

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ МОНИТОРИНГА ЛЮДЕЙ В ВИДЕОПОТОКЕ

Полупанов Алексей Александрович

к.т.н., доцент

Марков Алексей Размикович

ФГБОУ ВО «Кубанский государственный университет»

Аннотация: приведены основные элементы создания интеллектуальных систем мониторинга людей в видеопотоке, использующих технологии компьютерного зрения. Рассмотрены каскадные классификаторы признаков в качестве инструмента для детектирования людей на изображении. В качестве основы для каскадных классификаторов рассмотрены примитивы Хаара и локальные бинарные шаблоны. Разработана интеллектуальная масштабируемая система, осуществляющая детектирование движения и подсчет людей в видеопотоке с камер видеонаблюдения.

Ключевые слова: OpenCV, C++, FFmpeg, RTSP, local binary patterns, LBP, cascade classifier, распознавание людей, видеонаблюдение, optical flow.

DEVELOPMENT OF AN INTELLECTUAL MONITORING SYSTEM OF PEOPLE IN THE VIDEO FLOW

Polupanov Alexey Aleksandrovich

Markov Alexey Razmikovich

Abstract: The main elements of creating intelligent systems for monitoring people in a video stream using computer vision technologies are presented. Cascade feature classifiers are considered as a tool for detecting people in an image. As a basis for cascade classifiers, Haar primitives and local binary patterns are considered. An intelligent scalable system has been developed that performs motion detection and people counting in a video stream from surveillance cameras.

Key words: OpenCV, C++, FFmpeg, RTSP, local binary patterns, LBP, cascade classifier, recognition of people, video surveillance, optical flow.

Компьютерное зрение – это технология создания систем, использующих информацию, полученную из изображений. Области применения компьютерного зрения являются производственная промышленность, медицина, транспорт, оборонная промышленность. Другой важной сферой применения компьютерного зрения является видеонаблюдение. Создание автоматизированных систем видеонаблюдения обосновано тем, что позволяет

значительно уменьшить затраты человеческого труда на ведение наблюдения за территорией, а также повысить эффективность процесса и исключить ошибки, вызванные человеческим фактором.

В качестве основы для создания такой системы использован язык C++ и библиотека OpenCV. Данная библиотека, имеющая открытый исходный код, представляет собой набор алгоритмов компьютерного зрения и обработки изображений, разработанный на C/C++. Данная библиотека также является стандартным интерфейсом при создании приложений в области компьютерного зрения [1].

Важным элементом системы видеонаблюдения является возможность детектирования людей в кадре для дальнейшего использования этой информации, например для уведомления охранника или для подсчета прошедших людей за определенный промежуток времени. Эту информацию можно использовать для сбора статистики о посещаемости объектов.

В качестве инструмента для решения данной задачи рассмотрены системы, основанные на использовании сверточных нейронных сетей или на использовании каскадных классификаторов признаков. Важным требованием к разрабатываемой системе наблюдения является возможность её работы в режиме реального времени. Таким образом, высокая скорость интеллектуальной обработки кадров выступает ключевым фактором. По этой причине принято решение использовать каскадные классификаторы признаков вместо нейронных сетей, так как первые показывают значительно более высокую скорость работы и позволяют применять их в системах реального времени.

В задачах классификации классификатором называется функция, выносящая решение, к какому из заранее известных классов относится исследуемый объект. В нашем случае есть лишь один заранее известный класс людей, и задача классификатора принимать решение относится ли объект к этому классу или нет.

Для решения сложных задач классификации используется технология бустинга. Бустинг — комплекс методов, способствующих повышению точности аналитических моделей. Его принцип заключается в улучшении простых классификаторов для достижения необходимого уровня точности. Каскады классификаторов являются одним из примеров технологии бустинга. Ее суть заключается в использовании совокупности классификаторов, применяемых последовательно. При этом каждый следующий классификатор использует информацию, полученную предыдущими.

В качестве признаков, являющихся основой для классификатора рассмотрены примитивы Хаара и локальные бинарные шаблоны. Каскадные классификаторы, основанные на применении локальных бинарных шаблонов, превосходят в скорости работы свои аналоги, основанные на примитивах Хаара. Так как важным критерием разрабатываемой системы является быстродействие, в качестве основы для классификаторов выбраны локальные бинарные шаблоны.

