

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

На правах рукописи

ЛАПТЕВ НИКИТА ВИТАЛЬЕВИЧ

**ОБНАРУЖЕНИЕ ОБЪЕКТОВ ВИДЕОРЕЯДА
НА ОСНОВЕ ТЕХНОЛОГИИ
МАШИННОГО ОБУЧЕНИЯ
(НА ПРИМЕРЕ ЛЕСНЫХ ПОЖАРОВ)**

Специальность: 2.3.1 – Системный анализ, управление и обработка информации,
статистика

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель – доктор технических наук,
профессор Гергет Ольга Михайловна

Томск – 2023

Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1 АНАЛИЗ АЛГОРИТМОВ И МЕТОДОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ.....	11
1.1. Актуальность исследования, обзор состояния проблемы	11
1.2. Обнаружение пожара в лесу с помощью цветовой обработки	15
1.3. Обнаружение пожара в лесу с помощью цветовой обработки и выделения динамических признаков	17
1.3.1 Временная разница	18
1.3.2 Вычитание фона	18
1.3.3 Оптический поток	19
1.4. Классификация объектов	20
1.4.1 Классификация на основе обучения	21
1.4.2 Статистическая классификация.....	21
1.5. Обнаружение пожара в лесу с использованием методов глубокого обучения	22
1.6. Опорные области для обнаружения объектов	33
1.7. Локализация ограничивающей рамки, объединение предсказаний	36
1.8. Рекуррентные нейронные сети	39
1.9. Используемые метрики оценки качества детектирования объектов...	41
1.10. Выводы по первой главе	44
ГЛАВА 2 ТЕХНОЛОГИЯ ОБНАРУЖЕНИЯ ОБЪЕКТОВ ВИДЕОПотока НА ОСНОВЕ ТЕХНОЛОГИИ МАШИННОГО ОБУЧЕНИЯ (НА ПРИМЕРЕ ЛЕСНЫХ ПОЖАРОВ)	46
2.1. Разработка алгоритма обнаружения дымового облака	46
2.2. Разработка алгоритма выделения динамических признаков объекта .	49
2.3. Разработка алгоритма локализации объектов на изображении	53
2.3.1 Трекинг объектов детектором EfficientDet	53
2.3.2 Алгоритм кластеризации прогнозов	58
2.4. Разработка алгоритма фильтрации обнаруженных объектов	64
2.5. Состав комплекса программных средств	72

2.6. Выводы по второй главе.....	75
ГЛАВА 3 ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ РАЗРАБОТАННЫХ АЛГОРИТМОВ	77
3.1. Подготовка эксперимента	77
3.2. Тестирование программной реализации алгоритмов.....	79
3.3. Выбор модели НС	87
3.4. Результаты тестирования алгоритмов объединения предсказаний.....	91
3.5. Результаты тестирования классификатора.....	96
3.6. Результаты тестирования технологии раннего обнаружения пожароопасных объектов в лесном массиве	104
3.7. Оценка работоспособности предложенных алгоритмов при решении различных задач	107
3.7.1 Выделение динамических признаков для отслеживания перемещений дистального конца катетера.....	107
3.7.2 Обнаружение объектов в решение задачи навигации роботов..	111
3.7.3 Фильтрация обнаруженных объектов с помощью гибридной архитектуры НС в задаче оценки парковочного пространства.....	113
3.8. Выводы по третьей главе	119
ЗАКЛЮЧЕНИЕ	120
СПИСОК СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ	122
СПИСОК ЛИТЕРАТУРЫ.....	124
ПРИЛОЖЕНИЕ А	137
ПРИЛОЖЕНИЕ Б.....	141
ПРИЛОЖЕНИЕ Г.....	146

ВВЕДЕНИЕ

Актуальность темы исследования. Обнаружение объектов в видеоряде является одной из ключевых проблем исследований в области искусственного интеллекта, которые включены в направление Н1 Стратегии научно-технического развития Российской Федерации. При разработке эффективных методов и алгоритмов обнаружения объектов на основе таких признаков как форма, цвет и текстура требуется учитывать ряд мешающих факторов: размытие объектов при движении, вариативность признаков, изменение условий освещения, перекрытие объектов в процессе перемещения и другие. Игнорирование этих факторов может привести к снижению точности обнаружения и увеличению вероятности ложного срабатывания систем распознавания.

Перспективными инструментами для решения задач анализа статических и динамических признаков на кадрах видеозаписей являются технологии машинного обучения, в частности нейронные сети (НС). Для их применения необходимо осуществлять сбор, разметку и предварительную подготовку исходных данных; выбор модели и подбор параметров архитектуры НС; тонкую настройку гиперпараметров НС, т.е., скорости обучения и объема данных (размера батча) с целью снижения времени сходимости в процессе обучения; интерпретацию результатов анализа данных НС, в том числе анализ хронологии изменения объекта в видеоряде.

Задачи автоматического обнаружения динамических объектов в видеопотоке являются наиболее актуальными в таких приложениях, как автоматическое распознавание лиц [1], медицинская диагностика [2], робототехника [3], транспортные системы [4] и др.

Одним из приложений, имеющих социально важное значение, является разработка систем раннего обнаружения лесных пожаров [5]. Ежегодно в Российской Федерации фиксируется от 10000 до 35000 лесных пожаров, а их общая площадь превышает 1 млн га. Согласно заявлению Федерального агентства лесного хозяйства в 2022 году общая площадь лесных пожаров в России составила 8,96 млн

га, а за последние 3 года в среднем выгорело около 17,8 млн га леса. При этом экономический ущерб составил в среднем 9,7 млрд рублей.

На точность обнаружения пожаров влияет множество факторов, включая освещенность, погодные условия, прозрачность окружающей среды, схожесть характерных признаков с облачностью и туманностью. Дополнительные сложности при обнаружении пожаров создает непостоянство таких признаков, как форма, цвет и текстура. При этом пожар является одним из наиболее сложных объектов обнаружения. Поэтому существует необходимость разработки специальных методов и алгоритмов, учитывающих мешающие факторы, которые влияют на точность и надежность обнаружения объектов в видеоряде.

Целью диссертационной работы является разработка технологии и алгоритмов обнаружения объектов, обладающих динамическими признаками на изображениях видеоряда, обеспечивающих высокое качество распознавания.

В связи с поставленной целью должны быть решены следующие **задачи**:

- провести анализ существующих методов и алгоритмов обнаружения объектов видеоряда с динамическими признаками;
- разработать технологию автоматического обнаружения объектов с динамическими признаками, состоящими из предложенных и реализованных алгоритмов;
- разработать программное обеспечение и провести вычислительный эксперимент с целью оценки качества результатов обнаружения объектов, полученных на основе предлагаемых алгоритмов.

Методы исследования. Используются методы цифровой обработки изображений, технология компьютерного зрения, методы преобразования и анализа изображений, машинного обучения и прикладного программирования. Экспериментальные численные исследования проводились с использованием методов математического анализа, моделирования и математической статистики.

Научная новизна

- Предложена технология анализа изображений в видеопоследовательности, позволяющая достичь высокой точности обнаружения объектов за счет локализации и классификации каждого объекта с учетом как статических, так и динамических признаков.
- Предложен и обоснован алгоритм объединения предсказаний нейросети, позволяющий объединять плотные группы объектов, выделенных на основе статических признаков в один объект на основе оценки отношения площадей пересечения соседних объектов, используемые для формирования кластеров, локализирующих распознаваемые объекты.
- Разработан алгоритм классификации обнаруженных объектов на основе, состоящей из сверточной и рекуррентной нейросетей гибридной архитектуры, обеспечивающей неявное представление хронологии изменения объектов при повышении точности классификации изображений в видеопотоке.

Достоверность полученных результатов диссертационной работы подтверждается результатами тестирования, апробации и внедрения разработанных алгоритмов и технологии на собственных и общедоступных базах изображений.

Практическая ценность работы. Разработанные в диссертационной работе алгоритмические и программные средства могут быть использованы для решения задач обнаружения объектов видеоряда в различных прикладных областях: медицинская диагностика, контроль перемещений роботов, мониторинг местности и т.п.

Положения, выносимые на защиту:

1. Технология анализа изображений в видеопоследовательности на основе предложенных алгоритмов локализации и классификации с учетом статических и динамических данных позволяет достичь высокой точности классификации объектов по метрике F-score не менее 95 %.

2. Алгоритм объединения предсказаний нейросети, позволяющий объединять плотные группы объектов, выделенные на основе статических признаков в один объект с целью повышения точности локализации распознаваемых объектов. Реализация предложенного алгоритма позволяет повысить точность локализации на 6 %.
3. Алгоритм классификации обнаруженных объектов на основе предложенной гибридной архитектуры нейросети позволяет увеличить качество классификации по метрике F-score на 9 %.

Реализация и внедрение результатов работы. Результаты диссертационного исследования использованы при выполнении следующих научно-исследовательских проектов: государственное задание "Наука" №FFSWW-2020-0014 "Разработка научных основ технологии роботизированной мультипараметрической томографии на основе методов обработки больших данных и машинного обучения для исследования перспективных композиционных материалов", 2020-2022 гг.; грант РФФИ № 20-37-90055 "Методы и интеллектуальные технологии планирования движения беспилотных транспортных средств", 2020-2022 гг.; грант УМНИК №16068ГУ/2020 "Система детектирования и локализации очага возгорания в видеоряде на основе средств компьютерного зрения", 2020-2023 гг.

Результаты работы также используются: в ООО "НТП КИБЕРЦЕНТР" (г. Томск) – в программном обеспечении информационно-телекоммуникационной системы интеллектуального видеонаблюдения и аналитики лесопожарной обстановки (ИТС «ИВА»), в ООО "Спотпаркинг" (г. Томск) – в составе системы для обнаружения автомобилей на парковочном пространстве и в учебном процессе в отделении информационных технологий Инженерной школы информационных технологий и робототехники ТПУ. Акты о внедрении приложены к диссертационной работе.

Апробация. Основные результаты данной работы освещались на следующих конференциях и симпозиумах: Международная научно-практическая конференция

студентов, аспирантов и молодых ученых, г. Томск, 2019 г.; 14-й Международный форум по стратегическим технологиям "IFOST 2019", г. Томск, 2019 г.; XXXIV Международная научная конференция "Математические методы в технике и технологиях", г. Санкт-Петербург, 2021 г.; 4-й Международный семинар по фотограмметрии и методам компьютерного зрения для видеонаблюдения, биометрии и биомедицины, г. Москва, 2021 г.; 31-я Международная конференция по компьютерной графике и зрению "ГрафиКон 2021", г. Нижний Новгород, 2021 г.; International Symposium on Electrical, Electronics and Information Engineering, Online, 2021 г.; 6-я Международная конференция по тестированию программного обеспечения, машинному обучению и комплексному анализу процессов, ТМРА 2021, г. Томск, 2021 г.; SIBCON-2022 международная IEEE-сибирская конференция по управлению и связи, г. Томск, 2022 г.

Личный вклад соискателя. Основные научные результаты, выносимые на защиту, получены автором лично. Данная работа выполнена в сотрудничестве с учёными и исследователями, оказавшими содействие при проведении настоящего исследования, а именно: постановка задач исследования по теме диссертации выполнена совместно с научным руководителем Гергет О.М. (Томский политехнический университет); обзор существующих алгоритмов анализа, обработки и визуализации данных выполнен совместно с Даниловым В.В. (Томский политехнический университет) и научным руководителем Гергет О.М. (Томский политехнический университет); сбор, разметка и валидация данных пожаров выполнены совместно с сотрудниками компании «ИНКОМ»; разработка алгоритмов выполнена совместно с Лаптевым В.В. («Спутник») и Колпащиковым Д.Ю. (НОЛ ОАБД ИШИТР, Томский политехнический университет).

Публикации. Основные результаты диссертационного исследования отражены в 13 работах: 5 публикаций в рецензируемых журналах, рекомендованных ВАК РФ, из них – 2 публикации проиндексированы в базах научного цитирования Scopus и Web of Science; 1 патент на изобретение, 3

свидетельства о государственной регистрации программ для ЭВМ; 4 статьи в сборниках трудов международных и российских конференций.

Структура и объем диссертации. Диссертационная работа состоит из введения, трех глав, заключения, списка литературы из 133 наименований. Работа содержит 163 страниц основного текста, включая 54 рисунков и 10 таблиц.

Во введении обоснована актуальность темы исследования, сформулированы цель и задачи работы, определена научная новизна, а также отражены практическая значимость и положения, выносимые на защиту.

В первой главе введены необходимые термины и обозначения; проведен обзор методов обнаружения объектов на кадрах видеоряда, описаны известные современные нейросетевые методы обнаружения объектов. Подробно рассмотрены модели рекуррентных нейронных сетей, позволяющих решить задачу выделения временной составляющей движения дымового облака. Проведен анализ алгоритмов объединения прогнозов нейросетевого анализа данных, назначение которых выделить исследуемый объект с нескольких кадров. Обоснована необходимость разработки комплексной технологии раннего обнаружения пожаров.

Во второй главе представлены: технология анализа изображений в видеопоследовательности, на основе предложенных алгоритмов локализации и классификации с учетом статических и динамических данных; алгоритм, позволяющий объединять разрозненные предсказания нейросети в итоговое решение; новая гибридная архитектура нейронной сети, основанная на интеграции данных сверточной нейронной сети в рекуррентную нейронную сеть, отличающаяся от существующих неявным представлением временной составляющей, агрегированной аугментацией данных и динамическим подбором размера батча в зависимости от скорости сходимости модели на прошлых итерациях, что позволяет осуществлять классификацию изображений в видеопотоке с высокой точностью.

В третьей главе представлены вычислительные эксперименты с целью оценки результатов обнаружения объектов, полученных на основе предлагаемых алгоритмов. Сравнение полученных результатов с аналогичными системами обнаружения, чьи характеристики представлены в открытом доступе, показало, что предложенная технология выделения объектов с динамическими признаками демонстрирует более высокую точность обнаружения и меньшее количество ложных срабатываний для задачи обнаружения лесных пожаров. Приведены примеры применения предложенных алгоритмов в других областях деятельности, что дает возможность говорить о свободной встраиваемости алгоритмов в системы в разных областях деятельности и их универсальности.

ГЛАВА 1

АНАЛИЗ АЛГОРИТМОВ И МЕТОДОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ

В этой главе введены необходимые термины и обозначения; проведен обзор методов обнаружения объектов на кадрах видеоряда, описаны известные современные нейросетевые методы обнаружения объектов. Подробно рассмотрены модели рекуррентных нейронных сетей, позволяющих решить задачу выделения временной составляющей движения дымового облака. Проведен анализ методов объединения прогнозов нейросетевого анализа данных, назначение которых выделить исследуемый объект с нескольких кадров. Обоснована необходимость разработки комплексной технологии раннего обнаружения пожаров.

1.1. Актуальность исследования, обзор состояния проблемы

В связи с высокой опасностью техногенных катастроф следует отметить актуальность разработки систем раннего предупреждения лесных пожаров путем наземного видеомониторинга. В последние несколько лет изменение климата и антропогенные факторы оказывают значительное влияние на окружающую среду. Некоторые из этих явлений, такие как экстремально высокие температуры и засухи, существенно увеличивают вероятность возникновения пожаров. Лесные пожары оказывают катастрофические последствия для локальных и глобальных экосистем, наносят серьезный ущерб инфраструктуре, приводят к человеческим травмам и жертвам. Чаще всего лесные пожары являются результатом вмешательства человека в лесную экосистему. В связи с этим весьма актуальным остается разработка систем раннего обнаружения пожаров.

Ежегодно в Российской Федерации фиксируется от 10 тысяч до 35 тысяч лесных пожаров, а общая площадь горения ежегодно составляет от 1 миллиона гектаров леса [6]. За последние 10 лет наиболее сильные лесные пожары фиксировались в 2012 году, когда площадь пожаров распространилась на территории в 16 миллионов гектаров. В 2019 и в 2020 годах площадь лесных пожаров составила 16,5 миллионов гектаров. Согласно заявлению Федерального

агентства лесного хозяйства, в 2021 году общая площадь лесных пожаров в России составила 17 миллионов гектаров.

Лесные пожары представляют большую угрозу не только Российской Федерации, но и являются общемировой проблемой. За последние 10 лет в мире произошло несколько крупнейших лесных пожаров, среди которых стоит выделить пожар в Австралии, случившийся в 2020 году и являющийся самым масштабным в истории страны. В результате пожаров было уничтожено около 18,6 миллионов гектаров леса и прочих территорий, что примерно составляет 186 квадратных километров, разрушено 5900 единиц различных сооружений. В результате стихийного бедствия погибли не менее 34 человек. В 2014 году по оценкам правительства Канады было сожжено 3,5 миллионов гектаров (8 600 000 акров) леса, а тушение пожара обошлось в 55 миллионов канадских долларов. В 2020 году начались сильнейшие пожары в американском штате Калифорния. К концу года сообщалось, что площадь территорий, уничтоженных огнем, превысила 1,5 миллиона гектаров. Также в результате засухи в период с 2019 по 2021 годы были отмечены крупные пожары в Европе, а именно на территориях Турции и Греции.

Для мониторинга местности выделяют три типа систем, а именно: наземные, воздушные и космические системы. При этом изучаются различные модели, направленные на обнаружение возгорания с высокой точностью в разных погодных условиях.

Наземный контроль лесных зон осуществляется при помощи приборов дистанционного зондирования местности или визуального контроля. Системы раннего обнаружения, основанные на приборах дистанционного зондирования местности, могут состоять либо из отдельных датчиков (фиксированные, поворотные или камеры 360°), либо из сетей наземных датчиков. Как правило, такие датчики устанавливают на достаточном удалении друг от друга, чтобы обеспечить максимальный охват контролируемой местности. Таким образом, они обычно располагаются на сторожевых вышках или сооружениях, расположенных на возвышенных позициях. Для такого типа систем в качестве датчиков

используются оптические камеры, которые предоставляют информацию о цвете детектируемого объекта [7]. Лазерные и ИК-системы предоставляют данные в диапазоне от низкого разрешения до сверхвысокого разрешения для различных сценариев обнаружения пожара [8]. Тепловизоры улавливают тепловой поток, который излучается пламенем и дымовым облаком от пожара [9,10]. К преимуществам такого рода систем обнаружения можно отнести высокую чувствительность, что в свою очередь обеспечивает низкую вероятность ложного срабатывания системы. Однако лазерные и ИК-системы так же, как и камеры видимого излучения подвержены влиянию погодных условий. Такие природные факторы как туман и дымка способны значительно снизить дальность их действия. По сравнению с системами, в основе которых лежит использование камер видимого диапазона, они имеют значительно более высокую стоимость. Дальность обнаружения устройства, работающего в инфракрасном спектральном диапазоне при условии идеальной видимости, может достигать 15 километров, в то время как в неблагоприятных условиях параметр снижается до величины менее 1-го километра [11]. Дальность действия системы обнаружения на основе видеокамер видимого диапазона при отличной видимости местности может достигать 15-25 километров при установке камер на высоте свыше 50 метров в зависимости от высоты леса [12]. Применение видеокамер видимого и ближнего инфракрасного частотного диапазона представляет собой экономически приемлемое и достаточно эффективное решение задачи автоматического обнаружения лесных пожаров.

Помимо наземного контроля широко применяется воздушный метод контроля лесных массивов. Службы противопожарной безопасности активно используют авиационные средства для совершения облетов лесных территорий. Также наблюдение за лесами и обнаружение лесных пожаров в последнее время осуществляют с помощью беспилотных летательных аппаратов (далее БПЛА) [13–16]. БПЛА представляют собой подходящий вариант для мониторинга, предотвращения и предупреждения возможных пожаров в лесах. Поскольку дроны обеспечивают гибкость, универсальность, маневренность, передачу в режиме

реального времени, низкую задержку, низкие затраты, это позволяет предоставлять эффективные решения по предотвращению, прогнозированию и немедленному реагированию в случае возникновения угрозы. К недостаткам данных систем можно отнести низкую автономность и с этим связанный небольшой охват исследуемой территории.

Также известны примеры применения дистанционного зондирования. Для этих целей используются спутниковые системы [17–19]. Спутниковое дистанционное зондирование имеет некоторые преимущества, такие как: широкий охват наблюдения, передача в режиме реального времени, локализация возгорания. Основным недостатком спутниковых систем обнаружения лесных пожаров являются: задержка в получении информации, высокие затраты, низкий уровень эффективности и практичности. Системы определения местоположения GPS демонстрируют проблемы с точностью в отношении слепых точек на земной поверхности.

Использование оптических датчиков является обязательным требованием для любого из методов наблюдения, чтобы гарантировать эффективность и автоматизацию обнаружения и выдачи предупреждений, поскольку полученная информация может обрабатываться с помощью различных алгоритмов, применимых к нескольким методологиям обнаружения пожара. Методы можно разделить на две группы: обнаружение с помощью алгоритмов машинного обучения и обнаружение с помощью цифровой обработки изображений на основе обработки цвета [20,21]. Для локализации контуров или фронтов огня решающее значение имеют следующие факторы: а) расположение мест, где интенсивность огня выше; б) направление распространения огня; в) разносторонний характер распространения огня и т. д.

О высоком интересе к проблеме обнаружения лесных пожаров свидетельствует большое количество публикаций по данной тематике за последние пять лет (2019 – 2023гг.) [16,22,31–37,23–30]. Наибольшее количество публикаций посвящено разработке методов и алгоритмов обработки изображений. Это

обусловлено возможностью повышения эффективности современных систем мониторинга за счет использования камер видимого диапазона совместно с беспроводными многосенсорными измерительными комплексами (тепловыми датчиками, датчиками атмосферного давления, относительной влажности, кислорода, углекислого и угарного газа и др.) [38–40]. Обнаружение цвета, яркостных и текстурных характеристик дымового облака [37,41–43] были одни из первых методов, используемых в системах мониторинга, а также анализ динамики характеристик движущихся объектов (скорости, направления, площади) [44–47]. Значительный вклад в развитие методов обнаружения лесных пожаров внесли ученые Govil K., Welch M. L., Ball, J. T. (США), B. U. Toreyin (США) [12], С. Yuan (Канада) [48], N. Alamgir (Австралия)[49], Васюков В.Н., Зайцева А.Ю., Слезин К.А.(Россия) и др. [50–57]

При автоматическом обнаружении лесных пожаров достаточно часто алгоритмы как машинного обучения, так и цифровой обработки изображений на основе цвета сталкиваются с общими проблемами: например, генерация ложных срабатываний из-за изменения освещения, низкой скорости передачи сигналов, невысокого качества разрешения, непостоянства контраста и т.п. Все это негативно сказывается на обработке изображений в видеоряде, являющихся базовыми элементами соответствующего процесса. В связи с чем необходимо провести исследование с целью определения наиболее эффективных методов обнаружения и локализации очага возгорания и разработки комплексной технологии раннего обнаружения пожароопасных объектов в лесных массивах.

1.2. Обнаружение пожара в лесу с помощью цветовой обработки

В качестве доминирующего признака пожара цвет является самым ранним и наиболее часто используемым признаком в разработке методов обнаружения на основе технического зрения. Информация о цвете обычно используется в качестве этапа предварительной обработки при обнаружении потенциального возгорания. Разработано множество алгоритмов, использующих дискриминационные свойства

цветового пространства: RGB [58–60] YCbCr [61], CIELAB [44], YUV [62] и HSV [63], для получения областей огня на изображении.

