

Е. А. Соколов

ИССЛЕДОВАНИЕ МЕТОДОВ И РАЗРАБОТКА ПО ДЛЯ РАСПОЗНАВАНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ БИБЛИОТЕКИ OPENCV

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В данной работе исследуются современные методы распознавания движущихся объектов и разработано ПО для решения данной проблемы. Для выявления и классификации движущихся объектов используются средства библиотеки OpenCV и нейросети YOLO. Результаты показывают баланс скорости работы и точности распознавания.

Ключевые слова: компьютерное зрение, распознавание движения, OpenCV, фоновое вычитание, YOLO, распознавание объектов, машинное обучение.

Проблема распознавания движущихся объектов в настоящее время становится все более актуальной: необходимость ее решения встречается, например, в логистике, на производстве, в сельском хозяйстве, военном деле. Цели работы — исследование современных методов распознавания движущихся объектов и готовое программное обеспечение, способное эффективно решать данную задачу.

Поскольку задача по распознаванию объектов достаточно сложна, часто применяются готовые библиотеки и модели. Также стоит отметить, что движущихся объектов может быть весьма много, а количество пикселей на обрабатываемых кадрах исчисляется обычно сотнями тысяч или миллионами, поэтому для ускорения обработки предпочтительно использовать методы определения движения, не основанные на машинном обучении, а нейросети применять для дополнительной обработки уже выделенных частей изображений при необходимости. В данной работе использована популярная кроссплатформенная библиотека OpenCV, в которой реализованы алгоритмы компьютерного зрения с открытым исходным кодом. Для выделения движущихся областей рассматриваются методы фонового вычитания, предоставляемые OpenCV, а непосредственно для распознавания объектов – средства нейросети YOLO.

Исследование методов определения движения

Вычитание фона – это метод для создания маски переднего плана путем вычитания текущего кадра из фоновой модели. Фоновое моделирование включает в себя два основных этапа: инициализацию фона и его обновление (модель корректируется с учетом изменений в сцене). В простейшем случае маска определяется как разность между фиксированным фоном и текущим изображением [1]. Данный метод самый быстрый, но требует статичного положения камеры и отсутствия движения

фона. Более современные методы фонового вычитания позволяют динамически обновлять фоновое изображение, что позволяет обрабатывать данные и без статичного фона. Рассмотрим некоторые методы, соответствующие этому требованию: MOG и MOG2.

Идея алгоритма MOG заключается в том, что каждый пиксель фона представляется смесью нескольких гауссовских распределений, отражающих вероятные цвета пикселя фона. Важный аспект алгоритма MOG – это определение временных пропорций (весов) для каждой компоненты смеси, обновляющиеся при каждом кадре. Эти пропорции указывают, как долго цвета пикселя остаются на изображении. Таким образом, вероятные цвета, которые остаются на изображении дольше и более статичны, имеют больший вес в смеси гауссовских распределений.

Алгоритм MOG2 похож на MOG, но динамически выбирает количество гауссовских распределений для каждого пикселя, используя модель адаптивной смеси гауссовских распределений. MOG2 также добавляет третий компонент в матрицу маски, который обозначает пиксель тени объекта, что позволяет лучше адаптироваться к изменениям в сцене и более точно отделять объекты от их теней [2].

Исследование методов распознавания объектов, основанных на машинном обучении

Ранее для распознавания объектов на изображениях широко использовались каскады Хаара, но в настоящее время они считаются устаревшими, поскольку появились новые нейронные сети, которые работают быстрее и выдают лучшие результаты распознавания. Одной из таких систем является YOLO — архитектура нейронных сетей, предназначенная для обнаружения объектов на изображении и видеопотоке, использующая следующий принцип: исходное изображение рассматривается как квадратная матрица размерности N , в каждой клетке которой записана информация о наличии объекта, его типе (классе), наличии ограничивающих рамок на соответствующей части картинки. Таким образом, нейросеть просматривает картинку только один раз, что существенно увеличивает скорость обработки. Также YOLO использует функцию потерь, учитывающую точность рамок и уверенность в классах объектов [3].

Разработка ПО для распознавания движущихся объектов

При проведении исследования также разработано ПО для распознавания объектов на видео с веб-камеры или из локального файла с выделением движущихся объектов на видеопотоке и их определением по требованию пользователя, если класс объекта известен системе.

Приложение состоит из серверной и клиентской частей: первая, разделенная на две отдельные программы, отвечает за чтение и обработку видео, а вторая позволяет отображать готовый видеопоток и передавать команды пользователя обратно на сервер.

Высокоуровневая структура приложения представлена на рисунке 1.

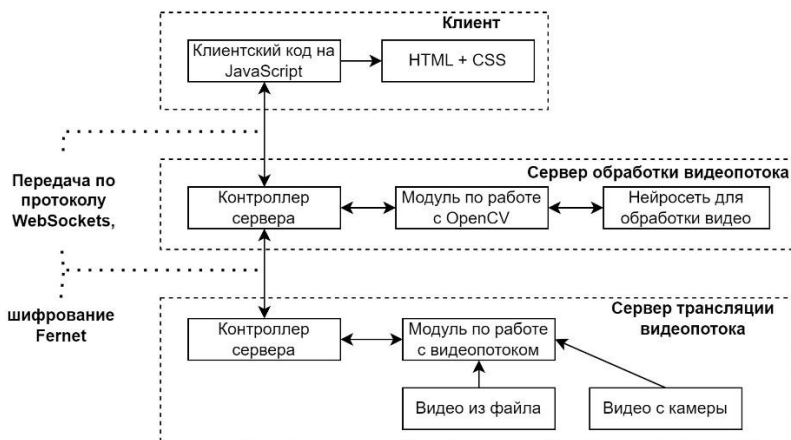


Рисунок 1 – Высокоуровневая структура приложения

Фрагмент интерфейса приложения представлен на рисунке 2.



Рисунок 2 – Фрагмент интерфейса разработанного приложения

Таким образом, исследована часть методов для распознавания движущихся объектов и написано ПО, решающее данную задачу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Интернет-ресурс [habr.com](https://habr.com/ru/articles/786436/) – URL: <https://habr.com/ru/articles/786436/> (дата обращения: 27.04.2024)
2. Интернет-ресурс [opencv.org](https://docs.opencv.org/3.2.0/db/d5c/tutorial_py_bg_subtraction.html) – URL: https://docs.opencv.org/3.2.0/db/d5c/tutorial_py_bg_subtraction.html (дата обращения: 29.04.2024)
3. Интернет-ресурс [leonardoaraujosantos.gitbook.io](https://leonardoaraujosantos.gitbook.io/artificialintelligence/machine_learning/deep_learning/single-shot-detectors/yolo/) – URL: https://leonardoaraujosantos.gitbook.io/artificialintelligence/machine_learning/deep_learning/single-shot-detectors/yolo/ (дата обращения: 14.05.2024)