



(IBIMA). Vision 2020: Sustainable Economic development, Innovation Management, and Global Growth, Madrid. 2017, 827-837.

4. Orlova E.V. Decision-Making Techniques for Credit Resource Management Using Machine Learning and Optimization // Information. 2020. Vol. 11(3). DOI: 10.3390/info11030144

Н.А. Паренский, Р.А. Парингер

## ИССЛЕДОВАНИЕ АЛГОРИТМОВ ДЕТЕКТИРОВАНИЯ ДВИЖЕНИЯ В ВИДЕОПОТОКЕ

(Самарский университет)

**Аннотация.** Данная статья посвящена исследованию различных алгоритмов обнаружения движения в видеопотоке. Сравнение алгоритмов в трех разрешениях на компактных вычислительных модулях Raspberry Pi 3 Model B и Nvidia Jetson nano по количеству кадров в секунду. Выбор наилучшего алгоритма по отношению качества видео к кадрам в секунду.

### Введение

Задача детектирования движущихся объектов является актуальной в современном мире. Её применение можно найти в таких задачах как отслеживание людей в видеопотоке, жестов человека, мимики человека, отслеживание автомобилей в движении, охранных системах, в видеосистемах для госорганов, развлекательно по, робототехнике. Систему видеонаблюдения будет моделировать на компактных вычислительных устройствах RaspberryPi 3 Model B и Nvidia Jetson nano. Задаче является изучить возможности двух данных устройств для построения системы видеонаблюдения с детектирование движущихся объектов с разрешением видео в трех форматах 1920x1080, 1280x720, 640x480 и прийти к выводу какой из заданных разрешений позволит получить плавный видео поток с детектирование движения.

### Методы детектирования движения

**Метод вычитания фона [1].** Один из трех способов детектирования движения. В основе заложено вычитание фонового кадра, который является первым кадром в видеопотоке, из следующих кадров. Тем самым мы разделяем кадр на передний план, где происходит само движение и задний, то, что является первоначальным кадром. Данный метод очень чувствителен к различным изменениям в видеопотоке. Одним из примеров будет являться освещение, которое будет изменяться с течением времени, что приведет к тому, что все изображение будет как бы в движении. Поэтому данный метод подходит для мест с одинаковым постоянным окружением.

**Метод усредненного фона [1].** Данный метод является улучшенной версией метода вычитания фона. Основным действием является вычитание первоначального кадра из последующих, но за исключением, что



первоначальный кадр будет являться усредненным из нескольких кадров в видеопотоке, который будет изменяться с течением времени работы программы. Данный метод не так чувствителен, как предыдущий, из-за усредненного фона, что позволяет использовать данный метод в системах видеонаблюдения в местах с изменяемым освещением при медленном изменении фона.

**Метод низкочастотного фильтра рекурсивного сглаживания [1].** Метод низкочастотного фильтра рекурсивного сглаживания, который, как и метод усредненного фона берет последовательность кадров как фон. Исключением данного метода является то, что начальные кадры не учитываются, а учитываются лишь несколько предыдущих. Особенностью данного метода является то, что считается наличие коэффициента использования текущего кадра. Чем больше значение коэффициента, тем больше влияние текущего кадра. Данный метод очень зависим от предыдущих кадров, что может сказаться на некоторых шумах от самого движущегося объекта, но данный метод хорош при быстро-сменяющемся фоне.

### Постановка эксперимента

Для эксперимента подготовили два устройства Raspberry Pi 3 Model B и Nvidia Jetson nano, с заранее записанным видеопотоком и запрограммированные на python три вышесказанных метода.

Таблица 1

	<b>Raspberry Pi 3 Model B</b>	<b>Nvidia Jetson nano</b>
Производительность	1	1,67
Процессор	64-битный 4-ядерный ARM Cortex-A53 с тактовой частотой 1,2 ГГц	64-битный 4-ядерный ARM Cortex-A57 MPCore с тактовой частотой до 1,42 ГГц
Графический процессор	Broadcom VideoCore IV	NVIDIA Maxwell с 128 ядрами NVIDIA CUDA
Оперативная память	1ГБ LPDDR2	4 Гб LPDDR4
Micro Sd карта	32 Гб	16 Гб

После запуска программы, она проходит несколько этапов. Первый, пробный этап, когда запускается видео поток на устройствах в различном разрешении без обработки, чтобы посмотреть количество кадров, выдаваемых изначально. Следом, этапы, где запускаются методы детектирования изображения. Все этапы проходят с изменением разрешения входного видеопотока. Сначала у видеопотока изменяется разрешение для увеличения производительности, изменяем цветовую палитру на серые тона, потом накладываем Гауссовский блюр, который размывает изображения, чтобы убрать мелкие детали способные повлиять на стабильную работу методов, вычитаем слои, чтобы убрать фон с изображения и оставить лишь объекты переднего



плана. Выделяем их белым цветом. После в несколько итераций увеличиваем объем белых объектов на изображении для отрисовки границ, которые будут показывать движущийся объект.