Основной идеей локальных бинарных шаблонов является использование в качестве характеристики для пикселя значений окружающих его пикселей, и последующее вычисление средних значений этих характеристик в некоторой области [2]. На рисунке ниже приведены основные локальные бинарные шаблоны, используемые для поиска на изображениях (рис. 1).

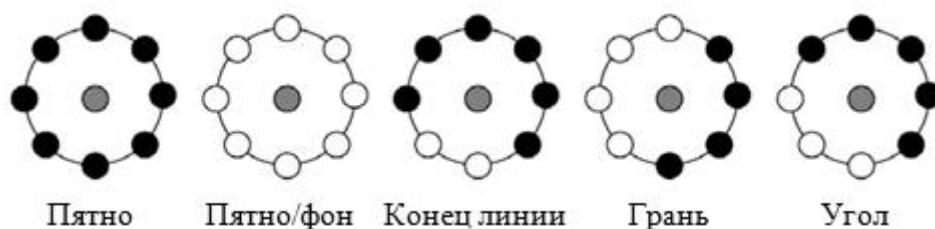


Рис. 1. Локальные бинарные шаблоны

Для создания каскадного классификатора использованы средства библиотеки OpenCV. Предварительно подготовлены обучающие наборы

положительных и отрицательных примеров. Далее к положительным примерам с помощью утилиты `opencv_createsamples` применены автоматические изменения наклона, яркости и фона для расширения объема обучающей выборки. После этого с помощью утилиты `opencv_traincascade` применялись различные параметры обучения, такие как размеры изображений, объем и соотношение выборки, количество этапов обучения, требуемая точность на каждом этапе, для достижения необходимых критериев классификатора [3]. В ходе работы для получения классификатора, обладающего требуемым качеством и скоростью обнаружения обучено около двадцати каскадных классификаторов с различными параметрами.

Для повышения быстродействия разрабатываемой системы, в качестве инструмента для получения и записи видео использован фреймворк FFmpeg. Данный фреймворк, имеющий открытый исходный код, предоставляет разработчику широкий функционал для работы с видео, позволяющий оптимизировать алгоритм работы программы, путем детальной настройки процесса взаимодействия с видео [4, с. 138].

Учитывая необходимость максимального повышения производительности разрабатываемой системы, принято решение добавить в систему предварительный анализ кадров на наличие движения на нем. В том случае, когда движение обнаружено на последовательных изображениях, обрабатываемые кадры передаются для анализа каскадному классификатору признаков. Это обосновано тем, что процесс обработки кадра классификатором значительно превосходит по затратам времени процесс обнаружения движения. При этом нет никакой необходимости осуществлять детектирование людей, в том случае если кадр не изменяется, а значит и человек не может там появиться. Детектор движения реализован с использованием функционала библиотеки OpenCV.

Основным элементом системы, осуществляющим управление остальными компонентами, является класс `VideoRecorder`. В главном элементе программы

создается дополнительный поток, в котором будет вестись обработка видеоданных с камеры. Для этого создается экземпляр указанного класса. После создания экземпляра данного класса вызывается его метод `StartWriting()`. Далее в этом объекте происходит инициализация класса `Reader`, реализованного с помощью фреймворка `FFMPEG`.

После этого начинается процесс обработки видеоданных с камеры. Создается дополнительный поток, осуществляющий получение данных с камеры с помощью объекта `Reader` и сохранение их в буфер, с которым работает основная часть объекта `VideoRecorder`. В основном потоке обработки видео, данные извлекаются из буфера с фиксированной периодичностью. Если включена функция детектирования людей с предварительным детектированием движения, то полученная структура `AVPacket` декодируется в изображение в структуру `AVFrame`, из которой далее формируется элемент `cv::Mat`.

Далее это изображение передается на анализ детектору движения. Если на последовательных трех кадрах обнаруживается движение, то изображение передается каскадному классификатору, предварительному обученному и загруженному из файла `cascade.xml`. Данный классификатор обрабатывает изображение, передаваемое ему в качестве параметра в метод `detectMultiScale()` [5]. Результатом работы классификатора является решение о том, есть ли на изображении люди, и если есть, то по каким координатам они располагаются.

