация людей (реидентификация), предполагающая поиск заданного человека по изображению на кадрах с пространственно-разнесенных камер видеонаблюдения. В зависимости от входных данных выделяют закрытые (close-world) и открытые (open-world) системы. Для открытых систем реидентификации в качестве входных данных используются неразмеченные видео, на которых необходимо обнаруживать людей и динамически формировать галерею изображений, среди которых осуществляется поиск по запросу. Поэтому процесс реидентификации в открытой системе будет состоять из двух этапов. На первом этапе выполняется детектирование людей на изображениях с применением сверточных нейронных сетей (СНС). После того, как человек на кадре видеопоследовательности с одной из камер обнаружен, другая СНС необходима для извлечения отличительных признаков человека, на основе которых осуществляется поиск в галерее. При наличии в ней других изображений этого человека, обнаруженному присваивается соответствующий идентификатор (ID). В противном случае предполагается, что человек впервые попал в поле зрения камеры, и его изображению присваивается новый ID. На этом этапе важное значение будет иметь качество обучения СНС и эффективность извлекаемых ею признаков. В зависимости от поставленной задачи, результатом реидентификации людей может быть отображение всех изображений искомого человека с указанием даты и места их получения или формирование набора данных для всех обнаруженных на видео пешеходов, которые упорядочены согласно их идентификаторам.

Алгоритм повторной идентификации людей

Разработан алгоритм для реидентификации людей, который состоит из основных этапов:

- разделение входных видеоданных на кадры;
- обнаружение людей;
- проверка корректности обнаружений;
- формирование вектора признаков для изображения каждого человека;

- установление соответствия между изображениями людей на кадрах видеопоследовательностей с заданного набора видеокамер;
- присвоение идентификатора человеку на изображении;
- формирование набора изображений обнаруженных и идентифицированных людей (сохранение в файл, располагающийся в папке с соответствующим ID).

На каждом кадре, с использованием СНС выполняется детектирование людей. Действительными считаются те изображения людей, для которых пороговая степень уверенности СНС составляет 60 % и соотношение высоты ограничительного прямоугольника к ширине меньше 2,5, что позволяет не учитывать обнаружения, на которых фигура человека в кадре полностью не отображается.

Для каждого обнаружения с использованием другой СНС вычисляется вектор признаков, отображающий отличительные особенности человека на изображении. Для установления соответствия при реидентификации вычисляется расстояние Эвклида:

$$d_{p,q} = \sum_{i=1}^n (p_i + q_i)^2,$$

где q_i – дескриптор запроса, p_i – дескрипторы изображений ранее обнаруженных людей.

Все изображения группируются и сохраняются в папках, имена которых определяются соответствующими идентификаторами. Для каждого нового изображения человека осуществляется поиск в галерее такого изображения, с которым расстояние между признаками минимально, после чего предполагается, что их идентификаторы одинаковы. Если расстояние между признаками больше установленного порога Thr , то считается, что этот человек ранее не присутствовал на кадрах с камеры видеонаблюдения и ему присваивается новый уникальный идентификатор.

Программная реализация прототипа системы реидентификации

Программная реализация разработана на основе представленного алгоритма и состоит из двух модулей: модуль, осуществляющий обработку видео, обнаружение людей и их реидентификацию, реализованный на Python; пользовательский графический интерфейс на C++. В качестве входных данных используются видео, полученные из различных источников (IP-камеры, Web-камеры, видео из файла), которые разбиваются на отдельные кадры с заданным интервалом и приводятся к единому размеру. Для распределения вычислительных ресурсов применяется многопоточная обработка данных. Схема организации потоков представлена на рисунке 1: поток UI (пользовательского интерфейса) и вычислительный поток, осуществляющий все ресурсоемкие вычисления, такие как обработка (захват видеокадров, изменение размера кадра), детектирование людей на изображениях, выделение признаков и непосредственно реидентификация. Оба потока связаны с хранилищем, которое включает два поддерживающих многопоточность контейнера, реализованные в классах AtomicList и AtomicCell.

Первый из них, AtomicList совмещает свойства списка и массива и используется для хранения информации о каждом обнаруженном человеке: его изображение, вектор признаков, идентификатор, информация о номере камеры (видео) и времени получения кадра. Позволяет нескольким потокам читать из него и добавлять новые данные одновременно, требуя при этом меньшего числа переходов по указателям для доступа к нужному элементу по сравнению с обычным списком.