Цветовое пространство (цветовая модель), которое является спецификацией системы координат, направлено на установление спецификации цветов стандартным и общепринятым способом [64]. В области цифровой обработки изображений преобладающими моделями обработки изображений являются цветовая модель RGB; цветовая модель оттенка, насыщенности и интенсивности (HSI); цветовая модель оттенка, насыщенности и значения (HSV) и цветовая модель Lab. Цветовая модель HSI (от англ. Hue, Saturation, Intensity) представляет цвета в виде трех параметров: оттенок (hue), насыщенность (saturation) и интенсивность (intensity). Оттенок отражает цветовой тон, например, красный, зеленый или синий. Насыщенность определяет, насколько ярким и насыщенным является цвет. Если насыщенность высокая, то цвет кажется ярким и насыщенным, если низкая, то цвет выглядит блеклым и более серым. Интенсивность определяет яркость цвета, то есть насколько светлым или темным он является. Цветовая модель HSV (от англ. Hue, Saturation, Value) представляет цвет в виде трех параметров: оттенок (hue), насыщенность (saturation) и яркость (value). Оттенок определяет цветовой тон, насыщенность - интенсивность цвета, а яркость - его светлоту. Цветовая модель Lab ($L^*a^*b^*$) — это модель значения светлоты от значения хроматической составляющей цвета (тон, насыщенность). Светлота задана координатами L (изменяется от 0 до 100, то есть от самого темного до самого светлого), хроматическая составляющая задается двумя декартовыми координатами a и b. Координата (a) изменяется в диапазоне от красного к зеленому, а координата (b) изменяется в диапазон от желтого к синему. Цветовые модели получили широкое распространение для представления изображений в соответствующих цветовых пространствах. На рисунке 1 пример отображения огня и дыма в каждом канале разных цветовых моделей.

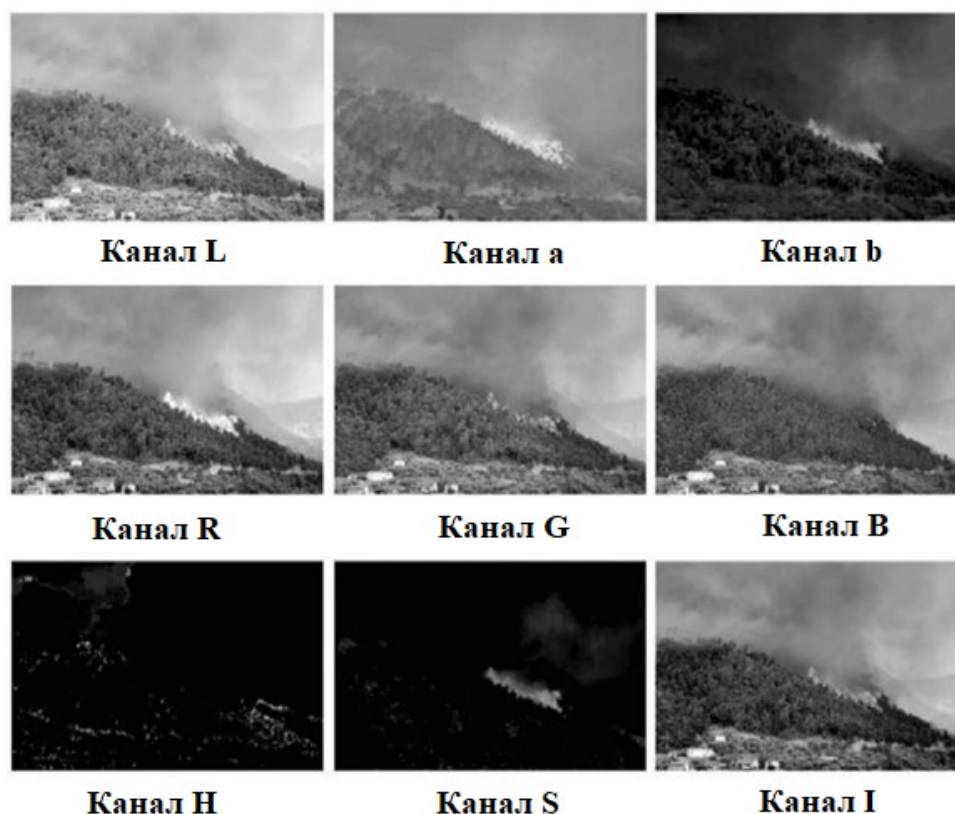


Рисунок 1 – Отображение изображения лесного пожара в разных цветовых каналах

Недостатком моделей обнаружения пожара на основе цвета является высокий уровень ложных тревог, поскольку одноцветной информации в большинстве случаев недостаточно для раннего и надежного обнаружения пожара.

1.3. Обнаружение пожара в лесу с помощью цветовой обработки и выделения динамических признаков

Как правило подходы к обнаружению возгараний, при которых внимание уделяется только объектам, отображающим цвет пламени, считаются ненадежными и имеют тенденцию повышать уровень ложных срабатываний, поэтому для создания более точных и надежных систем обнаружения настоятельно требуются дополнительный анализ характеристик пожара и более эффективные методы. Форма пожаров изменяется во времени, т.к. воздушный поток, создаваемый ветром, в большинстве случаев приводит к резким колебаниям и перемещению огня. Эти динамические особенности делают методы обнаружения

движения широко применяемыми при поиске очага возгорания для изоляции движущихся объектов, посредством исключения из рассмотрения пикселей, для которых не наблюдается изменение интенсивности, т.е. не относящиеся к огню или не соответствующие дымовому облаку.

В настоящее время большинство существующих методов сегментации движения используют либо временную, либо пространственную видеоинформацию и могут быть сведены к четырем основным категориям: временная разность, вычитание фона и оптический поток.

1.3.1 Временная разница

Временная разность — это метод прямого обнаружения движения, использующий временную разницу последовательных кадров в видеопоследовательности. Абсолютная разница каждого пикселя между двумя или тремя последовательными кадрами может быть вычислена, и для получения движущегося объекта используется пороговое значение, такое как алгоритм разности трех кадров, используемый в [30].

Временная разность проста в реализации и хорошо адаптируется к динамическим средам, но она не так эффективна для извлечения всей области или полных форм движущихся объектов. Особенно плохо метод, построенный на временной разности, справляется с сегментацией внутренней части движущегося объекта.

1.3.2 Вычитание фона

Вычитание фона — часто используемый подход для сегментации движения, особенно в ситуации, когда фон относительно статичен. В этом методе стационарные пиксели изображения определяются как фон, поскольку фон можно рассматривать как статичную во времени часть изображения. Если фоновая сцена наблюдается в течение определенного периода времени, то можно рассчитать и смоделировать пиксели, составляющие весь фон. Например, фон можно оценить, усредняя последовательные кадры инициализации, потому что движущиеся

области и объекты занимают только некоторые части фоновой сцены в изображении и их влияние со временем компенсируется усреднением. После того, как модель фона известна, вычисляется разность между пикселями предыдущего и текущего кадров. Если разность между пикселями превышает пороговое значение, то фрагмент изображения считается изменяющимся. Метод вычитания фона в своих работах использовали [66–69]

Хотя методы вычитания фона в большинстве случаев хорошо изолируют соответствующие пиксели движущихся областей. Однако, они становятся уязвимыми, когда сцена более сложная, например, при наличии динамических изменений, таких как движущиеся листья или внезапные изменения освещения. Поскольку предполагается, что камера неподвижна, модель фона не может достаточно хорошо отображать фон, если фон также меняется.

1.3.3 Оптический поток

Оптический поток описывается как двумерное распределение «скоростей» движения яркостных структур в плоскости изображения. Эту функцию можно применять для оценки перемещения локального пикселя изображения и указания скорости изменения каждого пикселя изображения между соседними изображениями. Пиксель изображения соответствует одному вектору скорости, и эти векторы скорости составляют поле оптического потока. Короче говоря, оптический поток способен преобразовывать информацию об изображении в предполагаемые поля движения для более сложного анализа.

Ключевая идея этой методики основана на концепции постоянства яркости. Если движение сравнительно мало и освещение предмета остается равномерным в пространстве и постоянным в течение периода, то предполагается, что яркость отдельной точки остается постоянной во времени движения. Предположение о постоянстве яркости математически представлено следующим уравнением:

$$\frac{d}{dt}I = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = I_x u + I_y v + I_t = 0, \quad (1.1)$$

где $I(x, y, t)$ – функция интенсивности изображения от пространственных координат (x, y) и времени t . Вектор потока $(u, v) = (x_t, y_t)$ направлен в направлении движения пикселя (x, y) .

Таким образом, многие из разработанных методологий объединяют информацию о цвете и движении в изображениях и видео [70]. Zhang et al. [71] используют вероятностную модель на основе цвета для обнаружения очагов пожара и особенностей движения для окончательного решения о возникновении пожара. Авгеринакис и др. [72] предложили идентифицировать блоки-кандидаты дыма, а затем построили гистограммы ориентированных градиентов (HOG) и гистограммы оптического потока (HOF). Таким образом, принимается во внимание как внешний вид, так и информация о движении. Аналогичным образом Mueller et al. [73] использовали две схемы оптического потока. А именно, оптимальные модели массопереноса и модели оптического потока, управляемые данными.

Chen et al. в своей работе [74] предложили объединить обнаружение движения с использованием модели смещения Гаусса, анализ цвета с использованием цветовой фильтрации RGB и временной анализ мерцания. Алгоритм был применен к набору видеоданных, состоящему из различных дневных и ночных условий, однако ночью анализ цвета менее полезен, и ночной дым менее заметен. Таким образом, обнаружение лесных пожаров в ночное время обычно основывается на анализе движения.

1.4. Классификация объектов

Как правило, территории, захваченные камерой, включают в себя различные объекты, такие как животные, растения, транспортные средства и пожары. Для дальнейшего отслеживания целевых объектов и анализа их поведения требуется точно отличать их от других объектов. Таким образом, для решения этой задачи используются методы классификации объектов, которая классифицирует тип обнаруженных объектов на этапе сегментации.

В настоящее время предложено множество подходов к классификации объектов, которые в целом можно разделить на три типа: классификация на основе обучения, классификация на основе статистики и классификация на основе правил.

1.4.1 Классификация на основе обучения

В работе [75], Varnproutis et al. предложили извлекать дополнительную функцию, нацеленную на оценку энергии пространственно-временной согласованности. С целью повышения надежности обнаружения пожаров был использован классификатор опорных векторов (SVM), чтобы. В работе [67] для обнаружения дыма используется комбинация методов. На первом этапе осуществляется извлечение дыма с переднего плана изображения на основе модели смеси Гаусса, которая позволяет учитывать движение дыма. На следующем этапе реализован метод сегментации дыма посредством выделения областей с более высоким цветовым контрастом на основе модели HSV. На завершающем этапе реализован метод классификации SVM, что позволило точно выделить области дыма из сегментированных изображений. В исследовании [76] цветовая модель YCbCr применяется для выделения областей интереса (ROI), апостериорно оцениваются структуры в 2D и 3D в формах статистики и динамики, это позволяет получить такие характеристики, как форма, мерцающая область и движение. С целью отделения части изображения, содержащего пламя от остальной части сцены, используется классификатор на основе Extreme Learning Machine (ELM).

1.4.2 Статистическая классификация

В работе [77] движение обнаруживается с помощью экстрактора визуального фона (ViBe). Для этого дым извлекается из остальной части сцены через контраст, вызванный дымом, с учетом компонентов цвета в модели HIS. Апостериори характеристики текстуры дыма извлекаются через адаптивный масштабный локальный бинарный паттерн (AS-LPB). Эта информация проходит

процесс нормализации с учетом использования фильтра Гаусса и анализа главных компонент (РСА). Анализ и классификация осуществляются с помощью скрытой марковской модели (НММ), при этом полученные результаты показывают снижение ложных срабатываний и стабильность обнаружения.

Рассмотренные выше алгоритмы в основном сосредоточены на обнаружении и анализе цветовой и временной составляющей дыма и пламени в последовательных видеоизображениях. Однако, ряд методов обнаружения фокусируются на характеристиках цвета и формы вместе с временным поведением дыма и пламени. Из-за изменчивости формы, движения, прозрачности, цвета, текстуры дыма и пламени многие из существующих подходов уязвимы для ложных срабатываний.

1.5. Обнаружение пожара в лесу с использованием методов глубокого обучения

Глубокое обучение — это специализированная форма методов машинного обучения, которая имитирует работу нейронов в головном мозге. Сверточная нейронная сеть (CNN) — один из наиболее часто используемых методов машинного обучения при обнаружении и автоматической классификации лесных пожаров. Архитектура CNN построена как серия этапов, таких как свертка, нелинейность, уровни объединения, за которыми следуют дополнительные слои свертки и полносвязные слои. CNN привлекли значительное внимание исследователей компьютерного зрения, поскольку они извлекают признаки непосредственно из входных данных, обладая при этом высокой точностью распознавания. В этом разделе описываются ключевые концепции, включая популярные архитектуры CNN, трансферное обучение и обнаружение дыма и огня на основе CNN.

1.5.1 Технология трансферного обучения, обнаружение дыма и огня на основе CNN

Трансферное обучение — популярный и эффективный подход в методах глубокого обучения. Концепция трансферного обучения заключается в том, чтобы взять уже существующие модели в качестве отправной точки для обучения и передать знания, полученные в результате аналогичной задачи. В случае трансферного обучения проще и быстрее настроить сеть, чем проводить обучение с нуля. В структуре CNN ранние уровни содержат общие функции, которые можно повторно использовать для различных задач. Напротив, последние уровни более специфичны для приложений. Основываясь на этом свойстве, начальные слои хорошо сохраняются, в то время как конечные корректируются для обучения с новым интересующим набором данных.

Высокая точность распознавания CNN требует огромных размеченных наборов данных для обучения его глубокой архитектуре. Однако создавать размеченные данные дорого и утомительно, в том числе, существуют задачи, для которых невозможно получить большое количество обучающих выборок. По этой причине трансферное обучение является альтернативой обучению CNN с ограниченным набором данных.

Технологию трансферного обучения в своей работе применили авторы [78], они объединили функции глубокого обучения и ручной работы для распознавания областей огня и дыма. Для статических функций была адаптирована архитектура AlexNet[79], а для динамических функций использовался алгоритм адаптивного взвешенного направления. Более того, Sharma et al. [80] использовали оптические изображения и перенастроили две предварительно обученные CNN, основанные на магистралях VGG16[81] и ResNet50[82]. Стоит упомянуть, что для обучения был создан несбалансированный набор данных, включающий больше изображений без огня. В своей работе [83] Zhang et al. предложил глубокие CNN для каскадного обнаружения лесных пожаров. Во-первых, чтобы определить содержит изображение огонь или нет, был обучен классификатор пожаров с полным

изображением. Далее для локализации участков возгорания на этом изображении был реализован классификатор с мелкими фрагментами. Классификатор полного изображения реализован посредством глубокой CNN, которая была точно настроена с помощью AlexNet, а мелкозернистый классификатор патчей - это двухуровневая полностью подключенная нейронная сеть, обученная с помощью улучшенных функций слоя Pool-5. Хорошие результаты получены авторами работ [36, 38-40], которые применяли для обнаружения пожаров такие архитектуры нейронных сетей, как GoogleNet[85], YOLO LeNet-5, SqueezeNet.

Следует отметить работу [86], где авторы объединили AlexNet в качестве базовой архитектуры и Интернет мультимедийных вещей (IoMT) для обнаружения пожаров и управления бедствиями. В разработанной системе введен механизм адаптивной приоритизации камер в системе наблюдения, позволяющий активировать камеры с высоким разрешением для подтверждения возгорания и анализа данных в режиме реального времени. Кроме того, Даннингс и Брекон [87] предложили использовать несложные архитектурные варианты CNN и подход суперпиксельной локализации, направленный на снижение вычислительной производительности, предлагая снизить время обработки до 17 кадров в секунду.

Поскольку количество общедоступных наборов данных о лесных пожарах все еще ограничено, Sousa et al. [88] предложили метод обнаружения пожара, основанный на расширении данных и обучении с передачей данных. Метод, который включал обучение на базе данных ImageNet [89] модели Inception-V3, был повторно обучен и оценен с использованием десятикратной перекрестной проверки в базе данных Corsican Fire [90]. Расширяя подходы к глубокому обучению, Barmproutis et al. [91] объединили мощь более быстрой рекуррентной сверточной нейронной сети (R-CNN) с многомерным динамическим анализом текстуры на основе LDS более высокого порядка, нацеленного на раннее обнаружение лесных пожаров. В работе [92] авторы предложили объединить более быстрый R-CNN (с не максимальной аннексией для определения местоположения дымовой цели на основе статической пространственной информации) с 3D CNN (для распознавания

дыма путем комбинирования динамической пространственно-временной информации). Jadon et al. [93] пошли еще дальше и предложили сверточную нейронную сеть FireNet с использованием стандартного набора данных и самостоятельно предложенного набора данных о пожарах, достигнув обнадеживающих характеристик с точки зрения ряда показателей оценки.

1.5.2 Архитектуры обнаружения объектов CNN

В соответствии с концепцией глубокого обнаружения объектов методы можно разделить на два основных класса: двухэтапные методы [94–97] и одноэтапные методы [98–100]. Двухэтапные методы сначала генерируют некоторые наборы объектов-кандидатов, а затем классифицируют эти наборы по определенным категориям. Одноэтапные методы одновременно извлекают и классифицируют все предложения объектов. В результате двухэтапные методы имеют более низкую скорость обнаружения и более высокую точность обнаружения, в то время как одноэтапные имеют более высокую скорость обнаружения и сопоставимую точность обнаружения

1.5.3 Двухэтапные методы глубокого обнаружения объектов на примере модели R-CNN

Двухэтапные методы рассматривают обнаружение объектов как многоступенчатый процесс. При наличии входного изображения сначала извлекаются некоторые предложения возможных объектов. После сверточная нейронная сеть извлекает признаки из каждой области. Заключительным этапом является работа полносвязного слоя SVM.

Предложения по регионам могут быть созданы различными методами. Для модели R-CNN это выборочный поиск [101], представленный на рисунке 2. При осуществлении выборочного поиска, к изображению применяется алгоритм сегментации, и на основе карты сегментации выделяются предложения регионов (ограничивающие рамки).

R-CNN: Выделение областей CNN

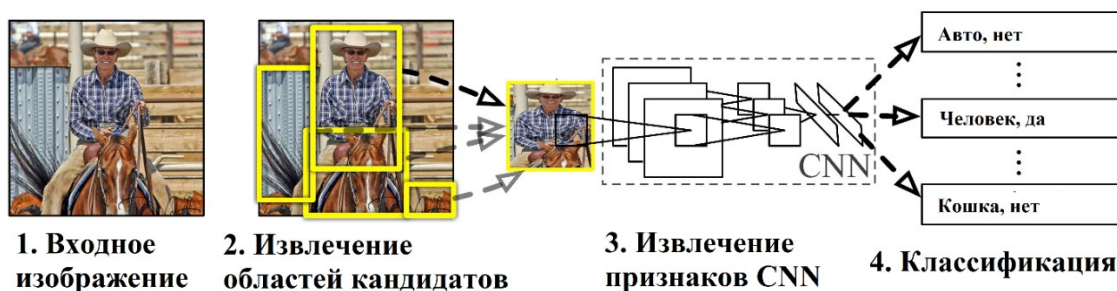


Рисунок 2 – Принцип работы модели нейронной сети R-CNN

Второй и третий этапы - работа CNN на подготовленных на 1-ом этапе изображениях. В качестве CNN может быть использована, например, как в работе [79] нейронная сеть AlexNet, которая сначала предварительно обучается в ImageNet [89], а затем осуществляется настройка на конкретный набор данных обнаружения объектов. Поскольку предложения по регионам имеют разные размеры, для повышения точности определения местоположения предложения используется модель линейной регрессии, которая дополнительно обучается для прогнозирования более точной ограничивающей рамки.

Предполагая, что исходная ограничивающая рамка представлена как (P_x, P_y, P_w, P_h) , где P_x и P_y - координаты центра предложения, P_w и P_h - ширина и высота предложения P . Ограничивающая рамка, соответствующая размеченным данным (G_x, G_y, G_h, G_w) . Тогда для определения смещения ограничивающей рамки t_* для (P, G) может быть записана как:

$$t_x = (G_x - P_x)/P_w, \quad (1.2)$$

$$t_y = (G_y - P_y)/P_h, \quad (1.3)$$

$$t_w = \log(G_w/P_w), \quad (1.4)$$

$$t_h = \log(G_h/P_h), \quad (1.5)$$

Для прогнозирования смещения, где t_x, t_y, t_w, t_h , для нового предположения (т.е. P) используются характеристики слоя Pool-5, представленные как $\phi_5(P)$. Таким образом $\hat{t}_*(P) = w_*^T \phi_5(P)$, где $*(x, y, w, h)$ является линейной функцией w_* – вектор, может быть оптимизированы с помощью регуляризованной цели наименьших квадратов следующим образом:

$$w_* = \arg \min \sum_i^N (\hat{t}_* - \hat{w}_*^T \phi_5(P))^2 + \lambda \|\hat{w}_*\|^2, \quad (1.6)$$

где λ - коэффициент регуляризации, который обычно устанавливается равным 1000. При обучении регрессионной модели для каждого класса используется предложение, которое имеет перекрытие IoU более 0,6 с размеченной ограничивающей рамкой. В противном случае предложение игнорируется. На основе изученного w_* (т. е. w_x, w_y, w_w, w_h) прогнозируемая ограничивающая рамка предложения (P) может быть рассчитана следующим образом:

$$\hat{P}_x = P_w w_x^T \phi_5(P) + P_x, \quad (1.7)$$

$$\hat{P}_y = P_h w_y^T \phi_5(P) + P_y, \quad (1.8)$$

$$\hat{P}_w = P_w \exp(w_w^T \phi_5(P)), \quad (1.9)$$

$$\hat{P}_h = P_h \exp(w_h^T \phi_5(P)). \quad (1.10)$$

Новые прогнозируемые предложения имеют более высокую точность определения местоположения.

При точной настройке сети CNN предложение помечается как положительное для соответствующего класса, если оно имеет максимальное перекрытие по метрике IoU с ограничительной рамкой, и перекрытие составляет не менее 0,5. В противном случае предложение помечается как класс фона. На основе функций CNN, извлеченных из обученной сети CNN, линейные классификаторы SVM для разных классов проходят дальнейшее обучение соответственно. При обучении классификатору SVM в каждом классе только

ограничивающий прямоугольник "истина-основа" помечается как положительный. В противном случае предложение помечается как отрицательное, если его метрика имеет значение ниже 0,3 со всеми ограничивающими рамками "истинность". Поскольку извлеченные функции CNN слишком велики для загрузки в память, метод начальной загрузки используется для извлечения жестких негативов при обучении классификаторов SVM.