### Результаты

Таблица 2. Таблица производительности методов детектирования движения (к/с)

	1920x1080		1280x720		640x480	
	Jetson	Raspberry	Jetson	Raspberry	Jetson	Raspberry
Пробный запуск	64.9	16.8	28.5	9.7	50.4	14.4
Метод вычитания фона	6.4	1.4	9.9	2.5	26.7	7.2
Метод усредненного фона	2.	0.5	4.1	1.1	13.8	3.7
Метод низкочастотного фильтра рекурсивного сглаживания	1.3	0.3	2.6	0.7	9.2	2.6

### Вывод

В целях эксперимента было исследовано влияния разрешения на производительность различных методов детектирования движения. Изначально было известна большая производительность Nvidia Jetson nan, чем у Raspberry Pi 3 Model B, поэтому данный эксперимент лишь показывает возможности данных устройств при использовании данных методов детектирования движения. Также было замечено, что нужно изменять разрешения входного видеопотока с использованием целых делителей, хотя бы по одному параметру разрешения видео, что позволит сократить или же совсем убрать использование интерполяцию. Данный параметр существенно увеличит производительность на любом устройстве. Для увеличения производительности также можно разделять каждый кадр видеопотока на несколько частей, что позволит распределить вычисления между ядра процессора. Следующим шагом в данной статье будет являться проверка точности детектирования движения в зависимости от разрешения входного видеопотока. Одним из способов определения объекта на видео, с помощью информации о цвете [2].

### Литература

1. Белясников С.А., Дорофеев Р.С. Методы обнаружения движущихся объектов в видеопотоке / Молодежный вестник ИрГТУ. – 2016. С. 1-5
2. Артемов А. А., Кавалеров М. В., Кузнецов Г. С. Проблема поиска объектов на изображениях с помощью компьютерного зрения на основе



информации о цвете / Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2011. – С. 70-79

3. Kulchandani J. S., Dangarwala K. J. Moving object detection: Review of recent research trends / 2015 International Conference on Pervasive Computing (ICPC). – 2015. – pp. 1-5

4. Mitrokhin, A. Event-based moving object detection and tracking / 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – 2018. – pp. 1-9

5. Shijila B., Tom A. J., George S. N. Simultaneous denoising and moving object detection using low rank approximation / Future Generation Computer Systems. – 2019. – pp. 198-210

Е.Г. Плешаков, Л.С. Зеленко, Д.С.Оплачко

## РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «НЕЙРОСЕТЕВОЙ КЛАССИФИКАТОР ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ»

(Самарский университет)

### Введение

Текстовое представление информации является наиболее часто используемым, особенно в сети интернет. Книги, статьи, новости, твиты, посты, – всё это различные формы текста. На множестве серверов хранятся терабайты, а порой и петабайты информации в текстовом виде. И с каждым днём количество текстовой информации продолжает расти. В таком большом объёме информации, которая чаще всего не структурирована, очень трудно найти текст по конкретной теме, что серьёзно обесценивает эту информацию.

Наиболее перспективным подходом при поиске текстовой информации на данный момент является машинное обучение, когда система на основе небольшой выборки размеченных данных сама создаёт правила для классификации и впоследствии на их основе присваивает категории новым текстам. Наиболее часто в качестве основы для систем классификации используется нейронная сеть [1].

### Структура нейросетевого классификатора текстов на естественном языке

Система классификации состоит из двух основных частей: частотный анализатор со словарем и нейросетевой классификатор, схема представлена на рисунке 1. На вход системы поступает текст, на выходе получаем тему (номер класса), которой посвящен этот текст.

Для решения задачи приведения слов к основной форме существует несколько способов: лематизация (все слова в тексте приводятся к нормальной форме (единственное число, именительного падежа)) и стеминг (выделение