Второй класс AtomicCell используется для передачи текущего кадра видео для вывода на экран и позволяет записывать или считывать только один новый кадр с обнаруженными на нем людьми. AtomicCell содержит счетчик, гарантирующий что кадр не будет прочитан, если между началом и окончанием чтения произошла запись. Для этого перед и после записи счетчик увеличивается на 1. Если в момент начала чтения значение счетчика нечетное, т.е. запись началась, но не завершилась, или не совпадает с показателем счетчика до начала и после окончания чтения, то кадр пропускается.

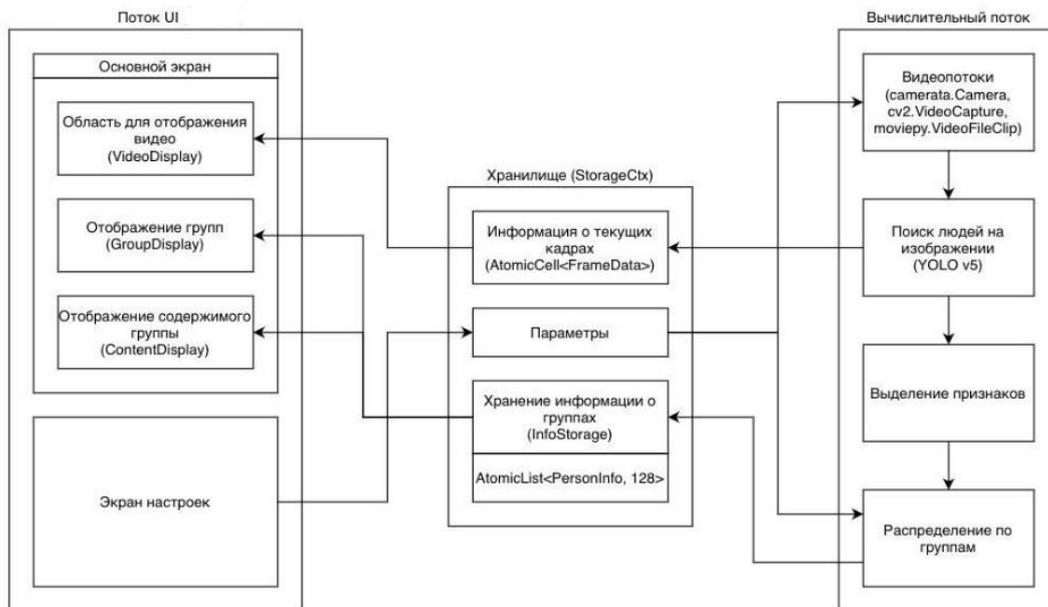


Рис. 1. Схема организации многопоточной обработки

Основной функцией системы является повторная идентификация людей из нескольких видео и формирование набора изображений, разделенных на группы по идентификаторам (*ID*). В качестве источника данных могут выступать видео, полученные с Web- или IP-камер, а также видеоматериалы, полученные ранее с любых других средств видеофиксации и сохраненные в наиболее распространенных форматах: OGV, MP4, MPEG, AVI, MOV, MKV и GIF. Библиотека, используемая для захвата видеопотока, определяется в зависимости от источника видео. При обработке видео из файла для извлечения кадров используется *moviepy* для Python [1]. Проверка наличия подключенных WEB-камер и захват видеопотока с них выполняется с помощью *camerata* [2] для Python. Кадры с IP-камер извлекаются с применением библиотеки компьютерного зрения *OpenCV* [3]. Для передачи видеопотока с IP-камер используется протокол *rtsp* (real time streaming protocol, потоковый протокол реального времени).

Для обнаружения людей используется СНС версии *YOLOv5x6* [4], архитектура которой описана на языке Python и предложена компанией *Ultralytics* [5]. Для повторной идентификации используется СНС *DenseNet-121* [6] в реализации *pyTorch* и файл весовых коэффициентов *net_last.pth* [7].

Для увеличения скорости обработки изображений применяется программно-аппаратная архитектура параллельных вычислений для операций с СНС, позволяющая повысить производительность за счет возможностей GPU *Nvidia* с технологией *CUDA*. Для ускорения работы на GPU используется библиотека примитивов для нейронных сетей *cuDNN*, доступная зарегистрированным на сайте *Nvidia* [8] разработчикам.

Для разработки пользовательского интерфейса используется кроссплатформенная библиотека *SFML* [9]. Поддерживает распространенные операционные системы (*Windows*, *Linux*, *MacOS*) и большое число языков программирования (*C* и *.Net* подобные языки программирования, *Java*, *Python*, *Ruby*, *Go* и др.). *SFML* включает 5 основных модулей: *System* (для управления временем и потоками), *Window* (для управления окнами и потоками), *Graphics* (отображение графических примитивов и изображений), *Audio* (интерфейс для управления звуком) и *Network* (для сетевых приложений). В процессе разработки прототипа системы реидентификации использовались первые три модуля: *System*, *Window* и *Graphics*.