Детектор Faster R-CNN. В работе [102] введен нейросетевой модуль Region Proposal Network (RPN), отвечающий за грубое вычисление областей интереса. Он располагается на последнем слое CNN и разделяет сверточные слои с нейросетевым детектором Fast RCNN [96]. Поэтому его вычислительная нагрузка имеет небольшую долю в общей работе метода. Для учета масштаба объектов на изображении введена пирамида опорных областей (anchors), которые вычисляются в процессе классификации. Данный метод эффективно решает задачу классификации для разных масштабов и разных соотношений сторон объектов. Всего используется 9 опорных областей, каждая из которых имеет заданную форму и соотношение сторон (w/h). Несмотря на то, что система состоит из двух модулей, RPN и Fast RCNN детектора, оба имеют общие сверточные слои. Авторы предложили способ совместного обучения RPN и Fast RCNN детектора. На вход подается изображение любого размера, в отличие от YOLO и SSD, которые используют изображения фиксированного размера. Самый последний сверточный слой выполняет свертку с предыдущим слоем скользящим окном $n \times n$ размера, где $n = 3$. Поэтому последний сверточный слой также может быть отнесен к первому модулю RPN. Второй модуль получает на своем входе области кандидаты и значения последнего слоя свертки для этих областей. Детектор реализован на основе полносвязной нейронной сети (FC) и имеет два слоя. Слой reg имеет $4 \times k$ выхода, которые представлены координатами k опорных областей (anchors) и слоем классификации cls , представленным выходом $2 \times k$, который дает возможность оценить вероятность наличия или отсутствия объекта для каждой опорной области. Детектор имеет инвариантность к смещению, поскольку используются

разделяемые признаки на всех уровнях нейронной сети и одинаковый набор опорных областей. Faster RCNN представленный на рисунке 3, использует алгоритм не максимального подавления (NMS) для удаления повторяющихся областей, указывающих на один и тот же объект.

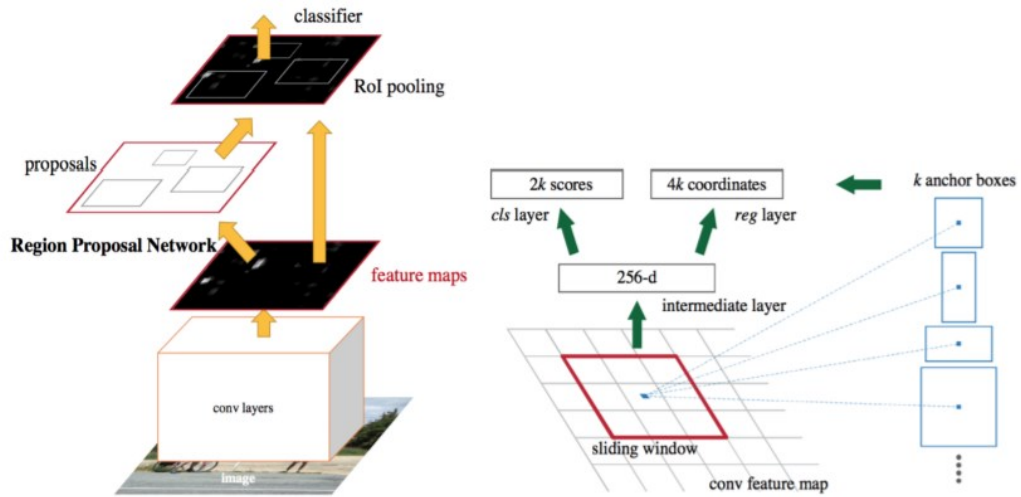


Рисунок 3 – Архитектура Faster R-CNN [96]

Функция потерь детектора Faster R-CNN вычисляется по формуле:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, \hat{p}_i) + \lambda \frac{1}{N_{reg}} \sum_i \hat{p}_i L_{reg}(t_i, \hat{t}_i), \quad (1.16)$$

где i индекс anchor, p_i предсказанная вероятность, что anchor i является объектом, $p \times i$ метка того, что anchor является позитивным (1) или негативным (0), t_i вектор, представляющий собою 4 координаты предсказанного бокса, $t \times i$ индикатор того, что размеченный бокс ассоциирован с позитивным anchor, L_{cls} ошибка классификации, L_{reg} ошибка регрессии.

1.5.4 Одноэтапные методы обнаружения глубоких объектов

В отличие от многоступенчатого процесса двухэтапных методов, одноэтапные методы направлены на одновременное предсказание категории объекта и его положения. По сравнению с двухэтапными методами одноэтапные методы имеют гораздо более высокую скорость обнаружения и сопоставимую

точность обнаружения. Среди одноэтапных методов репрезентативными методами являются YOLO [100] и SSD [99].

Детектор YOLO. YOLO делит входное изображение на сетку размерностью $k \times k$. Каждая ячейка сетки предсказывает N ограничивающих прямоугольников с пересечением истинной области и с вероятностями условного класса. Прогнозы для каждого ограничивающего прямоугольника: (X, Y, W, H, s) , где (X, Y, W, H) задают расположение ограничивающего прямоугольника, а s - показатель достоверности объекта ограничивающего прямоугольника. Таким образом, выходной слой имеет размер тензора $n \times n \times (B \times 5 + c) = 7 \times 7 \times 30$, где $n = 7$ количество выходных ячеек, которые проецированы на изображение, $B = 2$ количество областей (bounding boxes), $5 = x, y, w, h, s$ характеристики области, $C = 20$ количество классов. Основываясь на предсказании ограничивающего прямоугольника и предсказании соответствующего класса, YOLO может одновременно давать вероятность объекта и вероятность категории для каждой ячейки. На рисунке 4 показана архитектура YOLO. Он состоит из 24 сверточных слоев и 2 полностью связанных слоев. Для увеличения скорости обнаружения в некоторых средних слоях используются чередующиеся сверточные слои 1×1 . Размер входного изображения YOLO фиксированный (то есть 448×448).

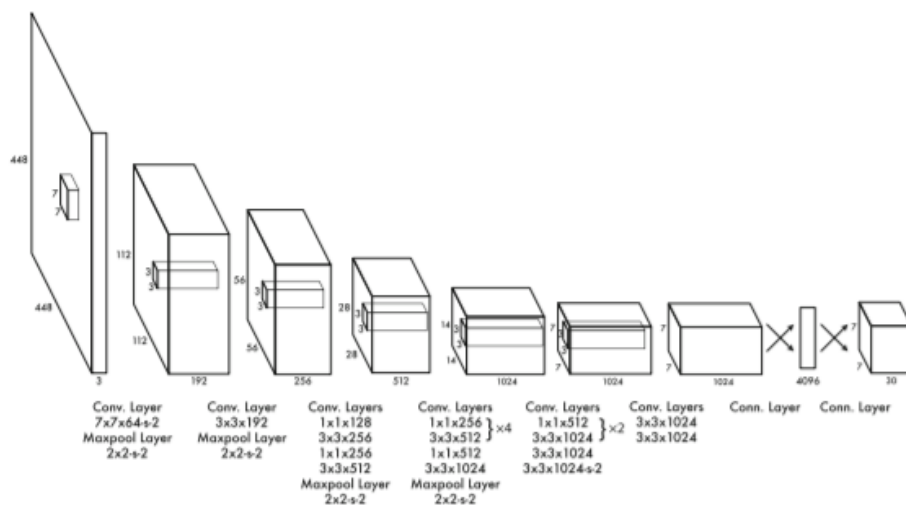


Рисунок 4 – Архитектура нейронной сети YOLO [100]

На этапе обучения первые 20 сверточных слоев YOLO сначала проходят предварительное обучение, а затем вся сеть YOLO настраивается на набор данных обнаружения объектов. YOLO использует ошибку квадрата суммы для оптимизации. Поскольку существует много ячеек сетки, не содержащих объектов, это повлияет на градиент от ячеек, содержащих объекты. Таким образом, вес потери предсказания местоположения бокса увеличивается, тогда как вес потери предсказания бокса уменьшается. Для каждого ограничивающего прямоугольника назначается область истины, с которой он имеет наибольшее перекрытие IoU.

Функция потерь детектора YOLO:

$$\begin{aligned}
 E_{data} = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2 \\
 & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned} \tag{1.11}$$

Детектор SSD (single-shot detector). SSD [99] - однократный детектор для обнаружения обычных объектов. Базовая сеть построена на VGG16 [81] для классификации изображений, но без окончательных уровней классификации. К базовой сети VGG16 добавляются дополнительные сверточные слои для обнаружения. Сверточные слои в конце базовой сети уменьшаются в размерности. Это необходимо для прогнозирования объектов разного масштаба. В частности, для выходного сверточного слоя с заданным разрешением он сначала использует сверточный фильтр 3×3 для создания новых карт характеристик, а затем прогнозирует оценки категорий объектов и местоположения объектов на новых

картах характеристик. Прогнозирование ограничивающих рамок и достоверности для различных объектов на изображении осуществляется не по одной, а по нескольким картам характеристик разного размера, которые представляют несколько масштабов. Постепенно уменьшающиеся сверточные слои уменьшают размер карты объектов и увеличивают глубину. Глубокие слои покрывают более крупные рецептивные поля и создают более абстрактные представления. Это полезно при обнаружении более крупных объектов. Начальные сверточные слои покрывают меньшие воспринимающие поля и помогают обнаруживать более мелкие объекты, присутствующие на изображении.

Для определения местоположения объектов используется концепция опорных областей «anchor boxes». В SSD по умолчанию использует 8732 опорных областей разного размера и соотношением сторон. На рисунке 5 показана архитектура SSD. Это помогает обнаружить ограничивающую рамку, которая больше всего перекрывается с размеченной ограничивающей рамкой, содержащей объекты. Предполагая, что количество классов объектов равно «с» и каждая карта признаков предсказывает k объектов, это приведет к выходному вектору (dcx , dcy , w , h) для заданных карт признаков размером $m \times n$. При решении задачи регрессии участвуют только те области, для которых Intersection over Union (IoU) между сформированной и размеченной (ground truth) областями составляет более 0,5. Для них вычисляется разница ошибки и выбирается область с наименьшей разницей.

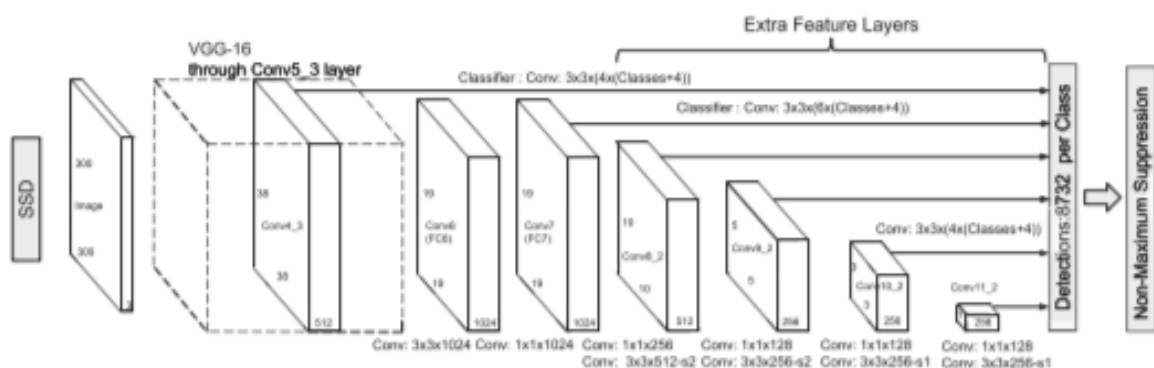


Рисунок 5 – Архитектура нейронной сети SSD [99]

Функция потерь детектора SSD:

$$L(x, c, l, g) = \frac{1}{N} (L_{con}(x, c) + \alpha L_{loc}(x, l, g)), \quad (1.12)$$

где N - количество ограничивающих областей (bounding box), для которых наблюдалось перекрытие с истинным расположением объекта, c – вероятность, принадлежности объекта к искомому классу, g – размеченная зона объекта (ground truth box), l – предсказанная зона объекта (predicted box), L_{loc} – функция ошибки между l и g , L_{conf} – потери при определении объекта (confidence loss).

Несмотря на высокое качество обнаружения объектов при решении конкретных задач, выбор нейросетевой модели зависит от требований к скорости работы, точности обнаружения объектов и входных данных. Рассмотренные модели нейронных сетей имеют свои преимущества и недостатки. В связи с перспективностью применения НС при решении задачи обнаружения объектов, требуется проведение дополнительных исследований с целью выявления наиболее эффективной модели обнаружения пожароопасных объектов.

1.6. Опорные области для обнаружения объектов

Опорные области (anchor boxes) – это набор предопределенных ограничивающих рамок определенной высоты и ширины. Опорные области задаются для определения масштаба и соотношения сторон определенных классов объектов. Опорные области обычно выбираются на основе размеров объектов в ваших наборах данных для обучения. Во время обнаружения предопределенные области располагаются по всему изображению. Сеть предсказывает вероятность и другие атрибуты, такие как фон, пересечение по объединению (IoU) и смещения для каждого мозаичного опорного блока. Прогнозы используются для уточнения каждого отдельного опорного блока.

Нейросети не прогнозируют напрямую ограничивающие прямоугольники, а скорее прогнозируют вероятности и уточнения, которые соответствуют мозаичным опорным областям. Нейросеть возвращает уникальный набор прогнозов для каждой определенной области. Последняя карта функций представляет обнаружение объектов для каждого класса. Использование опорных блоков

позволяет сети обнаруживать несколько объектов, объекты разного масштаба и перекрывающиеся объекты.

Каждый блок выложен плиткой по изображению. Количество сетевых выходов равно количеству опорных блоков, выложенных плиткой (см. рисунок 6).

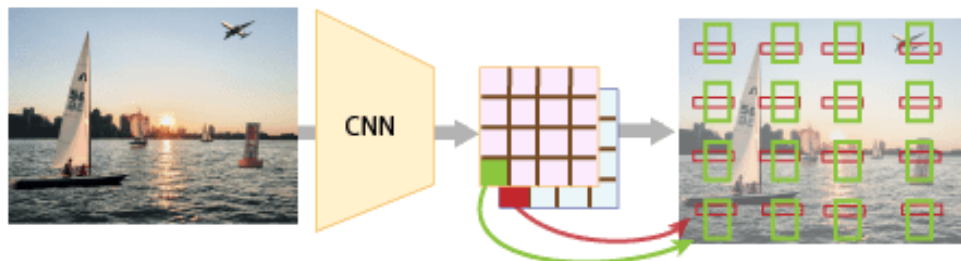


Рисунок 6 – Схема предсказания класса опорных блоков

Расстояние или шаг между мозаичными якорными блоками является функцией количества субдискретизации, присутствующей в СНС. Часто встречаются коэффициенты понижения дискретизации от 4 до 16, см. рисунок 7. Эти факторы понижающей дискретизации создают грубо замощенные блоки привязки, что может привести к ошибкам локализации.

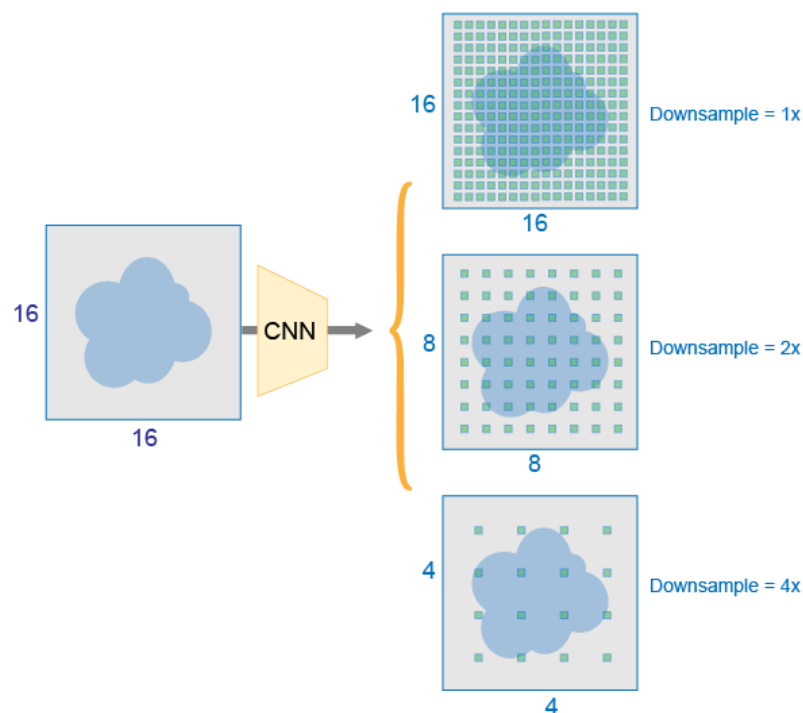


Рисунок 7 Пример влияния понижения дискретизации от 4 до 16.

Чтобы исправить ошибки локализации, детекторы объектов глубокого обучения прогнозируют смещения, применяемые к каждому мозаичному блоку привязки, уточняя положение и размер блока привязки рисунок 8.

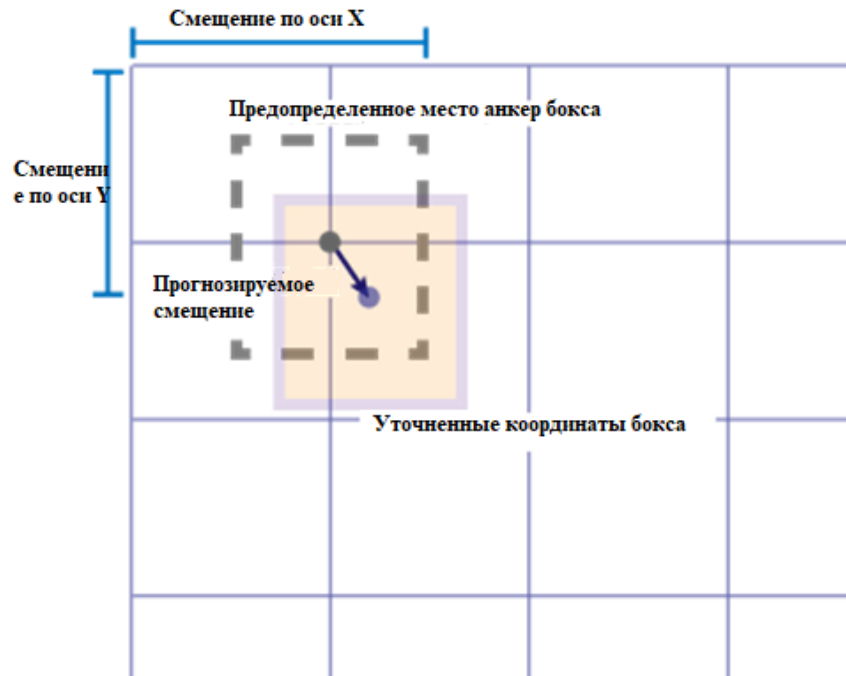


Рисунок 8 – Пример уточнения координат предсказанного бокса

Чтобы окончательно выделить обнаруженные объекты, мозаичные поля привязки, принадлежащие фоновому классу, удаляются, а оставшиеся фильтруются по их степени достоверности. Блоки привязки с наибольшей степенью достоверности выбираются с использованием немаксимального подавления (NMS), см. рисунок 9.



Рисунок 9 – Пример обнаружения объектов с использованием немаксимального подавления (NMS)

1.7. Локализация ограничивающей рамки, объединение предсказаний

Целью обнаружения объектов, как части алгоритма обнаружения и классификации, является точная локализация объектов путем максимизации IOU или какой-либо другой метрики. Поскольку ограничивающая рамка (в дальнейшем также обозначаемая BB) представляет собой грубую оценку местоположения целевых объектов, возможен ряд улучшений. Крайне важно уточнить BB, исключаемые ограничивающие рамки, которые представляют один и тот же объект, а также принять во внимание тот факт, что пиксели фона включены в BB.

Немаксимальное подавление (NMS) [103]. Обнаружение дубликатов, то есть нескольких перекрывающихся BB, содержащих один и тот же экземпляр объекта, является распространенной проблемой. В качестве этапа последующей обработки используется алгоритм NMS для устранения фоновых BB и получения окончательного BB.

Предсказания модели в задаче обнаружения изображения состоят из координат прямоугольника ограничивающей рамки, метки класса объекта и оценки достоверности (с вероятностью от 0 до 1), которая отражает, насколько модель уверена в этом предсказании.

В методе NMS блоки считаются принадлежащими одному объекту, если их перекрытие, пересечение по объединению (IoU) превышает некоторое пороговое значение. Таким образом, процесс фильтрации боксов зависит от выбора этого единственного порогового значения IoU , которое влияет на производительность модели. Определить данный порог сложно: если рядом есть объекты с меньшей вероятностью, они будут удалены. На рисунке 10 показан пример, иллюстрирующий один из таких случаев.

Алгоритм NMS на вход принимает набор гипотез для одного класса и порог, задающий величину максимального пересечения между гипотезами.

Этапы работы алгоритма NMS:

1. Сортировка обнаруженных объектов по убыванию вероятности (Score).

2. В цикле выбирается объект (который имеет наибольшую величину уверенности) и добавляется в результирующий набор.
3. Проверка всех оставшихся объектов. Если IoU между текущим объектом и выбранным ранее объектом больше определенного порога, то этот объект удаляется из списка.
4. Повторение шагов 2-3 до тех пор, пока все объекты не будут проверены.

Soft-NMS [104]. Метод Soft-NMS частично решает проблему удаления ограничивающих рамок. Алгоритм выполняет снижение веса обнаруженных объектов, которые сильно перекрываются друг с другом, чтобы уменьшить число ложных обнаружений.

Этапы работы алгоритма Soft-NMS:

1. Сортировка всех обнаруженных объектов по убыванию их оценки уверенности (Score).
2. Выбор объекта с наивысшей оценкой и удаление его из списка обнаруженных объектов.
3. Проверка оставшихся объектов на перекрытие с удаленным объектом. Если объекты перекрываются, то уменьшение веса оставшихся объектов в зависимости от степени перекрытия.
4. Повторение шагов 2-3 до тех пор, пока все объекты не будут удалены из списка.

Алгоритм Soft-NMS использует гауссово распределение для вычисления нового веса объекта на основе его старого веса и степени перекрытия с другими объектами. Это позволяет сохранить информацию о местоположении и размере объектов, которые были удалены из списка обнаруженных объектов.

Взвешенные коробки Weighted Boxes Fusion (WBF) [105]. В случае алгоритмов NMS и Soft-NMS исключаются некоторые предсказания. Алгоритм WBF использует все поля. Таким образом, он может исправить случаи, когда все поля предсказываются неточно всеми моделями. Этот случай проиллюстрирован

на рисунке 10, NMS, Soft-NMS оставит только один неточный блок, в то время как WBF объединит его, используя все предсказанные блоки.

Этапы работы алгоритма WBF:

1. На вход алгоритма подается набор ограничивающих рамок и их соответствующие оценки уверенности (Scores), полученные от различных моделей обнаружения объектов.
2. Ограничивающие рамки группируются по классам объектов.
3. Для каждого класса объектов выполняется следующее:
 - а. Определяется набор пересекающихся ограничивающих рамок.
 - б. Вычисляется взвешенное среднее положение и размеры рамок в наборе, используя их оценки уверенности в качестве весов.
 - с. Создается новая ограничивающая рамка с полученными координатами и оценкой уверенности, равной сумме оценок уверенности всех рамок в наборе.
4. Повторяется шаг 3 для каждого класса объектов.
5. Полученные ограничивающие рамки объединяются в единый набор и сортируются по убыванию оценок уверенности.
6. Применяется алгоритм NMS для удаления дублирующихся ограничивающих рамок.
7. Возвращается финальный набор ограничивающих рамок.