Далее эти координаты сохраняются в массив. При обработке следующего кадра, применяется функция библиотеки `OpenCV` для определения оптического потока `calcOpticalFlowPyrLK()`, в которую передаются два последовательных кадра, и координаты интересующих объектов на первом кадре [6, с. 384]. Данная функция вычисляет оптический поток для заданного набора точек, используя пирамидальный метод Лукаса-Канаде. Результатом работы функции является массив координат, определяющий, по каким координатам находятся объекты первого изображения на втором изображении. Таким образом, если человек пересек середину кадра, то программа увеличивает счетчик

прошедших людей. С определенной периодичность в лог-файл записывается информация о текущем количестве посчитанных людей.

В главной программе можно параллельно обрабатывать данные с нескольких камер в высоком разрешении. При тестировании велась обработка данных с трех камер, передающих видео в формате 4к. Такая производительность достигается за счет использования оптимальных алгоритмов обработки видео. На рисунке ниже приведена структура программы (рис. 2).

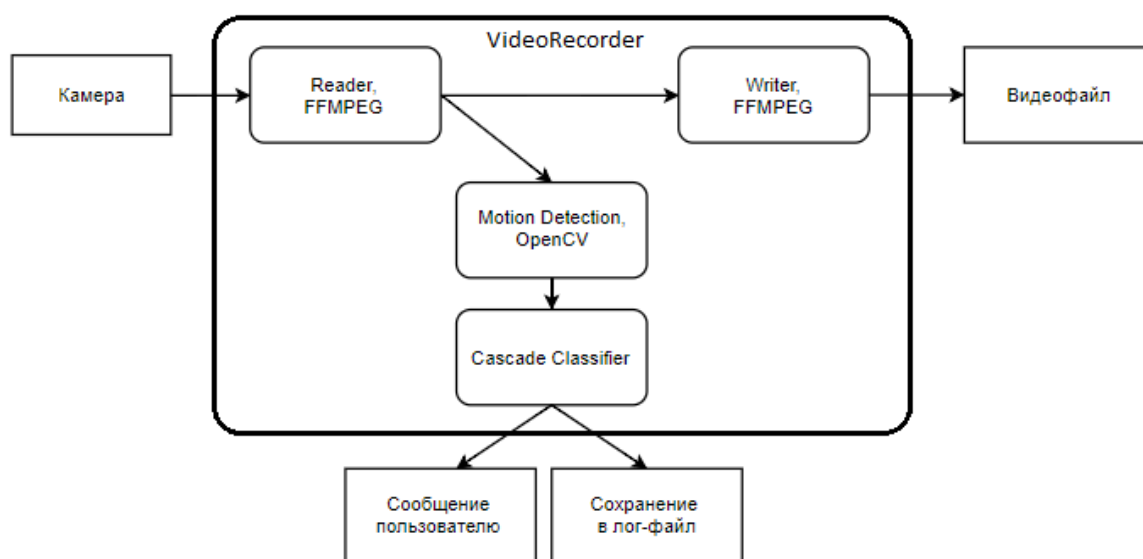


Рис. 2. Структура разработанной системы

Для разработки программного продукта изучено такое средство разработки программного обеспечения как OpenCV. Рассмотрены различные алгоритмы и структуры, используемые при работе с видео. В целях оптимизации работы системы изучен фреймворк FFMPEG позволивший достичь высокой производительности разработанной системы. В качестве интеллектуальной системы, ведущей распознавание людей выбран каскадный классификатор признаков на основе локальных бинарных шаблонов.

На основе полученных знаний разработана программа, параллельно ведущая обработку данных с нескольких камер и детектирующая людей на

видео. Данная программа протестирована на трех IP-видеокамерах IDIS DC-V1803, ведущих трансляцию в формате 4K Ultra HD.

Список литературы

- 1 OpenCV Library. URL: <http://opencv.org/> (дата обращения 01.05.2018)
- 2 Обзор дескрипторов изображения Local Binary Patterns (LBP) и их вариаций. URL : <https://habr.com/post/280888/> (дата обращения 13.05.2018)
- 3 Обучение каскадного классификатора. URL : <http://recog.ru/blog/opencv/85.html> (дата обращения 07.05.2018)
- 4 Руссел Р. FFMPEG – VSD, 2012 – 249с.
- 5 Cascade Classifier. OpenCV. URL: https://docs.opencv.org/3.0-beta/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html (дата обращения 24.05.2018)
- 6 Келер А. Learning OpenCV 3 / А. Келер, Г. Брэдки; пер. с англ. А. Слинкин. – ДМК Пресс, 2017 – 826 с.