Интерфейс приложения разбит на два экрана: основной, на котором отображаются видео с обнаруженными людьми, и окно с настройками, вызываемое нажатием клавиши «пробел». Основной экран содержит несколько областей. На рисунке 2 (1) отмечена область из двух чисел в формате n/m , где m – количество подключенных источников видео, n – порядковый номер источника. Стрелки позволяют выбрать какое из видео отображается на экран в области (2) на

рисунке 2. На кадр накладываются прямоугольные рамки, в каждой из которых находится обнаруженный человек. Все люди идентифицируются по внешнему виду и сортируются по группам, согласно их ID. Группы идентифицированных людей располагаются в области (4) на рисунке 2. Изучить содержимое интересующей группы можно нажатием левой кнопки мыши на цель в области 2 в содержимое ограничительного прямоугольника или в области (4) выбрав интересующую группу. На рисунке 2 (5) отображена выбранная группа, а ее содержимое можно изучить в (3). Наведение курсора на интересующее изображение человека выводит в нижнем левом углу экрана (рис.2 (6)) номер источника видео и время получения кадра.

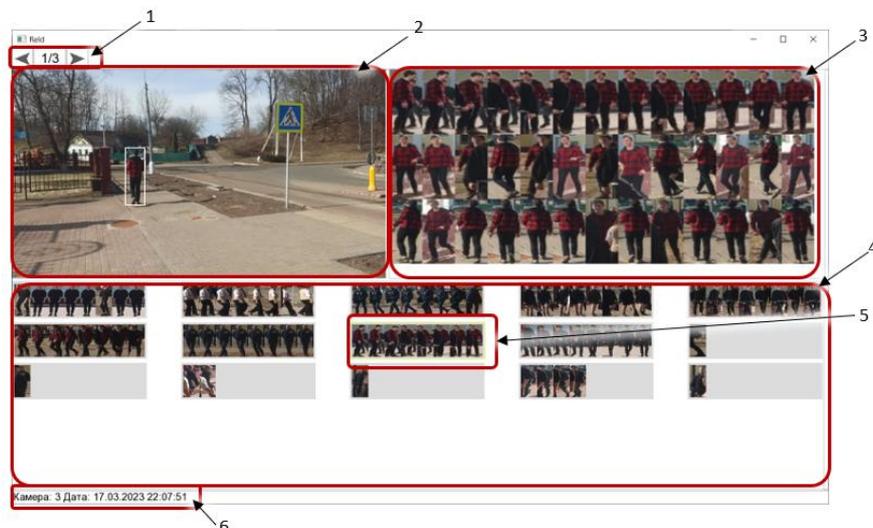


Рис. 2. Интерфейс прототипа системы повторной идентификации человека. Основной экран. 1 – область управления отображением кадров с интересующей камеры; 2 – отображение видео; 3 – содержимое группы изображений искомого человека; 4 – отображение всех идентифицированных людей упорядоченных согласно их ID; 5 – подсветка выбранной группы интересующего человека; 6 – отображение номера камеры и времени получения кадра для изображения, на который наведен курсор мыши в области 3

Все полученные и идентифицированные изображения людей сохраняются в отдельные папки, именами которым служит значение соответствующего идентификатора. В каждой соответствующей папке располагаются все изображения людей, которые предположительно принадлежат одному и тому же человеку. Размер изображений выбирается пользователем из значений $[32 \times 64]$, $[64 \times 128]$ или $[128 \times 256]$ в окне настроек. Кроме выбора размера сохраняемых изображений возможно так же изменить порог, при котором человеку будет присваиваться значение нового ID; добавить источник видео, для чего необходимо указать его расположение (для IP-камеры или видео из файла). Видео с Web-камеры добавляется к обрабатываемым в момент подключения.

Результаты экспериментов

Для определения наиболее эффективного значения порога Thr , позволяющего классифицировать изображения людей по идентификаторам проводилось три эксперимента. В первом использовалось три двадцатисекундных видео ролика, на которых присутствовало 5 человек. Во втором – три видео, на каждом из которых 7 человек, длительностью 10 секунд. В третьем эксперименте обрабатывалось три видео для 9 человек, продолжительностью 20 секунд. Примеры кадров для каждого эксперимента представлены на рисунке 3. Результаты приведены в таблице 1. Точность работы алгоритма реидентификации оценивалась по метрикам *precision* (точность) и *recall* (полнота):

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

$$recall = \frac{TP}{TP + FN},$$

где TP – верноположительные предсказания, FP – ложноположительные, FN – ложноотрицательные. $Precision$ отражает долю объектов (людей), названных классификатором положительными, и при этом действительно являющихся таковыми, $recall$ – отражает какая доля объектов положительного класса из всех верных ответов найдена алгоритмом.