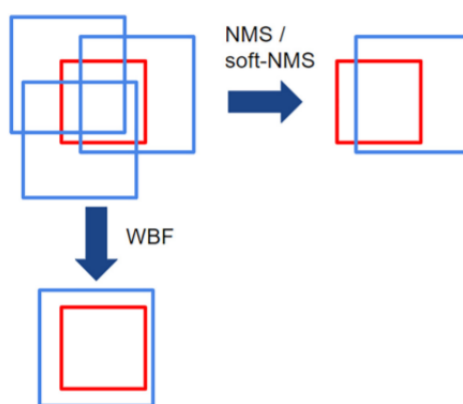


Рисунок 10 – Схематическая иллюстрация работы алгоритмов:

NMS, Soft-NMS и WBF. Синие боксы – предсказания различных моделей, красный бокс – абсолютно истинный бокс.

Алгоритм WBF позволяет эффективно объединять результаты от различных моделей обнаружения объектов, учитывая их оценки уверенности и устраняя дубликаты. Это позволяет получить более точные результаты объектного обнаружения и сегментации.

1.8. Рекуррентные нейронные сети

Методы, основанные на CNN, описанные выше, обычно обнаруживают дым от лесных пожаров путем применения CNN к неподвижным изображениям. Поскольку изображения лесных пожаров в основном делаются с камеры дальнего действия, скорость движения очень низкая, а с точки зрения внешнего вида характеристики схожи с характеристиками облаков и тумана. Отличительной чертой дыма от лесного пожара является то, что он поднимается вверх и распространяется с течением времени. Поэтому, чтобы повысить точность обнаружения дыма от лесных пожаров, необходимо отличать дым от других подобных дыму случаев, улавливая характеристики движения и рассеивания дыма в последовательных кадрах. Чтобы отразить динамические характеристики дыма от лесных пожаров, были предложены методы обнаружения лесных пожаров, основанные на CNN и рекуррентных нейронных сетях (RNN) [106], которые учитывают информацию о временных рядах.

Рекуррентная нейронная сеть (RNN) — это класс нейронной сети, предназначенный для распознавания шаблонов в последовательностях данных. Эти сети принимают на вход текущий вход, а также информацию с предыдущего временного шага последовательности. У RNN есть петля обратной связи, которая связывает каждый временной шаг с прошлыми решениями, поэтому обычно говорят, что у RNN есть память, поскольку они позволяют информации сохраняться. Последовательная информация сохраняется в скрытом состоянии сети. На рисунке 11 изображена развернутая модель RNN.

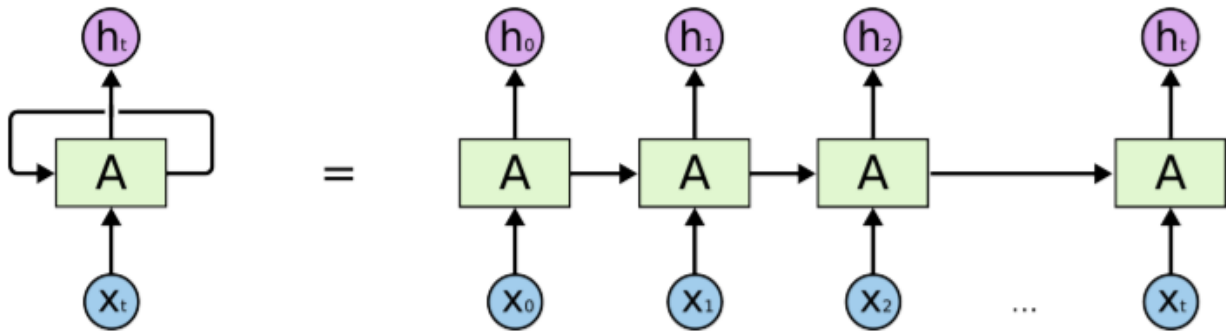


Рисунок 11 – Развернутая RNN. Имеет представление как несколько копий одной и той же сети, которые передают информацию от одного шага к другому.

Математическая формулировка для определения скрытого состояния обычной RNN показана в уравнении 2, где h_t — скрытое состояние RNN на шаге t . Скрытое состояние является функцией входа на этом же временном шаге x_t , модифицированного весовой матрицей W , а затем добавленного к скрытому состоянию с предыдущего временного шага h_{t-1} , модифицированного другой весовой матрицей U . Весовые матрицы W и U определить, сколько информации нужно сохранить из текущего ввода и из предыдущих скрытых состояний.

$$h_t = \phi(Wx_t + Uh_{t-1}). \quad (1.17)$$

Популярным типом RNN являются сети с долговременной кратковременной памятью (LSTM) [107]. Блок LSTM состоит из ячейки, входного затвора, выходного затвора и затвора забывания. Клетка способна запоминать значения с течением времени, а различные ворота регулируют поток информации в ячейку и из нее. В отличие от ванильных RNN, LSTM популярны благодаря запоминанию долгосрочных зависимостей. Ворота LSTM реализованы с полностью связанными слоями. Если вместо этого они представляют собой сверточные слои, то это модуль Convolutional LSTM (ConvLSTM) [108].

Рекуррентные нейросети имеют большой потенциал для распознавания объектов в видеопоследовательности. Они могут учитывать хронологическое изменение последовательности данных, что делает их более эффективными по сравнению с другими методами машинного обучения. Однако, в настоящее время

остаются нерешенные проблемы, связанные с обучением рекуррентных нейросетей, например, такие как затухание градиентов и переобучение, которые могут затруднить достижение высокой точности распознавания.

1.9. Используемые метрики оценки качества детектирования объектов

Для проверки обоснованности предлагаемых методов и оценки эффективности работы рассматриваются различные показатели на основе детального изучения литературы по современным подходам. Четыре возможных события рассматриваются для расчета различных показателей эффективности. Для получения классификации события пожара вычисляются истинно положительные (TP) и истинно отрицательные (TN), а для классификации событий, не связанных с пожаром, вычисляются ложноположительные (FP) и ложноотрицательные (FN). Результат TP указывает на то, что событие точно идентифицировано как событие пожара, тогда как TN указывает на то, что событие точно идентифицировано как событие, не являющееся пожаром. С другой стороны, результат FP указывает, что изображение прогнозируется как изображение пожара, но на самом деле это не изображение пожара, тогда как результат FN указывает, что изображение идентифицировано как изображение без возгорания, но на самом деле это изображение, затронутое пожаром.

Для измерения степени перекрытия, предсказываемой детектором области объекта с истинной областью (ground truth), используется величина IoU (intersection over union):

$$IoU = \frac{S_{overlap}}{S_{union}}, \quad (1.18)$$

где $S_{overlap}$ – площадь перекрытия двух областей, S_{union} – общая площадь областей.

Precision (точность): это отношение всех положительных прогнозов, так как учитываются только положительные прогнозы. Он измеряется как отношение истинно положительных результатов к общему количеству прогностических положительных результатов. Процентный диапазон точности лежит в диапазоне от

0 до 100%. Точность используется для измерения отклонения входных данных от их реального значения и оценки того, насколько точна и точна модель.

$$Precision = \frac{TP}{TP + FP}. \quad (1.19)$$

Recall (полнота) также известна как чувствительность, которая используется для измерения возможностей модели, определяющих, сколько фактических положительных результатов может зафиксировать модель. Данная метрика измеряется как отношение истинно положительных результатов к общему количеству фактических положительных результатов. Результат чувствительности обеспечивает меру способности модели к правильной классификации, чтобы идентифицировать истинные пиксели огня. Процентный диапазон отзыва лежит в пределах от 0 до 100% и больше; чем ближе отзыв к 100%, тем больше становится способность системы правильно обнаруживать события пожара.

$$Recall = \frac{TP}{TP + FN}. \quad (1.20)$$

F-score (среднее гармоническое) — это функция Precision и Recall, которая измеряется для обеспечения эффективности выполнения классификации. F-score учитывает как ложноположительные, так и ложноотрицательные события и находит баланс между значениями Precision и Recall. Он измеряется как взвешенная сумма результатов точности и отзыва. Процентный диапазон F-score находится в диапазоне от 0 до 100%, и чем ближе F-score к 100%, тем больше становится способность системы правильно прогнозировать ложные возгорания.

$$F_{score} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}. \quad (1.21)$$

Для классификации при тестировании или обучении используются значения потерь (loss) и точности (Accuracy). Loss – это сумма внутренних значений ошибок сопоставления для каждого образца класса на обучающей или тестовой выборках. Его значения зависят от функции сравнения, вычисляющей ошибку, которую

необходимо уменьшать в процессе обучения. Ниже представлена формула расчета бинарной кросс энтропии:

$$BCE(pt, pp) = -\frac{1}{N}(pt \cdot \log(pp) + (1 - pt) \cdot \log(1 - pp)), \quad (1.22)$$

где pt – распределение истинных ответов, pp – распределение прогнозируемых ответов.

Accuracy – измеряется как отношение правильно предсказанных наблюдений к общему количеству наблюдений и выражается. Процентный диапазон Точности лежит в пределах от 0 до 100%, и чем больше Точность приближается к 100%, тем больше становится способность системы точно прогнозировать истинные события пожара.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1.23)$$

mAP (mean average precision) средняя точность детектирования каждого класса при одном или нескольких значения IoU.

$$mAP = \frac{1}{N_{classes}} \sum_c AP(c), \quad (1.24)$$

где $AP(c)$ – средняя точность для одного класса по всем значениям Precision от Reccal.

$$AP = \frac{1}{N_{steps}} \sum_{Reccal \in \{0, step_0, \dots, 1\}} Precision(Recall), \quad (1.25)$$

Для классификации при тестировании или обучении используются значения функции потерь (loss) и точности (Accuracy). Loss это сумма внутренних значений ошибок сопоставления для каждого образца класса на обучающей или тестовой выборках. Его значения зависят от функции сравнения, вычисляющей ошибку, которую необходимо уменьшать в процессе обучения. Accuracy - это процент

правильно распознанных объектов. Обычно измеряется на проверочной выборке (validation dataset).

1.10. Выводы по первой главе

1. Исследованы методы распознавания динамических объектов (в том числе, огня и дыма) на изображениях с оптической камеры. Выявлены существующие проблемы использования алгоритмов детектирования на основе цвета и выделения динамических признаков.
2. Описаны известные современные нейросетевые методы обнаружения объектов на изображениях. Рассмотрены модели рекуррентных нейронных сетей, решающие задачу выделения временной составляющей движения дымового облака.
3. Обзор литературы о методах и моделях обнаружения динамических объектов позволил выделить глубокие нейронные сети, как наиболее предпочтительный при решении задачи обнаружения на кадрах видеоряда объектов, представляющих дым и огонь, поскольку преимуществом глубоких нейросетей является способность извлекать высокоуровневые признаки из входных данных, что позволяет достичь более точных результатов. Кроме того, глубокие нейронные сети позволяют адаптироваться к изменениям в данных, что делает их более универсальными и эффективными в решении задач обнаружения динамических объектов на изображениях. Использование глубоких нейронных сетей для выделения динамических объектов является перспективным направлением и может привести к значительному увеличению точности обнаружения объектов.
4. Проведен анализ методов объединения предсказаний, целью которых является точное выделение исследуемого объекта с нескольких кадров видеопоследовательности. На основе проведенного исследования сделан вывод о том, что для повышения качества распознавания наиболее эффективным является ансамблирование методов машинного обучения.

5. Показано, что существует потребность в разработке методов и алгоритмов обнаружения объектов, обладающих динамическими признаками на изображениях видеоряда.

ГЛАВА 2

ТЕХНОЛОГИЯ ОБНАРУЖЕНИЯ ОБЪЕКТОВ ВИДЕОРЯДА НА ОСНОВЕ ТЕХНОЛОГИИ МАШИННОГО ОБУЧЕНИЯ (НА ПРИМЕРЕ ЛЕСНЫХ ПОЖАРОВ)

В этой главе представлены технология и алгоритмы системы обнаружения объектов в видеоряде на основе средств машинного обучения. А именно: технология анализа изображений в видеопоследовательности, на основе предложенных алгоритмов локализации и классификации с учетом статических и динамических данных; алгоритм, позволяющий объединять разрозненные предсказания нейросети в итоговое решение; новая гибридная архитектура нейронной сети, основанная на интеграции данных сверточной нейронной сети в рекуррентную нейронную сеть, отличающаяся от существующих неявным представлением временной составляющей, агрегированной аугментацией данных и динамическим подбором размера батча в зависимости от скорости сходимости модели на прошлых итерациях, что позволяет осуществлять классификацию изображений в видеопотоке с высокой точностью.

2.1. Разработка алгоритма обнаружения дымового облака

Разрабатываемая технология обнаружения пожароопасных объектов в лесном массиве на основе данных камер наземного видеомониторинга представлена на рисунке 12 и состоит из следующих блоков:

1. Предварительная обработка изображений:

- чтение видеозаписи продолжительностью 10 секунд с последующем разбиением на кадры;
- формирование кадров для анализа с помощью алгоритма выделения динамических признаков (предложенный алгоритм). Основная задача алгоритма – фильтрация полученной информации, а именно объединение последовательности кадров в один элемент с исключением признаков

объектов со статическими признаками. Алгоритм получает на вход 7 изображений, в результате обработки алгоритм вырезает четыре изображения, относящихся к равным временным интервалам последовательности.

2. Выделение на кадрах объектов анализа:

- обнаружения объектов исследования в видеоряде и заключение их в ограничивающую рамку;
- объединение разрозненных объектов анализа в один объект на результирующем кадре посредством использования алгоритма кластеризации площадей обнаруженных объектов.

3. Фильтрация обнаруженных объектов:

- подготовка данных для классификации обнаруженных объектов. Полученные координаты и вероятности объектов обнаружения переносятся на кадры исходной видеозаписи. Из последовательности изображений выбираются n кадров, равно распределенных по времени. Из них вырезаются области по полученным координатам, и формируется последовательность из фрагментов изображений;
- классификация с помощью алгоритма фильтрации ранее обнаруженных объектов на основе гибридной архитектуры нейронной сети;
- визуализация результатов.

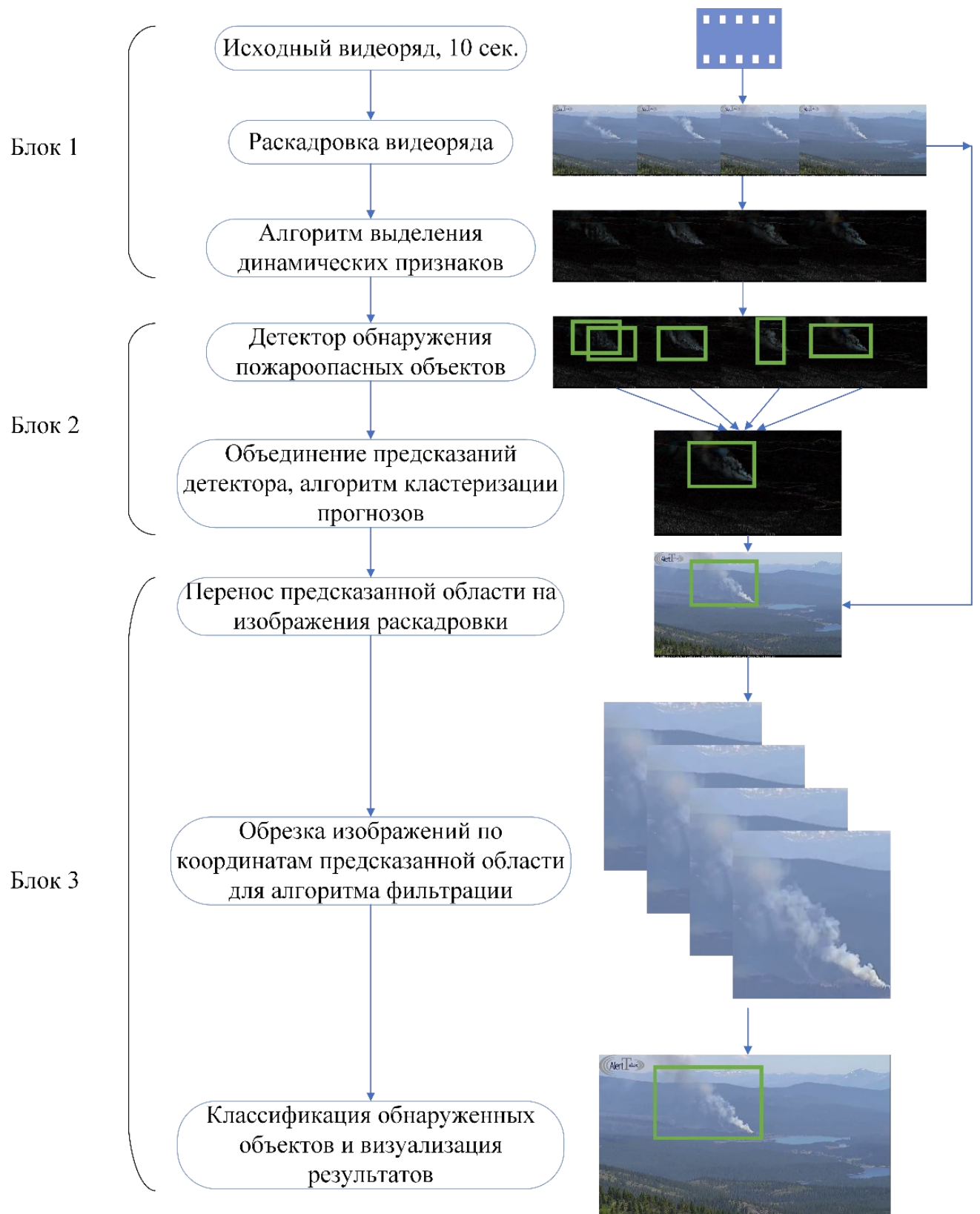


Рисунок 12 – Блок- схема архитектуры технологии раннего обнаружения пожаров в лесном массиве

2.2. Разработка алгоритма выделения динамических признаков объекта

Дым является продуктом сочетания газов, твердых и жидких частиц и имеет статические и динамические характеристики. Первое связано с цветом, текстурой и неровностями, а второе — с движением, изменением направления и скорости. Для получения информации о типе и локализации пожара в лесной обстановке необходимо знать местонахождение задымления, анализировать его цвет, плотность, высоту, объем, направление и т. д. Фактически лесные пожары выявляются по анализу статических и динамических характеристик. При этом необходимо отметить, что динамические признаки на видеоизображении несут больше информации о задымлении, чем статические. Такие признаки как цвет, текстура, неровности напрямую связаны с условиями освещения, типа возгорания и площади.

В отличие от дыма, который возникает в помещении, в случае дыма от лесных пожаров, ПЗС-камера устанавливается на высоком месте с возможностью максимального охвата территорий. В результате этого возникает проблема: в случае даже если действительно возникает дым от лесных пожаров, он часто слишком мал, чтобы его можно было идентифицировать. Кроме того, поскольку фокусное расстояние ПЗС-камеры велико, дым движется медленнее, чем в действительности. При этом применение сверточных нейронных сетей CNN, позволяет оценить статические признаки, при этом теряя динамические признаки. Это часто приводит к ложно положительным и позитивно отрицательным результатам. Предобработка кадров, отправляемых на детектор, позволяет частично решить проблему анализа динамических признаков, тем самым повысив точность вычисления детекций на изображениях.

Для выделения динамических признаков на изображении реализована операция разности кадров, преимуществом которой является нечувствительность к изменениям сцены (например, к освещению) и возможность адаптации к различным динамическим средам с хорошей стабильностью. Однако, серьезным

недостатком является невозможность извлечения всей площади объекта. В связи с этим предложена модификация операции разности кадров. Последовательность работы алгоритма представлена на рисунке 13 и состоит из следующих этапов:

1. Последовательность кадров делится на n равных интервалов, из которых выбираются значения верхней границы интервала.
2. Операция преобразования кадров из трех каналов RGB в один канал HBR (переход в градации серого), что позволяет экономить время вычислений на следующих этапах.
3. Операция инициализации "усредненного кадра" по формуле:

$$F = \{F(x, y), 0 \leq x < w, 0 \leq y < h\}, \quad (2.1)$$

$$F_c(x, y) = \frac{1}{N} * \sum_{i=1}^N F_i(x, y), \quad (2.2)$$

где x – количество пикселей, соответствующих ширине кадра, y – количество пикселей, соответствующих высоте кадра, w – максимальная ширина кадра, h – максимальная высота кадра $F_c(x, y)$ – "усредненный кадр", N – общее число, обрабатываемых кадров. $F_i(x, y)$ – текущий кадр последовательности.

Данная операция позволяет нивелировать проблему шумности камер и получать более стабильный результат.

4. Операция выделения динамических признаков на основе вычитания исходного кадра из усредненного:

$$F_{di}(x, y) = |F_c(x, y) - F_i(x, y)|, \quad i = 1, 2, \dots, N, \quad (2.3)$$

где $F_{di}(x, y)$ – разность текущего кадра последовательности и усредненного.