Рис. 3. Кадры из видео, использованные при проведении экспериментов

При расчете метрик не учитывались ошибки детектора. Тестирование выполнялось на персональном компьютере с характеристиками: Intel Core i5 3.11 GHz, 16 Gb RAM, Nvidia GeForce RTX-3060 6 Gb.

Табл. 1. Оценка эффективности работы прототипа системы реидентификации

Особенности эксперимента	Метрика	Значение Thr									
		0,98	0,96	0,94	0,92	0,9	0,8	0,7	0,6	0,5	0,4
3 видео по 20 секунд, 5 человек	$Precision$	0,94	0,94	0,94	0,94	0,94	0,94	1	1	1	1
	$Recall$	0,92	0,92	0,77	0,77	0,74	0,64	0,47	0,46	0,44	0,39
3 видео по 10 секунд, 7 человек	$Precision$	0,98	0,99	1	1	1	1	1	1	1	1
	$Recall$	0,94	0,95	0,96	0,96	0,96	0,93	0,93	0,77	0,67	0,61
3 видео по 20 секунд, 9 человек	$Precision$	0,67	0,78	0,85	0,98	0,98	0,99	0,99	1	1	1
	$Recall$	0,69	0,77	0,80	0,90	0,82	0,79	0,78	0,71	0,65	0,59

Анализ таблицы 1 показывает, что увеличение порогового значения приводит к уменьшению точности в метрике $Precision$. Это связано с тем, что разные люди, имея какие-либо схожие черты, объединяются под одним идентификатором. Чем больше количество человек на видео, тем ниже $Precision$ и $Recall$ при высоких значениях Thr . При этом, если на кадрах небольшое число людей, то увеличение Thr позволяет повысить значение $Recall$, т.к. один и тот же человек с разных ракурсов реже оказывается разделен на разные группы по ID . Уменьшение Thr приводит к понижению показателей точности и полноты. Это связано с тем, что изображениям людей, полученных с разных камер с отличающимися характеристиками, при различных условиях освещенности, ракурсах человека, будут присваиваться новые идентификаторы. Экспериментально установлено, что наиболее эффективен выбор Thr в диапазоне от 0,92 до 0,94.

Заключение

Разработанный экспериментальный прототип открытой системы реидентификации позволяет обрабатывать видео с трех камер видеонаблюдения, при присутствии на кадре одновременно до 9 человек. Следует отметить, что количество подключаемых камер ограничивается доступными вычислительными ресурсами. Приложение отображает результат повторной идентификации на экран с указанием даты и места, определяемого по номеру камеры, получения каждого изображения обнаруженного человека, формирует набор данных с идентифицированными людьми. Программное обеспечение может быть использовано также для тестирования алгоритмов реидентификации.

EXPERIMENTAL PROTOTYPE OF OPEN-WORLD PERSON RE-IDENTIFICATION SYSTEM IN MULTICAMERA VIDEO SURVEILLANCE

S.A. IHNATSYEVA, N.A. TOMASHEVICH, A.A. HALUBIONAK, R.P. BOHUSH

Abstract. An algorithm for person re-identification in a distributed video surveillance system using convolutional neural networks is considered. Experimental prototype of re-identification system has been developed that allows people to be detected and re-identification on frames from video surveillance cameras, to form images set sorted according to identifiers for each detected person. CNN YOLOv5 was used for people detection, DenseNet-121 for re-identification. The re-identification system accuracy in *precision* and *recall* metrics was assessed.

Keywords: re-identification, convolutional neural networks, information system.

Список литературы

1. MoviePy. [Электронный ресурс]. URL: <https://github.com/Zulko/moviepy>.
2. Camerata. [Электронный ресурс]. URL: <https://github.com/IntQuant/Camerata>.
3. OpenCV. [Электронный ресурс]. URL: <https://opencv.org>.
4. YOLOv5. [Электронный ресурс]. URL: <https://github.com/ultralytics/yolov5>.
5. Ultralytics. [Электронный ресурс]. URL: <https://ultralytics.com>.
6. Huang, G., Liu, Z., Weinberger, K.Q. // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). P. 2261-2269.
7. Bohush, R., Ihnatsyeva, S., Ablameyko, S. // Machine Graphics and Vision. 2022. Vol. 31(1/4). P. 93–109.
8. Nvidia. [Электронный ресурс]. URL: <https://www.nvidia.com>.
9. SFML. [Электронный ресурс]. URL: <https://www.sfml-dev.org>.