5. Операция шумоподавления. Данная операция позволяет выделить объекты с большей динамикой, отбросив при этом посторонние шумы. Фильтрация изображения с адаптивным порогом:

$$F_{\text{ii}}(x, y) = \begin{cases} m, & \text{если } F_{\text{ii}}(x, y) > th \\ 0, & \text{если нет} \end{cases}, \quad (2.4)$$

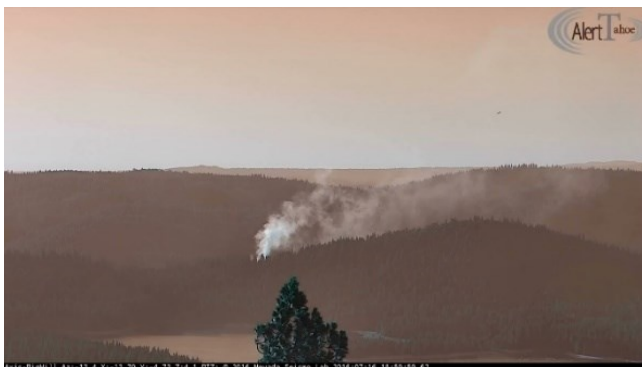
где $F_{\text{ii}}(x, y)$ – результирующий кадр операции шумоподавления, th – пороговое значение яркости, m – заданное значение яркости



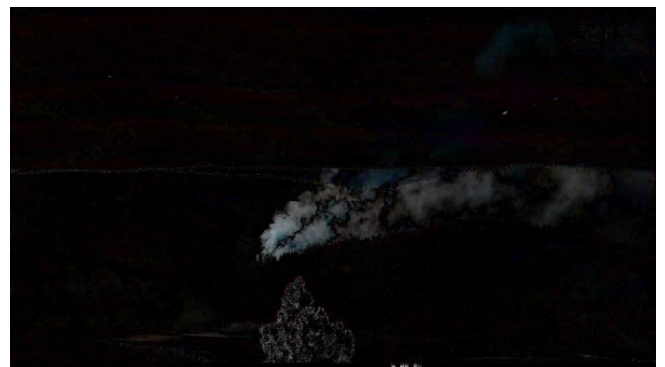
а) Исходное изображение



б) перевод в канал HBR



в) Обработка алгоритмом фильтрации
шума



г) Вычитание фона

Рисунок 13 – Визуальное представление работы алгоритма выделения динамических признаков

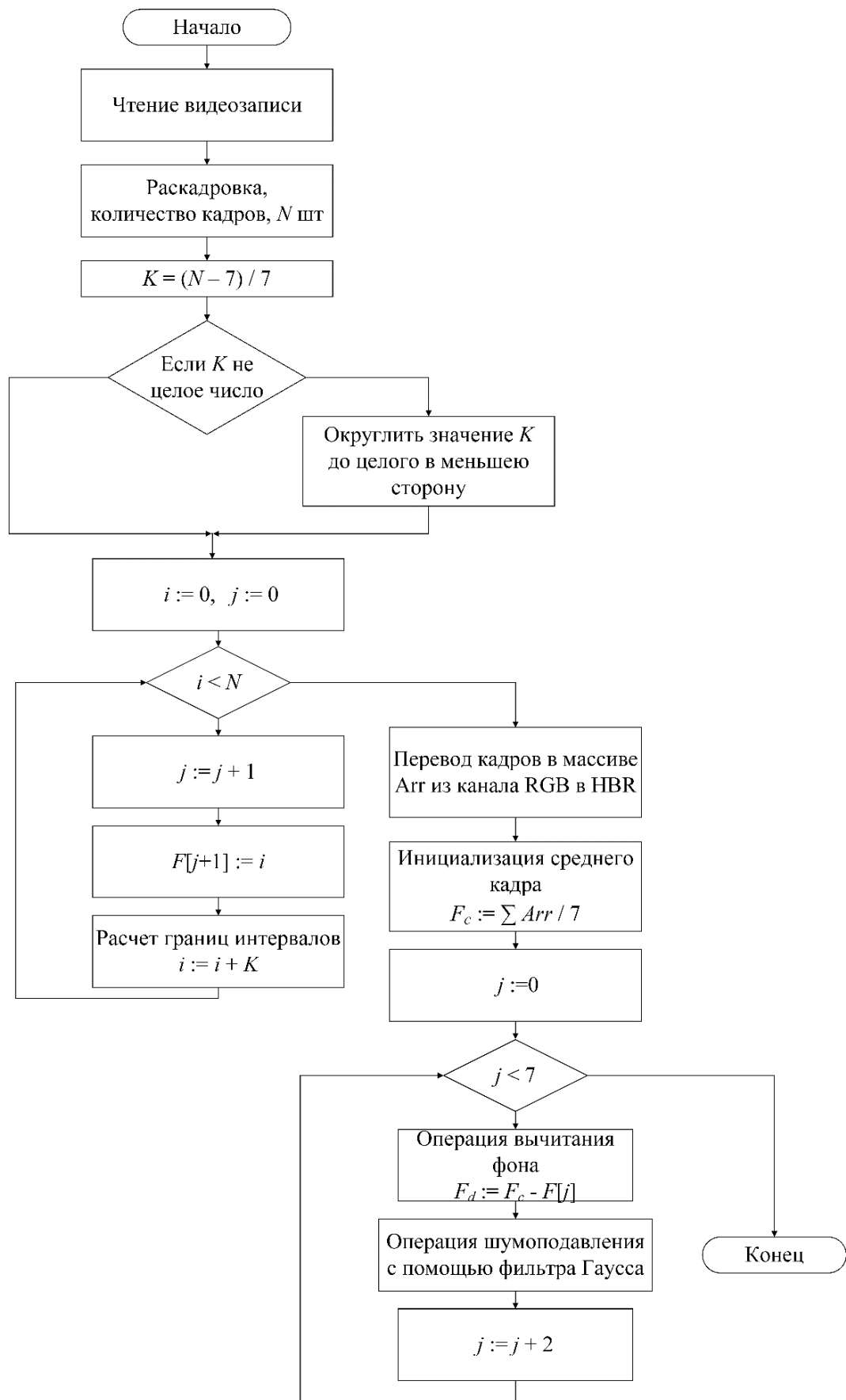


Рисунок 14 – Блок- схема алгоритма выделения динамических признаков объектов

2.3. Разработка алгоритма локализации объектов на изображении

2.3.1 Трекинг объектов детектором EfficientDet

Обнаружение объектов на изображении является достаточно сложной задачей, для выполнения которой, необходимо решить задачу классификации и локализации объекта. Современные модели детекторов зачастую с высокой эффективностью справляются с поставленной задачей. Определение дымового облака - задача более сложная. Отсутствие постоянных статических признаков у дымового облака, а также негативные эффекты, связанные с видеосъемкой, не позволяют реализовать полноценную систему используя только детектор. Поэтому предлагается алгоритм, основанный на использовании авторского алгоритма предобработки изображения (алгоритм выделения динамических признаков), применение которого позволяет объединить на изображении статические и динамические признаки, а также анализировать последовательность кадров. Далее с помощью детектора, в основе которого лежит модель EfficientDet-D0, выделяются области кандидаты на анализируемых изображениях. Затем с помощью авторского алгоритма кластеризации прогнозы предсказания с детектора объединяются в один или несколько боксов в зависимости от количества предполагаемых очагов возгорания.

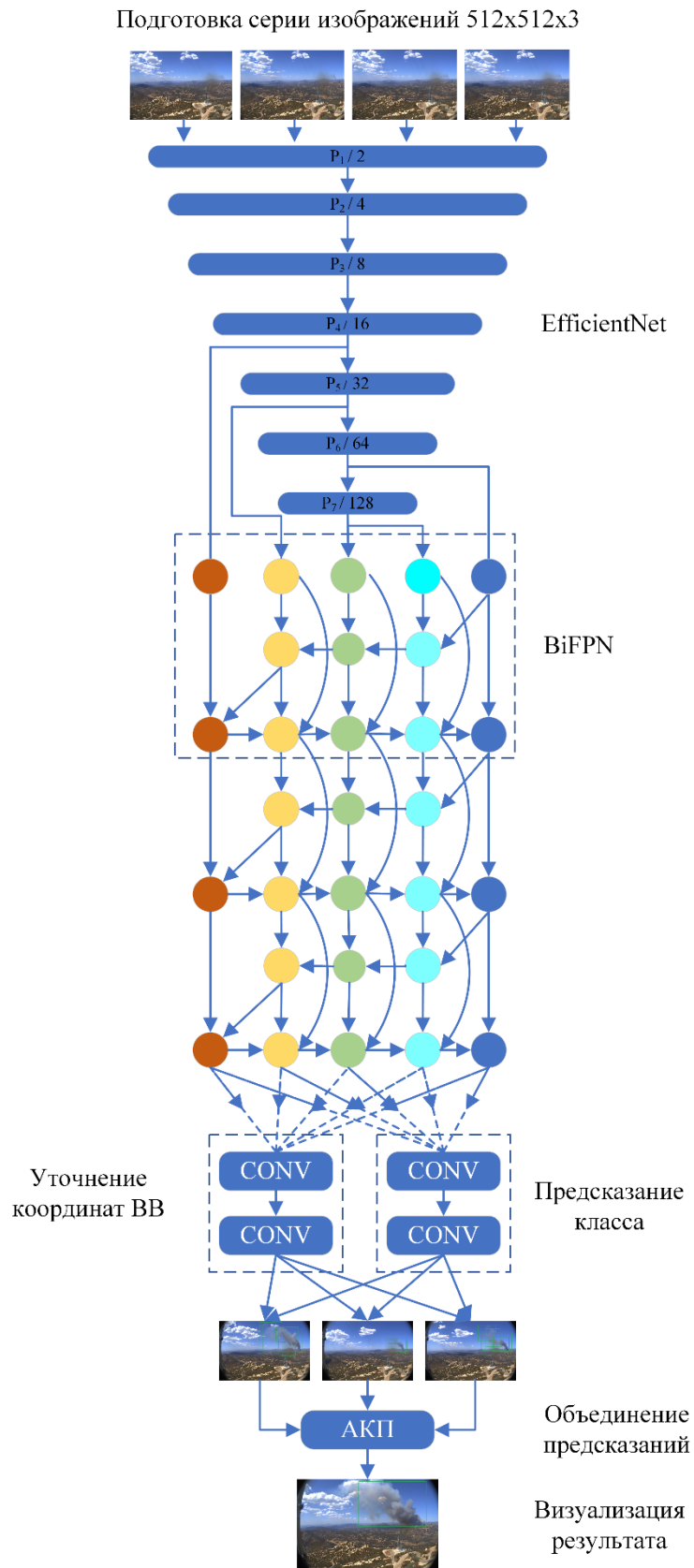


Рисунок 15 – Схема работы метода обнаружения областей кандидатов на изображении

Для выбора детектора, подходящего для дыма лесных пожаров, мы сравнили и оценили различные недавно предложенные детекторы объектов на основе CNN, такие как Faster-RCNN [96], SSD [109], EfficientDet [110]. Для обнаружения пожароопасных объектов была использована технология распознавания объектов, а именно детектор EfficientDet. Отличительной особенностью данного детектора от описанных в главе 1, является алгоритм масштабирования блоков объединения признаков, сверточной нейронной сети, трекинга и предсказания классов объектов.

Детекторы семейства EfficientDet – это одноступенчатые детекторы, очень похожие на SSD и RetinaNet. В основе которых лежит сверточная нейронная сеть EfficientNet [111], предварительно обученная на датасете ImageNet. В качестве блока объединения признаков используется BiFPN [112], которая берет функции уровней 3–7 {P3, P4, P5, P6, P7} из сверточной сети и многократно применяет двунаправленное слияние функций сверху вниз и снизу вверх. Объединенные функции передаются в сеть классов и блоков для получения прогнозов классов объектов и ограничивающих блоков соответственно. Веса класса и сети блоков являются общими для всех уровней функций.

В качестве базовой сети используется сверточная нейронная сеть EfficientNet-B0, основанная на перевернутых остаточных блоках MobileNet-V2 [113]. Архитектура сверточной нейронной сети EfficientNet-B0 представлена на рисунке 16.

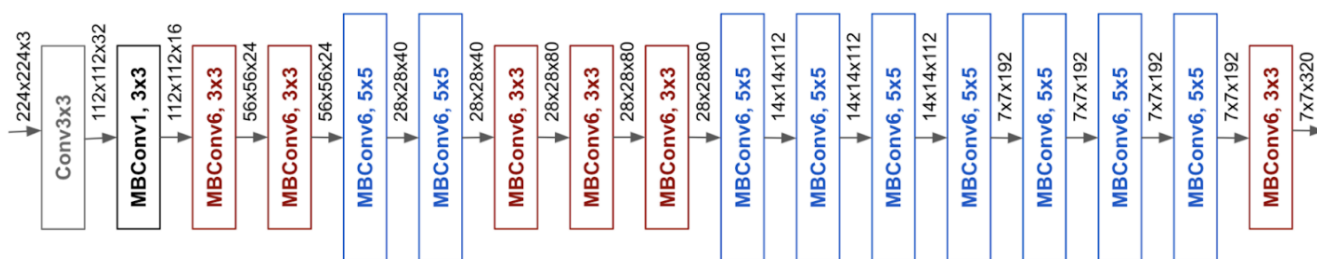


Рисунок 16 – Архитектура EfficientNet-B0

Отличительной чертой архитектур семейства EfficientNet является использование составного коэффициента масштабирования Φ для равномерного

масштабирования размеров сверточной сети, формула (2.5). На рисунке 17 представлен пример масштабирования модели.

$$\text{глубина: } d = \alpha^\phi$$

$$\text{ширина: } w = \beta^\phi \quad (2.5)$$

$$\text{разрешение: } r = \gamma^\phi$$

где $\alpha \geq 1, \beta \geq 1, \gamma \geq 1, \text{ s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$

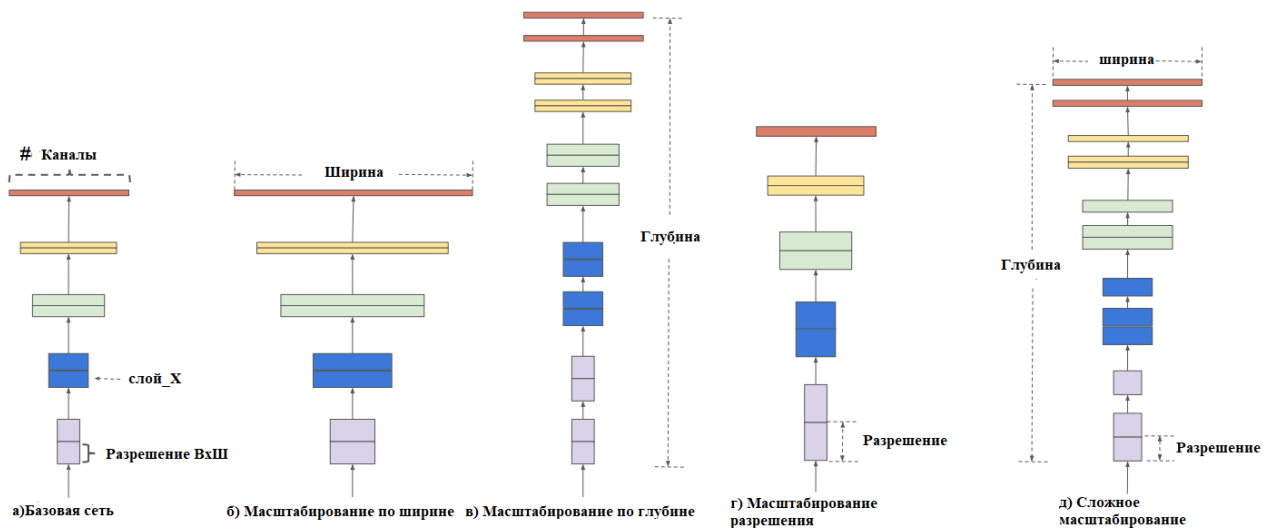


Рисунок 17 – Масштабирование модели. (а) – пример базовой сети; (б), (г) – обычное масштабирование, которое увеличивает только одно измерение ширины, глубины или разрешения сети. (д) – алгоритм сложного масштабирования, который равномерно масштабирует все три измерения с фиксированным соотношением.

Блок объединения признаков состоит из сети BiFPN (Взвешенная двунаправленная пирамидальная сеть признаков). BiFPN представляет собой тип пирамидальной сети признаков, который позволяет легко и быстро объединять многомасштабные признаки. Он включает в себя идею слияния многоуровневых функций от FPN, PANet и NAS-FPN. Это в свою очередь позволяет информации течь как сверху вниз, так и снизу вверх, используя при этом регулярные и эффективные соединения. По сравнению с PANet в PANet добавлен

дополнительный восходящий путь для потока информации за счет увеличения вычислительных затрат. В то время как BiFPN оптимизирует эти межмасштабные соединения, удаляя узлы с одним входным ребром, добавляя дополнительное ребро от исходного входа к выходному узлу, если они находятся на одном уровне, и рассматривая каждый двунаправленный путь как один слой функциональной сети, повторяя его несколько раз для более высокого уровня будущего синтеза, см. рисунок 18.

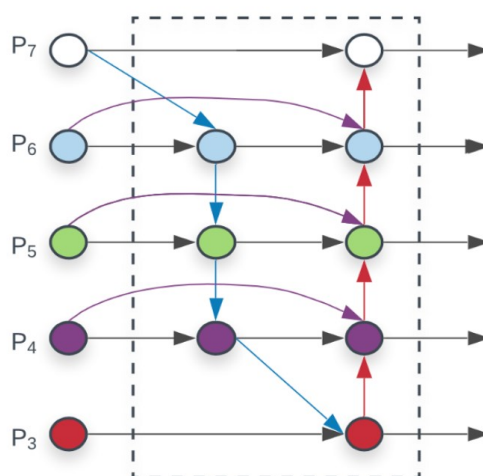


Рисунок 18 – Структура BiFPN

Традиционные подходы обычно одинаково обрабатывают все функции, вводимые в FPN, даже с разными разрешениями. Однако входные функции с разным разрешением часто имеют неодинаковый вклад в выходные функции. Таким образом, BiFPN добавляет дополнительный вес для каждой входной функции, позволяя сети узнать важность каждой из них. Все регулярные свертки также заменяются менее дорогими свертками, отделимыми по глубине.

Блок детектора состоит из слоя генерации опорных областей, слоя локализации (предсказывающего координаты смещения относительно центра сгенерированного прямоугольника, а также смещение по длине и ширине) и слоя, на выходе которого определяется вероятность того, что в исходном прямоугольнике содержится объект каждого из классов.

Количество прямоугольников, формируемых слоем генерации опорных областей, зависит от размерности карты признаков блока, на основе которого

строится детектор. Такие прямоугольники покрывают все изображение сеткой, как показано на рисунке 19. Количество генерируемых ограничивающих рамок для одного якоря составляет 9 штук. Ширина, высота и масштаб рамок определяется константами.

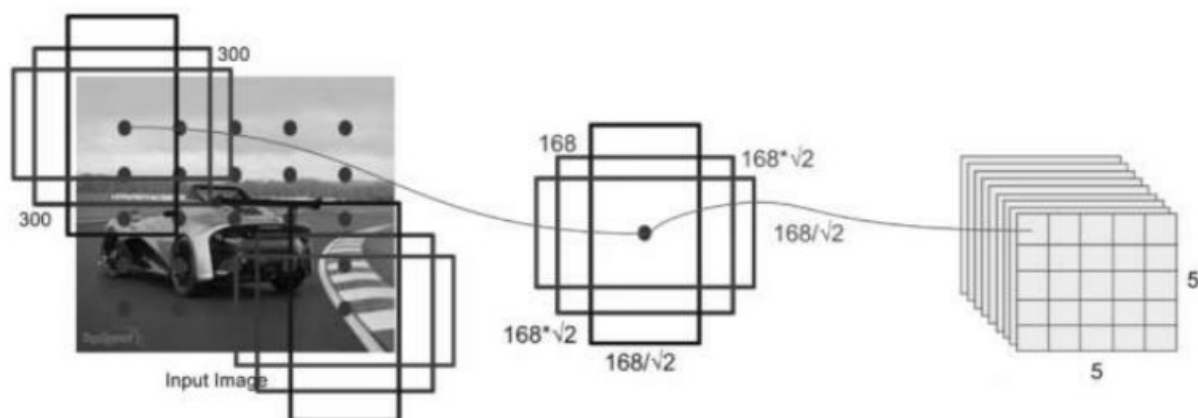


Рисунок 19 – Расположение якорных прямоугольников на изображении.

Для генерации окончательного обнаружения объектов мозаичные опорные области, принадлежащие фоновому классу, удаляются, а оставшиеся фильтруются по их оценке достоверности. Опорные области (рамки) с наибольшей степенью достоверности выбираются с использованием немаксимального подавления (NMS).

2.3.2 Алгоритм кластеризации прогнозов

Интенсивность истечения дыма зависит от нескольких факторов, например, длительности и количества вещества, подвергшегося горению. В результате может возникнуть ситуация, связанная с отсутствием дыма на изображениях в разные моменты времени, поэтому для минимизации ошибки необнаружения очага возгорания, на вход системы подается серия изображений, соответствующих началу, середине и концу видеоряда. Входное изображение преобразуется к разрешению 512×512 пикселей и имеет выделенные участки, полученные с помощью алгоритма выделения динамических признаков. На выходе системы формируются предсказания объектов: класс, вероятность обнаружения и

координаты объектов. В итоге имеется несколько объектов обнаружения с разной вероятностью и разными координатами. Необходимо объединить детекции и усреднить значения вероятностей, если они относятся к одному объекту.

На вход нейронной сети поступают 5 кадров одного ракурса, но разнесенные во временном промежутке. Так как дым имеет весьма нестабильную структуру (плотность, изменчивость формы, направление движения), то на каждом кадре имеем отличную от других кадров уникальную форму дымового облака. Таким образом, алгоритм в большинстве случаев выделяет наиболее характерные участки дыма в данный момент времени, что отчетливо прослеживается на рисунке 22. При таком подходе возможны случаи, когда на результирующем кадре может быть до 25 ограничивающих рамок, которые с разной степенью уверенности выделяют один и тот же объект. Эти боксы группируются в кластеры по признаку процента перекрытия при наличии в кластере 2 и более боксов. На вход алгоритма кластеризации прогнозов подается список всех обнаруженных ограничивающих рамок (боксов) на кадре, а также их вероятности принадлежности к классу. Блок-схема алгоритма кластеризации прогнозов обнаруженных объектов представлена на рисунке и состоит из следующих этапов:

1. Фильтрация данных. Проверка на условие прохождения минимального порога по вероятности ограничивающей рамки. Итоговый список содержит в себе все обнаруженные объекты, отсортированные в порядке уменьшения степени вероятности принадлежности к классу.
2. Расчет пересечения боксов (см. рисунок 20) по формуле:

$$\Delta S = [\min(p_{x_2}, p_{x_4}) - \max(p_{x_1}, p_{x_3})] \cdot [\min(p_{y_2}, p_{y_4}) - \max(p_{y_1}, p_{y_3})], \quad (2.6)$$

где ΔS – площадь пересечения боксов, $p_{(x,y)_1}$ – координата нижнего левого угла первого прямоугольника, $p_{(x,y)_2}$ – координата верхнего правого угла первого прямоугольника, $p_{(x,y)_3}$ – координата нижнего левого угла второго

прямоугольника, $p_{(X,Y)_4}$ – координата верхнего правого угла второго прямоугольника.

3. Формирование $n \times n$ матрицы $A = [a_{ij}]$, где n – количество боксов, которые остались после фильтрации. Элементы матрицы (см. рисунок 21) заполняются по формуле:

$$a_{ij} = \begin{cases} \frac{\Delta S}{S_i} \cdot 100, & \text{если } i > j \\ -2, & \text{если } i = j \\ \frac{\Delta S}{S_j} \cdot 100, & \text{если } i < j \end{cases}, \quad (2.7)$$

где S_i – площадь i -го прямоугольника, S_j – площадь j -го прямоугольника.

4. Перебор столбцов матрицы A с целью заполнения главной диагонали номерами кластеров боксов этапы 5-7 пока не будут переопределены все элементы главной диагонали.

5. Расчет количества пересечений для текущего столбца матрицы по формуле:

$$k = \begin{cases} k + 1, & \text{если } a_{ij} \geq \varepsilon \\ 0, & \text{если нет} \end{cases}, \quad (2.8)$$

где k – это количество элементов матрицы, удовлетворяющих условию образования кластера, ε – задаваемое значение минимального пересечения боксов.

6. Вычисление номера кластера для текущего столбца по формуле:

$$a_{ii} = \begin{cases} y, & \text{если } k \geq ms \\ -1, & \text{если нет} \end{cases}, \quad (2.9)$$

где y – номер кластера, ms – минимальное необходимое количество пересекающихся боксов

7. Рекурсивно каждому элементу a_{ij} найденного кластера y присваивается номер текущего кластера y

8. Расчет центра масс верхнего и нижнего углов ограничивающей рамки для каждого кластера по формуле:

$$P_{(X,Y)_c} = \frac{\sum_i P_{(X,Y)_i} P_i}{\sum_i P_i}, \quad (2.10)$$

где $P_{(X,Y)_c}$ – координата нового центра угла, $P_{(X,Y)_i}$ – координата угла обнаруженного объекта, P_i – вероятность обнаруженного объекта.

9. Вычисление новой вероятности P_c прогнозируемых боксов в зависимости от количества обнаруженных кластеров по формуле:

$$P_c = \frac{\sum_i P_i^2}{\sum_i P_i}. \quad (2.11)$$

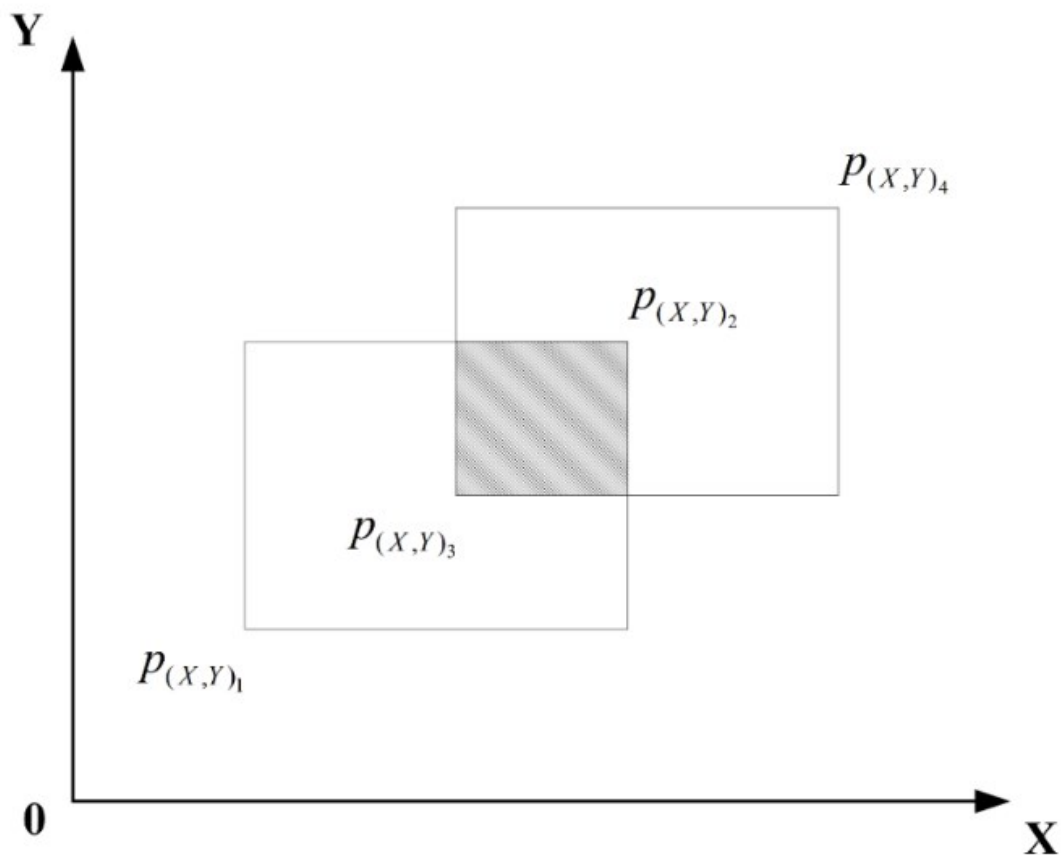


Рисунок 20 – Пример расчета пересечения двух боксов

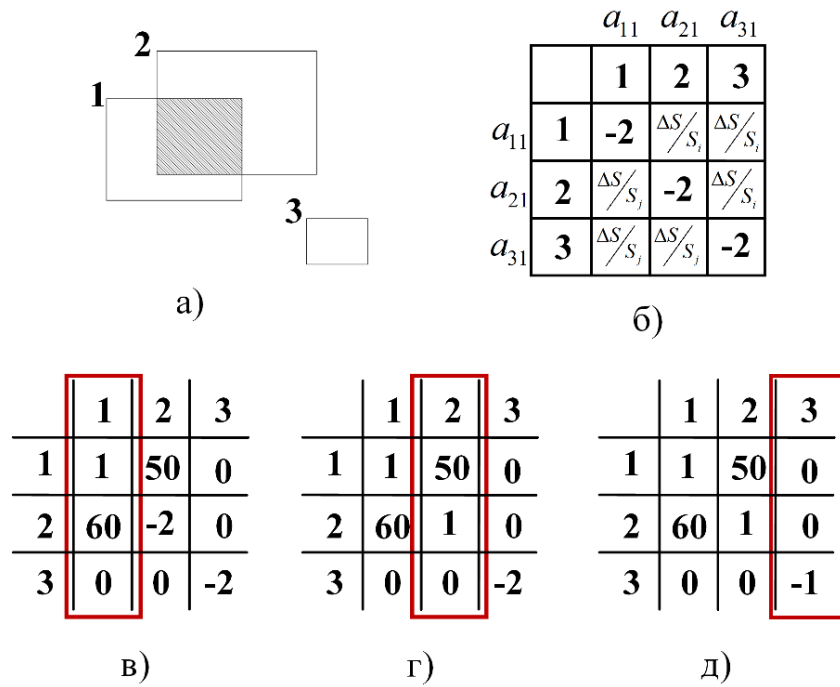


Рисунок 21 – Пример реализации алгоритма кластеризации площадей.

а) группа найденных боксов; б) матрица пересечений (этап 2); в) вычисление номера кластера (этап 5); г) вызов функции формирования кластеров для второго элемента (этап 6); д) вычисление номера кластера для изолированного бокса (этап 5);

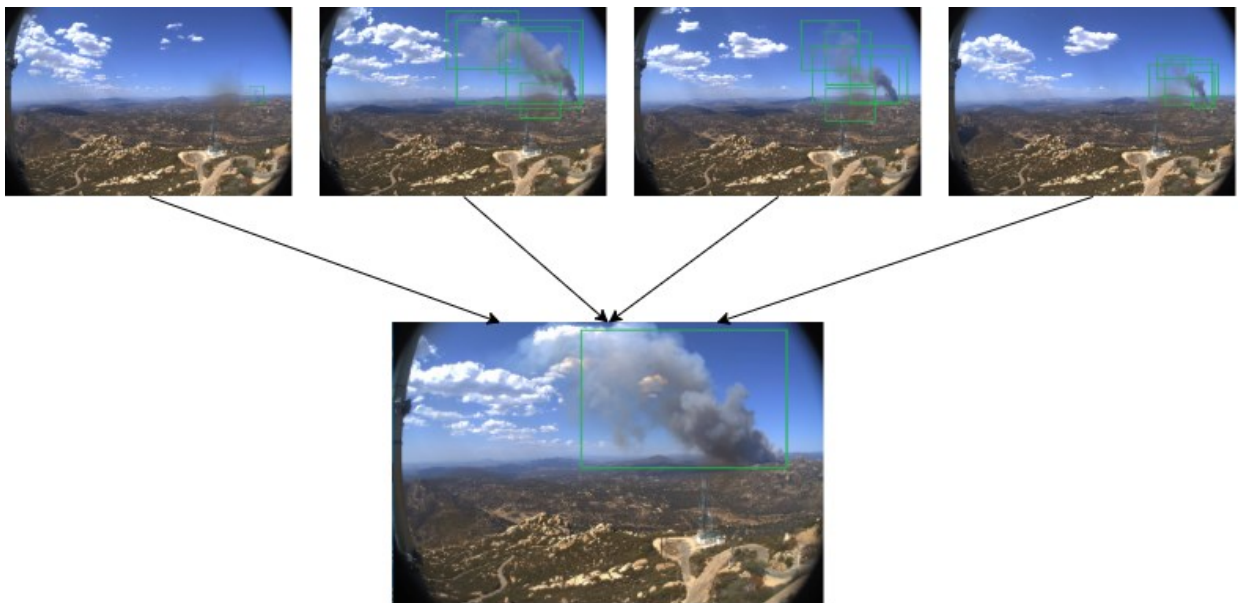


Рисунок 22 – Алгоритм объединения детекций

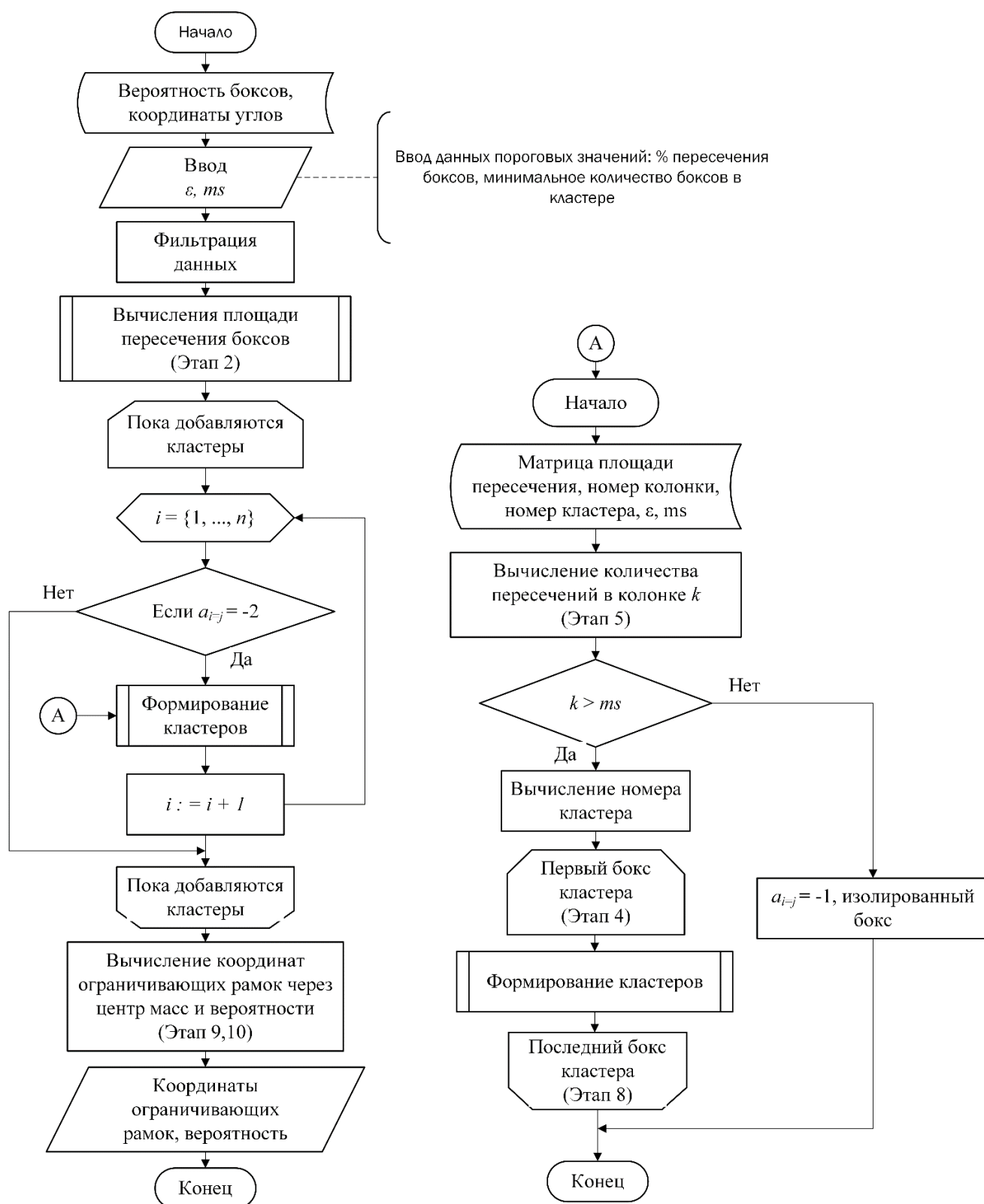


Рисунок 23 – Блок- схема алгоритма кластеризации прогнозов

Отличительной особенностью от существующих алгоритмов кластеризации является правило расчета метрики расстояния между точками. Под расстоянием понимается величина, равная отношению площади i -го и j -го боксов к площади

пересечения боксов (формула 2.6), таким образом, формируется ассиметричная матрица «расстояний». Реализация алгоритма в отличие от существующих позволяет решить проблему кластеризации для ассиметричной матрицы «расстояний».

2.4. Разработка алгоритма фильтрации обнаруженных объектов

2.4.1 Подготовка изображений

На видео, записанном камерой наблюдения, видны различные типы облаков и тумана, которые очень похожи на дым от лесного пожара. Следовательно, чтобы отличить их от настоящего дыма, необходимо учитывать временное движение дыма, которое изменяется под действием ветра или воспламеняющегося материала. С этой целью сначала выделяются участки изображения с высокой вероятностью обнаружения дыма с использованием детектора EfficienDet. Используемые кадры для выделения областей-кандидатов были предварительно обработаны согласно алгоритму, описанному в пункте 2.2. В результате применения алгоритма предварительной обработки была утрачена часть характеристических признаков. Поэтому координаты области-кандидата наносятся на первый кадр в исходной видеопоследовательности. Затем строится новая последовательность с использованием кадров предыдущих 10 секунд в той же позиции, что и соответствующая область-кандидат, как показано на рисунке 24. Последовательность с предполагаемой областью дыма от лесного пожара проверяется с помощью гибридной нейронной сети на основе СНС и РНС для извлечения пространственно-временных характеристик с последующей классификации объекта.



Рисунок 24 – Формирование серии изображений для анализа гибридной нейронной сети

На начальных этапах развития пожара движение дыма слишком медленное, а анализ всех кадров видеозаписи занимает существенные вычислительные ресурсы. Поэтому, чтобы сократить время обработки и увеличить динамику дыма, мы извлекаем только 4-5 кадров в секунду (fps) вместо того, чтобы использовать все кадры для последовательности. После того как последовательность построена, все кадры нормализуются до размера 299×299 пикселей, который является входным размером для анализа СНС.

2.4.2 Алгоритм аугментации данных

Для решения задач глубокого обучения требуется большое количество изображений для обучения. Одним из вариантов увеличения данных при их недостатке является аугментация. Процесс аугментации – это метод, который искусственно увеличивает размер обучающего набора данных путем создания модифицированных версий изображений в существующем наборе данных. Обучение моделей нейронных сетей на большем количестве данных может привести к созданию более совершенных моделей, а методы дополнения могут создавать варианты изображений, которые могут улучшить способность подходящих моделей обобщать то, что они узнали, на новые изображения. Преобразования включают в себя ряд операций из области обработки изображений, таких как сдвиги, перевороты, масштабирование, изменение яркости и контрастности и т.д. Необходимо отметить, что использование аугментации является удобным инструментом для увеличения данных для CNN, но совершенно не подходит для RNN. Поэтому предлагается авторская интерпретация данного метода увеличения данных.

С целью оптимального расширения выборки алгоритм аугментации осуществляется итеративно, когда в различные виды искажений постепенно добавляются в обучающую выборку. Для получения большей обобщающей способности модели в результате обучения принято решение применять алгоритм аугментации не на этапе подготовки данных к обучению, а в процессе подготовки batch-a. Также к отличительной особенности алгоритма аугментации от классической относится функциональная возможность применения идентичных изменений к N количеству подаваемых изображений. Тем самым структура данных (временная последовательность) не будет нарушать геометрические и качественные особенности экспозиции изображения. Алгоритм создания расширенной выборки изображен на рисунке 25.

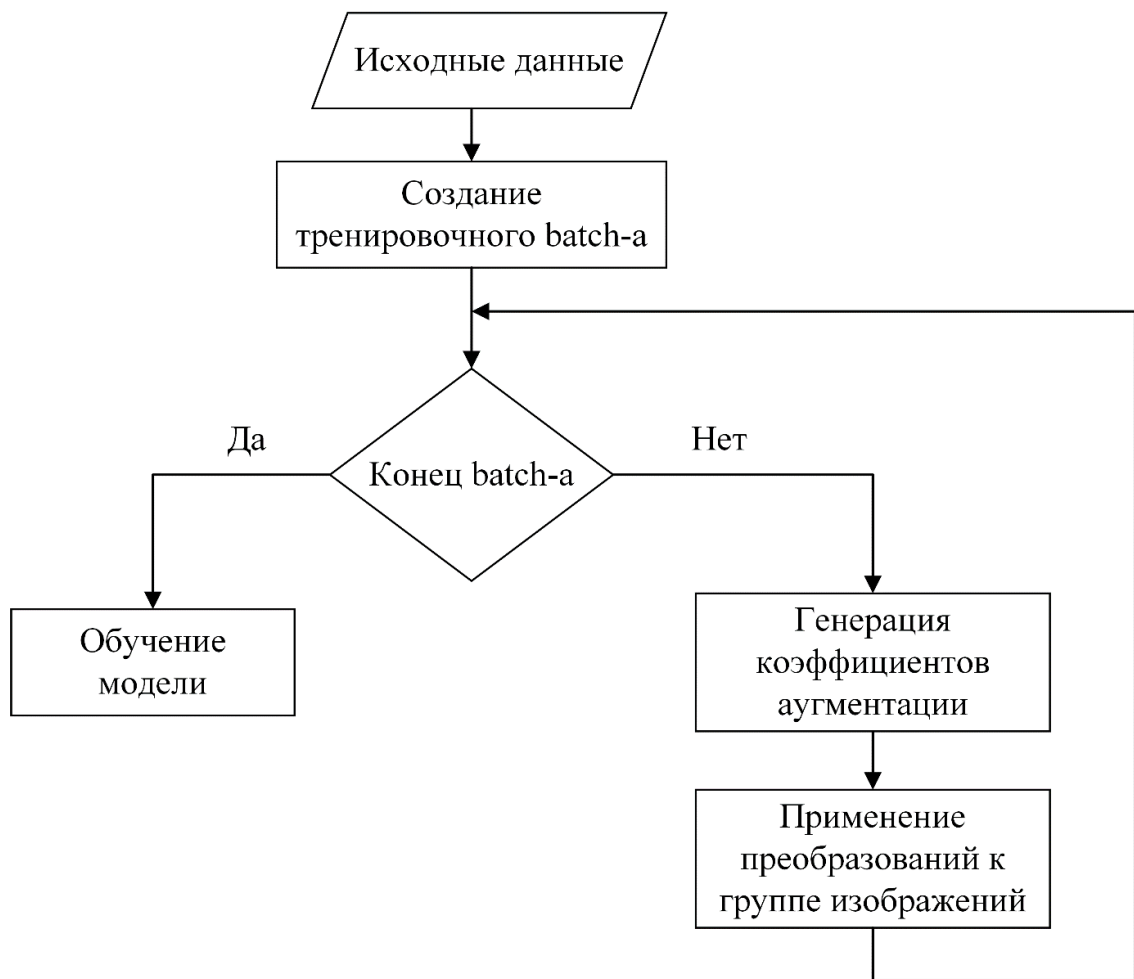


Рисунок 25 – Блок -схема алгоритма аугментации данных

В качестве преобразований выбраны следующие операции:

- Случайное приближение;
- Отражение по горизонтали;
- Случайный поворот;
- Случайное изменение яркости;
- Случайное изменение насыщенности;
- Случайное изменение контрастности;
- Случайный шум.

2.4.3 Гибридная архитектура нейронной сети для классификации объектов

Изучаемые видеопотоки содержат различные типы объектов, в частности облака и туманность, которые очень похожи по текстуре на дым от лесного пожара. Несмотря на то, что современные CNN используют глубокие уровни свертки, они не ищут новые функции во временной области. Важно помнить, что дым имеет динамическую природу и меняет свою форму с течением времени. Поэтому, чтобы правильно классифицировать обнаруженные объекты, необходимо учитывать временное изменение дымового облака. Исходя из этого, разработана гибридная архитектура нейронной сети, представленная на рисунке 26 с указанием параметров слоев в таблице 1, включающей сверточную (СНС) и рекуррентную (РНС) нейронные сети. Ключевой особенностью предложенной архитектуры является извлечение временной корреляции между изображениями путем сохранения памяти о прошлых изображениях.

Таблица 1 Параметры гибридной архитектуры нейронной сети (выделенные значения, исследуемые параметры)

Тип	Параметры		Характеристика
Слой TimeDistributed (Модель СНС Inception-V3)	Вход	4(300x300x3)	4 изображения
	Выход	4x2048	4 шага на вектор признаков
Слой РНС (LSTM)	Вход	4x2048	4 шага на вектор признаков
	Выход	1024	Нейронов
Слой Dense с функцией активации LeakyRelu	Вход	1024	Нейронов
	Выход	256	
Слой Dropout	Вход	256	Коэффициент 0,5
	Выход	256	
Слоя Dense с функцией активации Sigmoid	Вход	256	Нейронов
	Выход	1	Нейрон с выводом значения степени уверенности

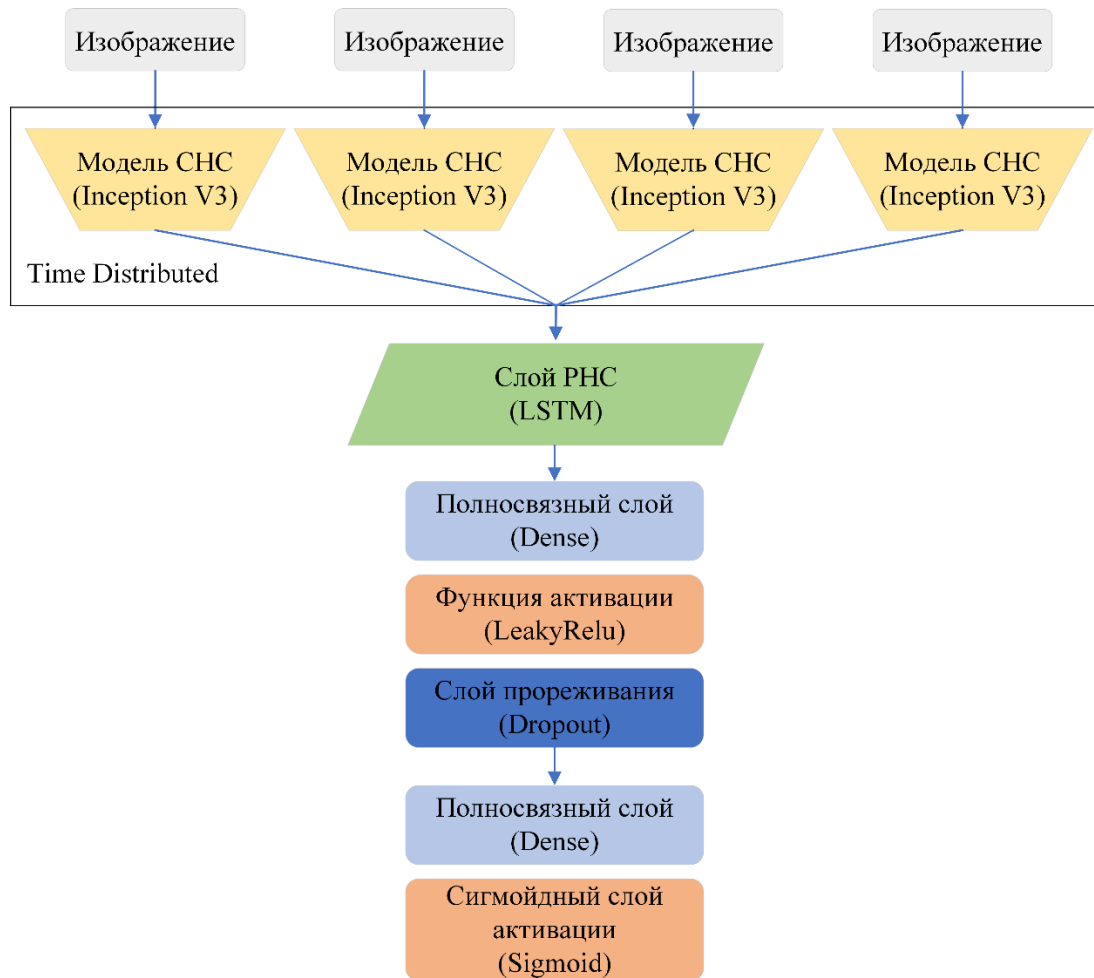


Рисунок 26 – Архитектура комбинированной модели нейронной сети CNN-RNN

Принцип работы заключается в следующем: кадры видеозаписи передаются модели CNN для извлечения признаков низкого уровня. В работе использована Inception V3, предварительно обученная на датасете ImageNet, в качестве базовой модели CNN. Нейронная сеть Inception-V3 [114], архитектура которой представлена на рисунке **Ошибка! Источник ссылки не найден.**, является модификацией Inception-V1,V2[85].

Основными модификациями модели Inception-V3 являются:

1. Факторизация в меньшие свертки
2. Пространственная факторизация в асимметричные свертки
3. Внедрение вспомогательных классификаторов
4. Уменьшение размера сети.

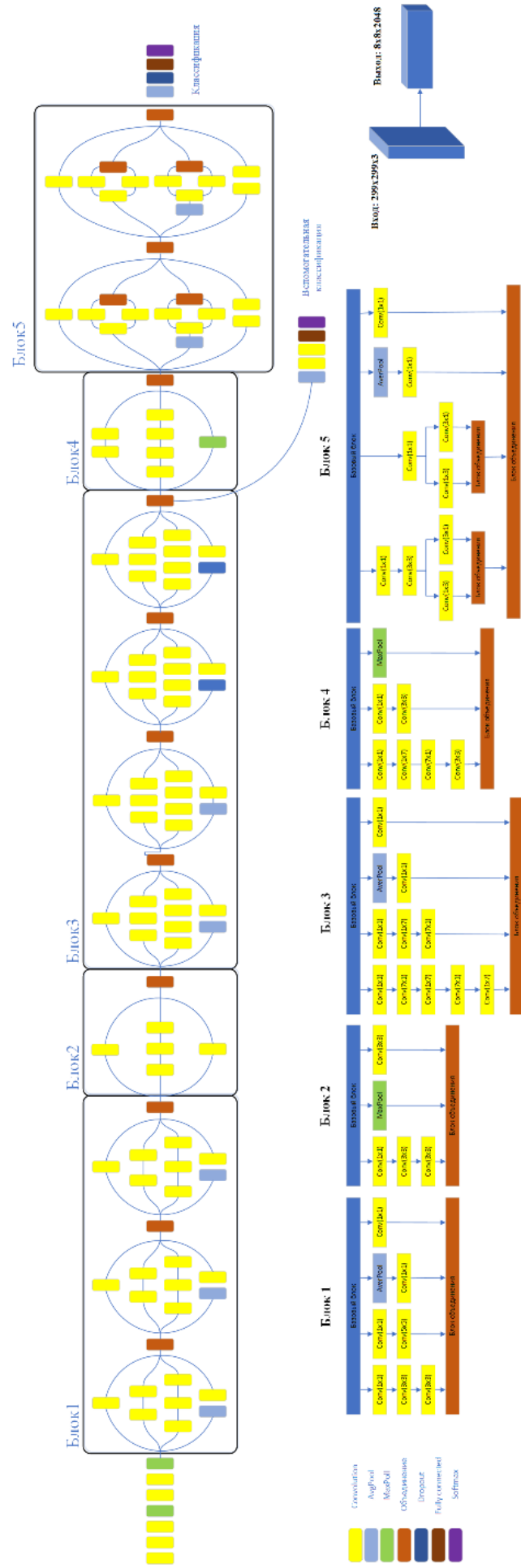


Рисунок 27 – Архитектура сети Inception-V3

Для формирования последовательности из векторов признаков на вход рекуррентной части сети подается результат конкатенации признаков от слоя TimeDistributed.

Блок RNN сети состоит из входного слоя, слоев LSTM, полносвязного слоя и слоя Dropout. Слой LSTM состоит из блоков, в которых есть ячейка памяти для хранения прошлой информации. Ячейка памяти имеет некоторые функции ворот, которые позволяют забыть прошлую информацию, зависящую от времени. На рисунке 28 представлена структура слоя LSTM. На вход x_t^m поступает вектор, соответствующий кадру m в момент времени t , h_t^m – выходной вектор модуля(ворот) LSTM в момент времени T . W_f, W_i, W_c и W_o – веса соответствующих ворот, b_f, b_i, b_c, b_o – смещение весов. H_t^m - набор выходных данных слоя LSTM $\{h_t^m(1), h_t^m(2), \dots\}$. Блок LSTM имеет вектор ворот или вентиля забывания f_t (вес «старой информации»), вектор входного вентиля или ворота i_t (вес получения новой информации), вектор с кандидатами \tilde{C}_t и вектор выходного вентиля или ворот o_t . Веса рассчитываются по следующим уравнениям:

$$f_t = \sigma(W_f \cdot [h_{t-1}^m, x_t^m] + b_f) \quad (2.12)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}^m, x_t^m] + b_i) \quad (2.13)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}^m, x_t^m] + b_c) \quad (2.14)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (2.15)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}^m, x_t^m] + b_o) \quad (2.16)$$

$$h_t^m = o_t \cdot \tanh(C_t) \quad (2.17)$$

Выходы модулей LSTM передаются на полносвязный слой с добавлением функции активации LeakyReLu и слоя Dropout. Последний слой сети представлен полносвязным слоем с одним выходом и сигмоидной функцией активации для вывода вероятности классификации обнаруженных объектов.

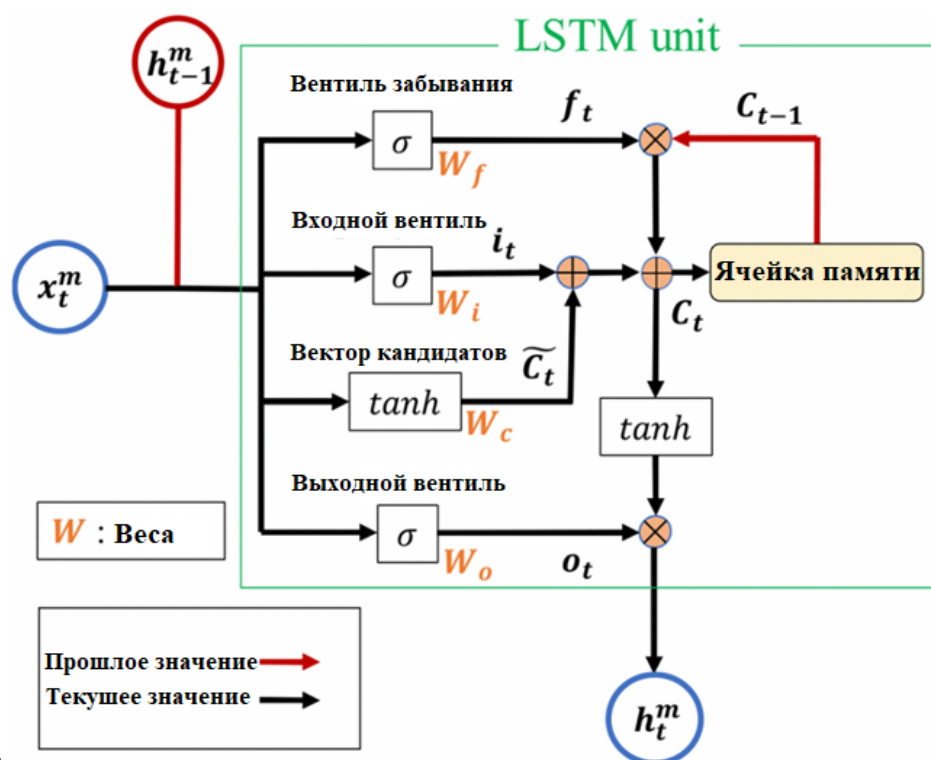


Рисунок 28 – Структура слоя LSTM

2.5. Состав комплекса программных средств

Разработанные алгоритмы внедрены в программно-аппаратный комплекс, реализующий захват видеопотока с цифровой камеры и запуск библиотеки для анализа данных. В библиотеке реализованы следующие функции: предобработка данных, запуск нейронных сетей для трекинга объектов, постобработка результатов трекинга, обрезка кадров по результатам постобработки, формирование серии изображений, фильтрация обнаруженных объектов, визуализация результатов. Библиотека имеет пользовательский интерфейс, представленный на рисунке 29. Интерфейс предназначен для корректировки параметров алгоритмов предобработки и постобработки пользователем. Библиотека написана на языках программирования C и C++.

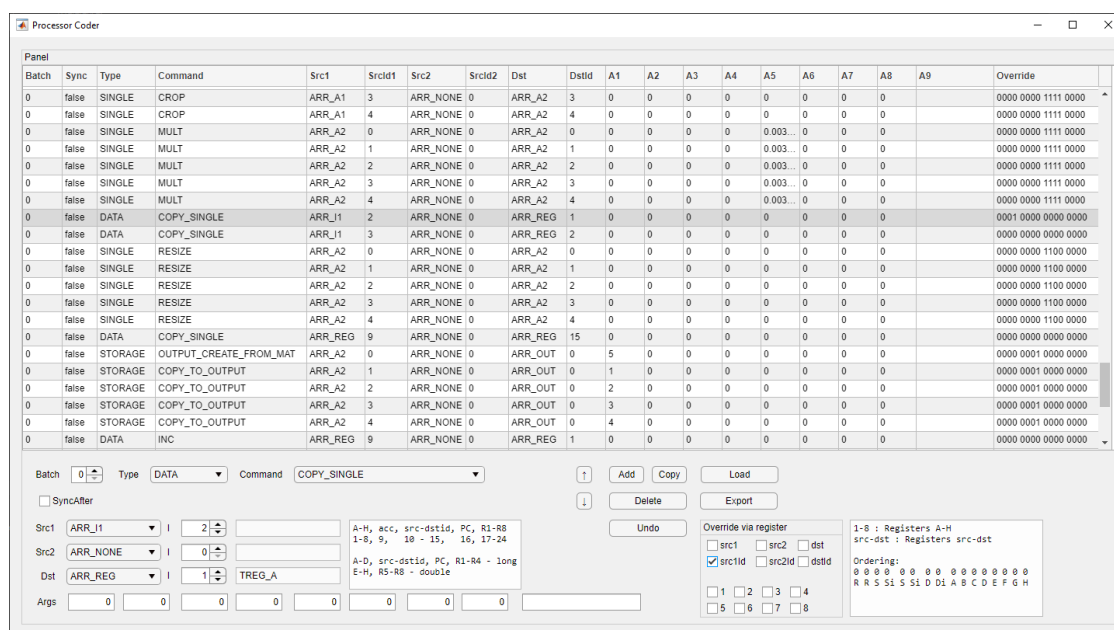


Рисунок 29 – Пользовательский интерфейс для корректировки параметров алгоритмов предобработки и постобработки пользователем

На сегодняшний день разработчики систем с искусственным интеллектом располагают возможностями использовать фреймворки глубокого машинного обучения, значительно сокращающие время разработки. Это TensorFlow [115], PyTorch [116]. Самым поддерживаемым открытым программным средством является открытая библиотека машинного обучения TensorFlow, но Pythorch является не менее перспективным и набирает все большую популярность за счет своей гибкости.

В данной работе был использован открытый фреймворк глубокого машинного обучения Tensorflow. Данный фреймворк не только позволяет самостоятельно построить архитектуру ИНС и провести тонкую настройку ее параметров, но и предлагает в качестве основы сети выделения признаков использовать предварительно обученные энкодеры. Также стоит отметить, что данный фреймворк имеет дополнительную надстройку в виде TensorFlow API Object Detection [117], используемую в данной работе. Данная надстройка позволяет без особого труда настроить модель по обнаружению объектов и провести ее обучение. Для данной надстройки предлагается большой выбор архитектур ИНС, обученных на таких больших множествах данных как ImageNet,

CoCo. Единственным недостатком данного Api является отсутствие возможности изменения выбранной архитектуры ИНС, но предоставляется возможность проведения тонкой настройки параметров сети. К положительным особенностям данного фреймворка можно также отнести возможность взаимодействия с ним на различных вычислительных ресурсах (CPU|GPU), а также возможность сохранения обученной модели (замороженный граф весов) в различных форматах, таких как h5, skpt, pb.

Для тестирования и визуализации работы был разработан интерфейс пользователя, представленный на рисунке 30. Интерфейс программы написан на языке Python с использованием веб-фреймворка для создания дэшбордов Dash-Python. Разработанный интерфейс позволяет пользователю загрузить видеофрагмент на одностраничный веб-ресурс и выполнить его анализ, задействовав соответствующую кнопку, в результате чего в случае обнаружения будет получено изображение с отображением соответствующей информации.

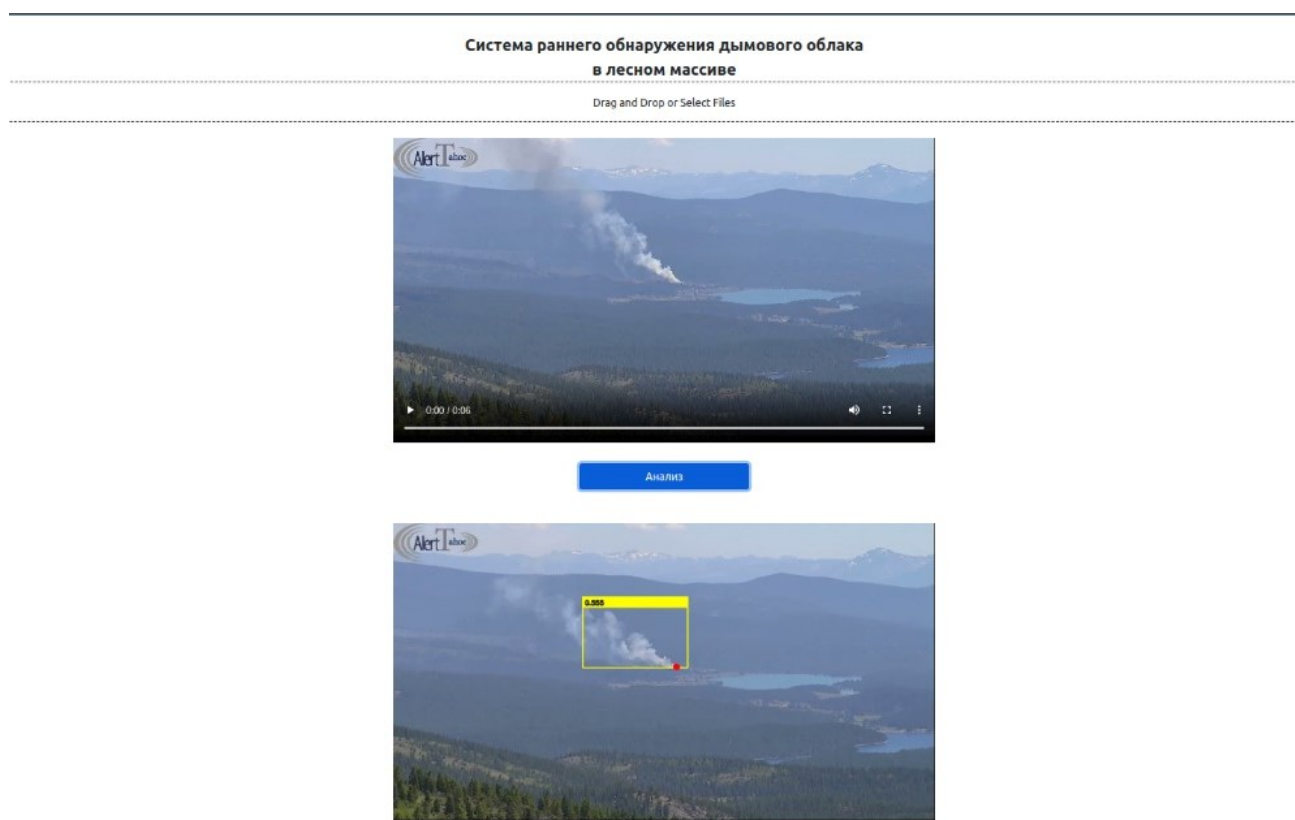


Рисунок 30 – Интерфейс пользователя для визуализации работы системы

Еще одним важным инструментом разрабатываемой системы является платформа визуализации процесса обучения Weights&Biases [118]. Данный сервис позволяет не только сохранять тренировочные и валидационные метрики, но также веса модели, ее граф и результаты предсказания различного характера (изображения, матрицы чисел и т. п.). Также стоит отметить, что данный сервис облегчает произведение тонкой настройки модели благодаря возможности построения многомерных графиков.

2.6. Выводы по второй главе

1. В связи с большим количеством ошибок первого рода при использовании одноэтапных методов обнаружения объектов была предложена технология выделения динамических объектов в видеоряде. В основе технологии лежит алгоритм обнаружения объектов с динамическими признаками, сочетающий в себе этапы предподготовки изображений, обнаружения объектов, объединения обнаруженных объектов с разных кадров видеозаписи в итоговое решение на результирующем кадре посредством работы алгоритма кластеризации прогнозов и классификации последовательности фрагментов изображения с обнаруженными объектами.
2. Разработан алгоритм кластеризации прогнозов. Отличие разработанного алгоритма от существующих, заключается в авторском подходе к задаче кластеризации ограничивающих рамок объектов на изображении и фильтрации вершин углов ограничивающих рамок обнаруженных объектов с последующим объединением полученных координат на основе уравнения центра масс.
3. Для более эффективного обучения модели классификации предсказанных областей предложено использовать алгоритмы динамического размера батча и аугментации данных (расширение выборки/набора данных в процессе обучения). В целях эффективного расширения выборки, алгоритм аугментации выполняется итеративно, учитывая такой параметр как количество изображений в батче. Таким образом, структура данных (временная последовательность) не нарушает геометрических и качественных

особенностей изображения. Для получения большей обобщающей способности модели в результате обучения принято решение применять алгоритм аугментации не на этапе подготовки данных к обучению, а в процессе подготовки батча.

4. Разработан алгоритм фильтрации предсказанных областей-кандидатов. В основе алгоритма лежит гибридная архитектура НС, сочетающая в себе модели сверточной и рекуррентной нейронных сетей. Для извлечения признаков низкого уровня используется сверточная нейросеть Inception-V3. С целью формирования последовательности признаков используется слой TimeDistributed. Полученная последовательность признаков анализируется рекуррентной нейронной сетью со слоем LSTM, что позволяет извлечь признаки, отражающие хронологию изменения объектов. Анализ хронологии изменения объектов позволяет повысить точность обнаружения путем снижения ошибок первого рода.

ГЛАВА 3

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ РАЗРАБОТАННЫХ АЛГОРИТМОВ

В этой главе представлены вычислительные эксперименты с целью оценки результатов обнаружения объектов, полученных на основе предлагаемых алгоритмов. Сравнение полученных результатов с аналогичными системами обнаружения, чьи характеристики представлены в открытом доступе, показало, что предложенная технология выделения объектов с динамическими признаками демонстрирует более высокую точность обнаружения и меньшее количество ложных срабатываний для задачи обнаружения лесных пожаров. Приведены примеры применения предложенных алгоритмов в других областях деятельности, что дает возможность говорить о свободной встраиваемости алгоритмов в системы в разных областях деятельности и их универсальности.

3.1. Подготовка эксперимента

Предложенная комплексная технология обнаружения пожароопасных объектов в лесном массиве, а также алгоритмы выделения динамических признаков, модель детектора, алгоритм объединения детекций и гибридная НС классификации объектов требуют экспериментальной проверки эффективности с точки зрения их применимости к задачам локализации и классификации, а также сравнения с существующими алгоритмами.

Для реализации эксперимента, подтверждающего эффективность разработанной технологии, необходимо осуществить выполнение алгоритмов на базах данных реальных видеозаписей и изображений. Предварительная обработка, обучение, тестирование и разметка данных осуществлялась на компьютере со следующими характеристиками: Intel Core i7-4820K CPU 3.7GHz, Nvidia GeForce GTX2080Ti, 32GB RAM и Windows10.

Исходной информацией для проведенных экспериментов являются данные, полученные из нескольких источников, в частности, данные открытых онлайн

ресурсов Nevada Seismological Laboratory при поддержке университета города Невады (США), [8],[9], Center for wildfire research основанного при поддержке университета города Сплит (Хорватия) [121], видеозаписи съемок лесных пожаров Пермского лесного хозяйства [122], видеозаписи базы данных Калифорнийского университета в Сан-Диего "HPWREN" [123]. С целью непосредственного обучения все данные были предварительно размечены с использованием средств web-сервиса "SuperVisely"[124]. Общее количество собранных видеозаписей составило 550 единиц, из них содержащих очаги возгорания – 350 единиц и 200 видеозаписи без пожаров. Для обучения из каждой видеозаписи взята последовательность из 7 кадров. Данное количество кадров получено экспериментально.

Эксперимент проведен в несколько этапов:

1. Формирование обучающих выборок:

- Последовательность изображений без цветовой обработки
- Последовательность изображений с алгоритмом выделения динамических признаков
- Последовательность изображений с цветовой обработкой (применения фильтра Гаусса)
- Последовательность изображений с цветовой обработкой (применение фильтра Гаусса и перевод в оттенки серого)

2. Оценка эффективности моделей детекторов, обученных на разных обучающих выборках

3. Оценка алгоритмов объединения детекций

4. Формирование обучающей выборки для обучения классификатора

5. Оценка работы классификатора на основе гибридной модели нейронной сети CNN-RNN

3.2. Тестирование программной реализации алгоритмов

В данном параграфе проводится сравнение эффективности алгоритмов подготовки данных для трекинга объектов на изображениях с лесными массивами, получаемых системой противопожарного мониторинга, с применением различных вариантов цветовой обработки и выделения динамических признаков на изображениях:

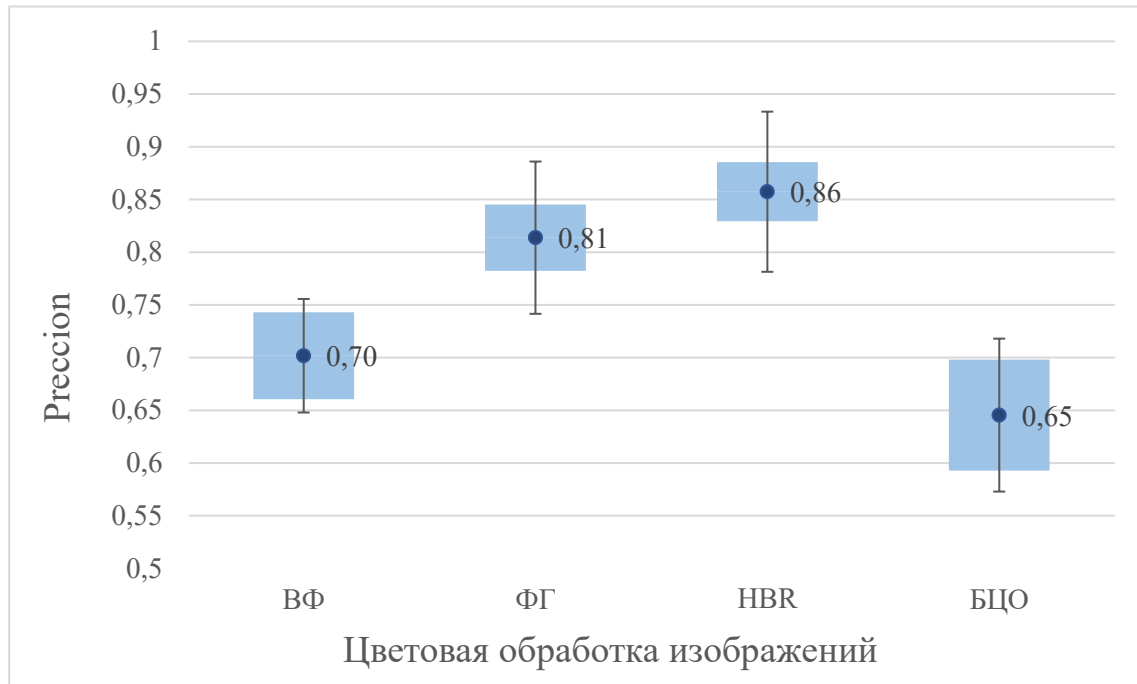
1. Однокомпонентный набор изображений без предварительной обработки в канале RGB, далее «БЦО».
2. Однокомпонентный алгоритм подготовки изображений с использованием алгоритма вычитания фона, описанного в пункте 2.1, далее «ВФ».
3. Двухкомпонентный алгоритм подготовки изображений с использованием авторского алгоритма вычитания фона и применения цветовой обработки фильтром Гаусса, далее «ФГ».
4. Трехкомпонентный алгоритм обработки изображений с использованием авторского алгоритма вычитания фона, фильтра Гаусса и цветовой обработки, а именно перевод из канала RGB в градации серого, далее «HBR».

Основная идея выполнения экспериментов заключалась в том, чтобы оценить влияние предварительной подготовки данных на точность и полноту обнаружений. Для выявления зависимостей было обучено 5 моделей нейронных сетей на всех типах данных.

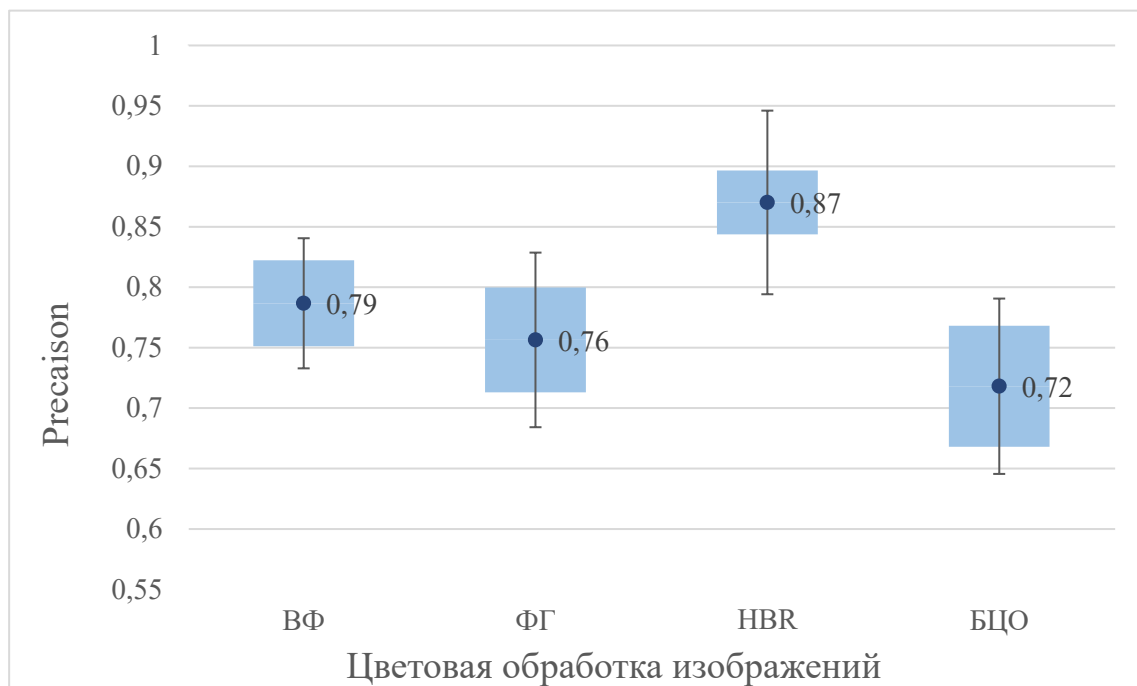
В целях первичной оценки значимости различных моделей использовался критерий Манна-Уитни. Данный метод использовался для оценки различий в группах. Нулевая гипотеза H_0 данного критерия указывает на то, что нет различий между результатами анализируемых кадров без обработки и алгоритмов предобработки изображений. Процедура состоит в упорядочивании значений в каждой строке и в суммировании полученных рангов; U -критерий вычисляется как сумма индикаторов попарного сравнения элементов выборки без обработки кадров с элементами выборки с обработкой кадров. В таком случае сравниваемая группа

относится к одной генеральной совокупности и соответственно их средние равны друг другу. Альтернативная гипотеза H_1 выдвигает предположение о том, что средние в исследуемой группе не равны.

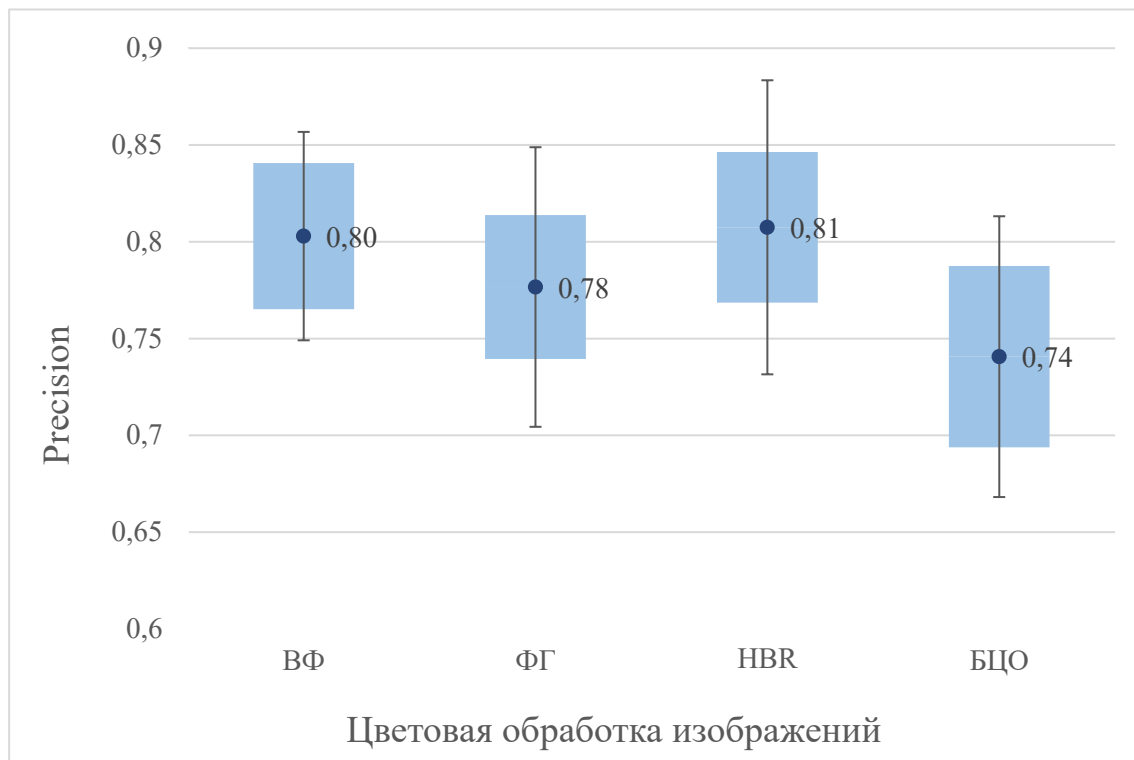
После проведения анализа были получены результаты, которые отражены в таблице 2 и на рисунках 31, 32. Как видно из полученных данных, для метрики Recall, для моделей EfficientDet-D0, EfficientDet-D1, SSD-MobilNet-V2 на трехкомпонентном наборе данных значение *p-value* меньше уровня значимости равное 0.5. Для модели EfficientDet-D2 есть различия между средними значениями для двухкомпонентного алгоритма обработки изображений. У модели FasterRCNN-ResNet50 имеются различия средних значений для трехкомпонентного алгоритма обработки изображений, при этом среднее значение выборки без обработки выше. Для метрики Precision наблюдаются значимые отличия для моделей EfficientDet-D0, EfficientDet-D1 на трехкомпонентном наборе данных, значение *p* меньше уровня значимости равное 0.5. Для модели EfficientDet-D2 нет различий между средними значениями между выборками с использованием алгоритмов обработки изображений и выборкой без обработки изображений. У модели Faster-Resnet50 имеются различия средних значений для однокомпонентного алгоритма обработки изображений. Для модели SSD-Mobilenet-V2 есть различия между средними значениями, при этом среднее значение выборки без обработки выше, чем значение выборки с трёхкомпонентным алгоритмом обработки изображений. Таким образом, на основании полученных результатов можно сделать вывод о том, что подготовка данных значительно влияет на качество обнаружения объектов. При этом для каждой модели НС положительное воздействие оказывает свой алгоритм предварительной обработки изображений.



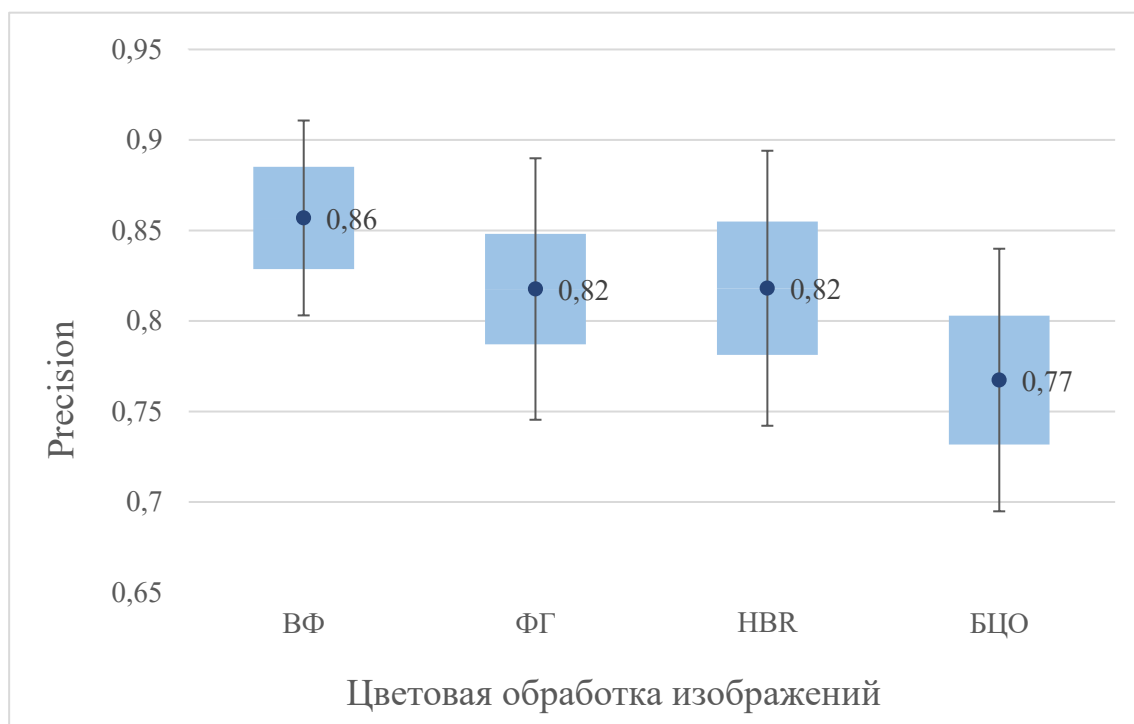
a) EfficientDet-D0



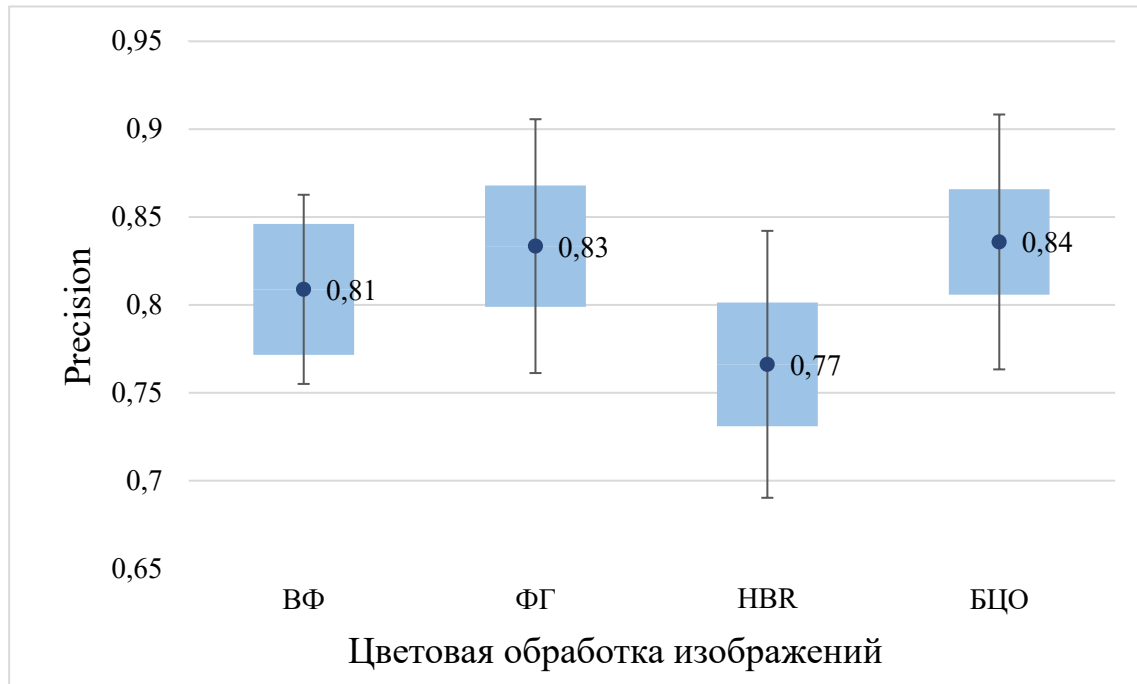
б) EfficientDet-D1



в) EfficientDet-D2

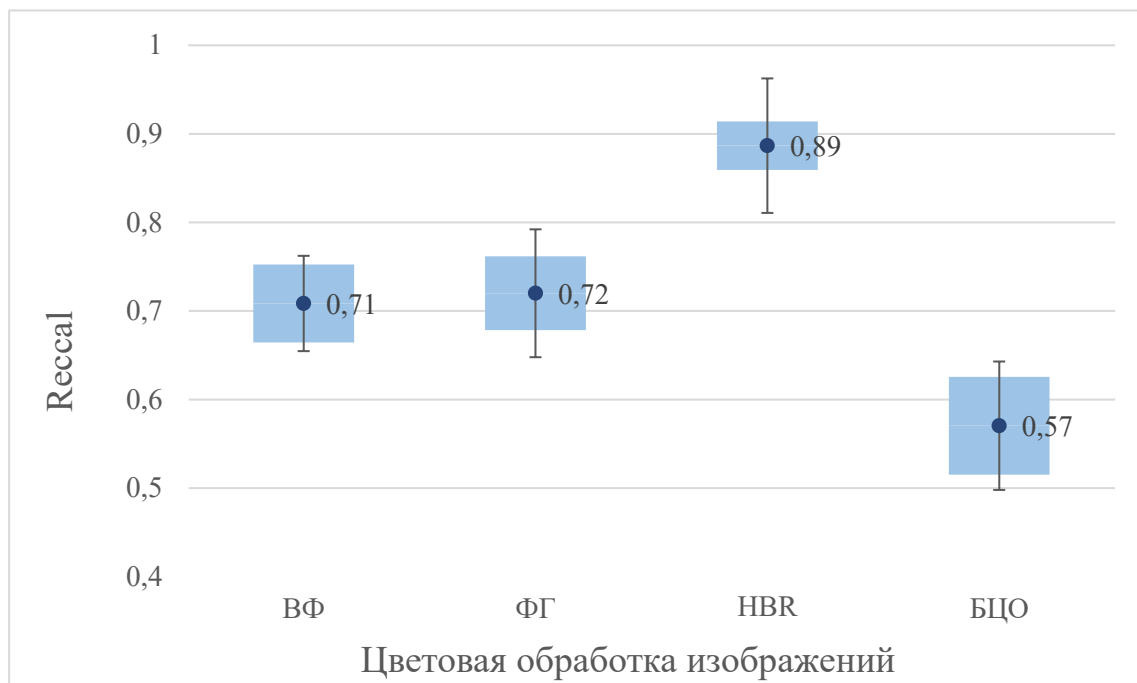


г) FasterRCNN-ResNet50

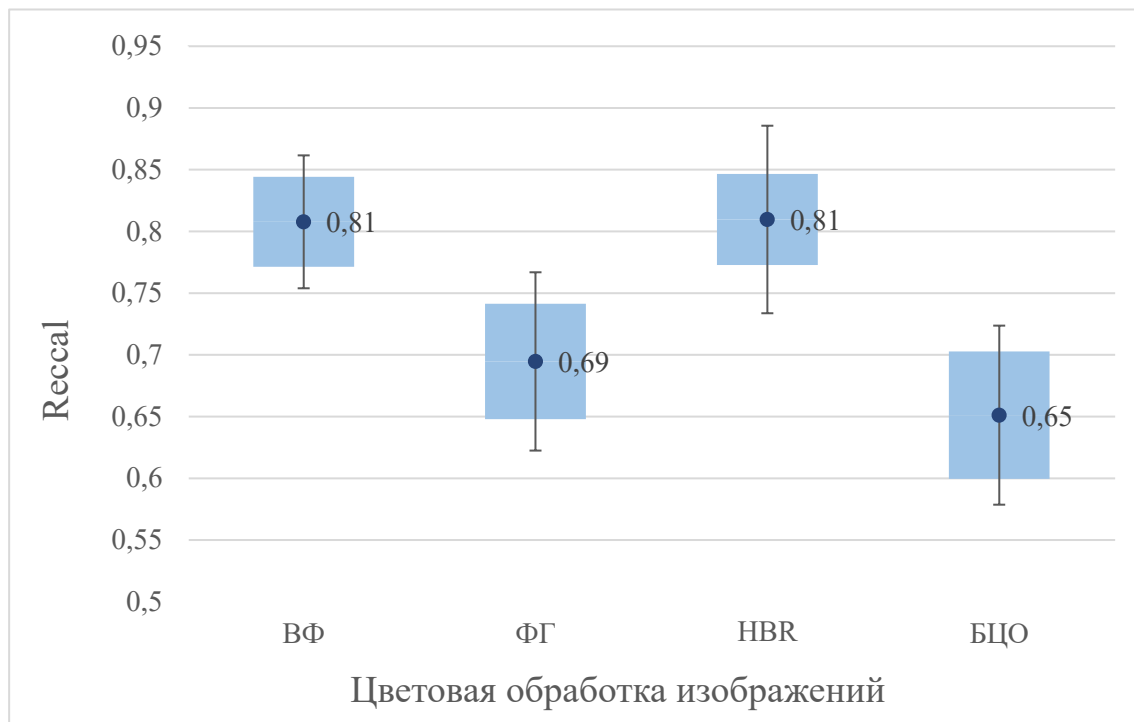


д) SSD-Mobilenet-V2

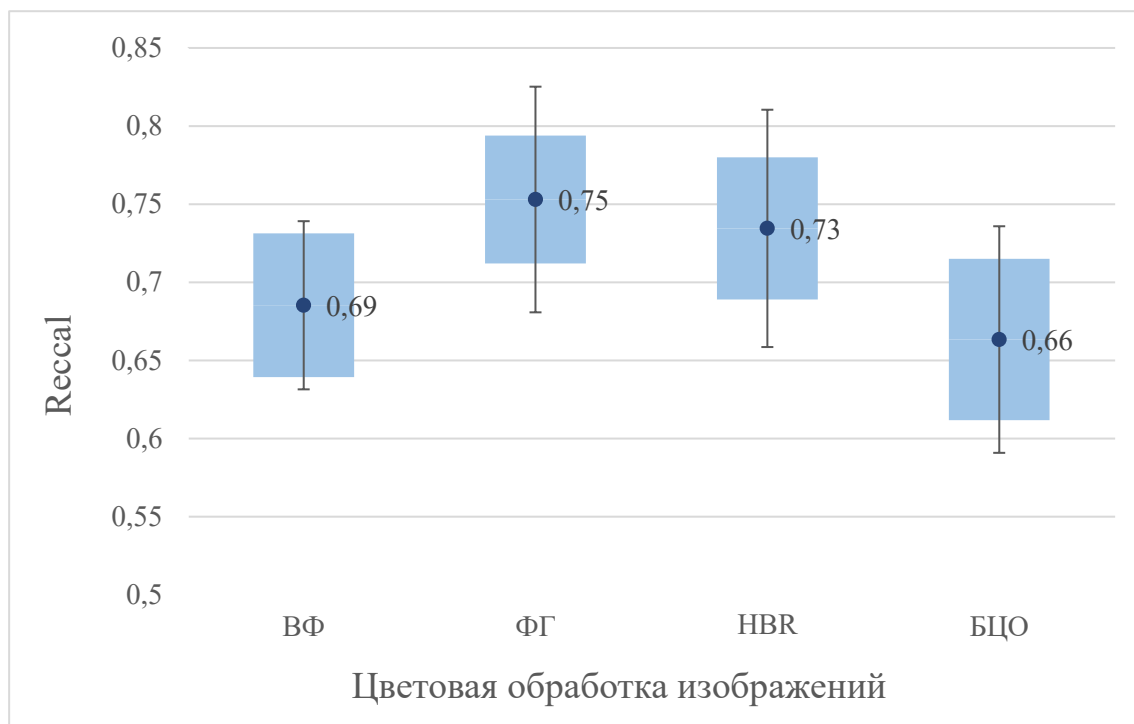
Рисунок 31 – График значимо различных параметров по метрике Precision для моделей НС.



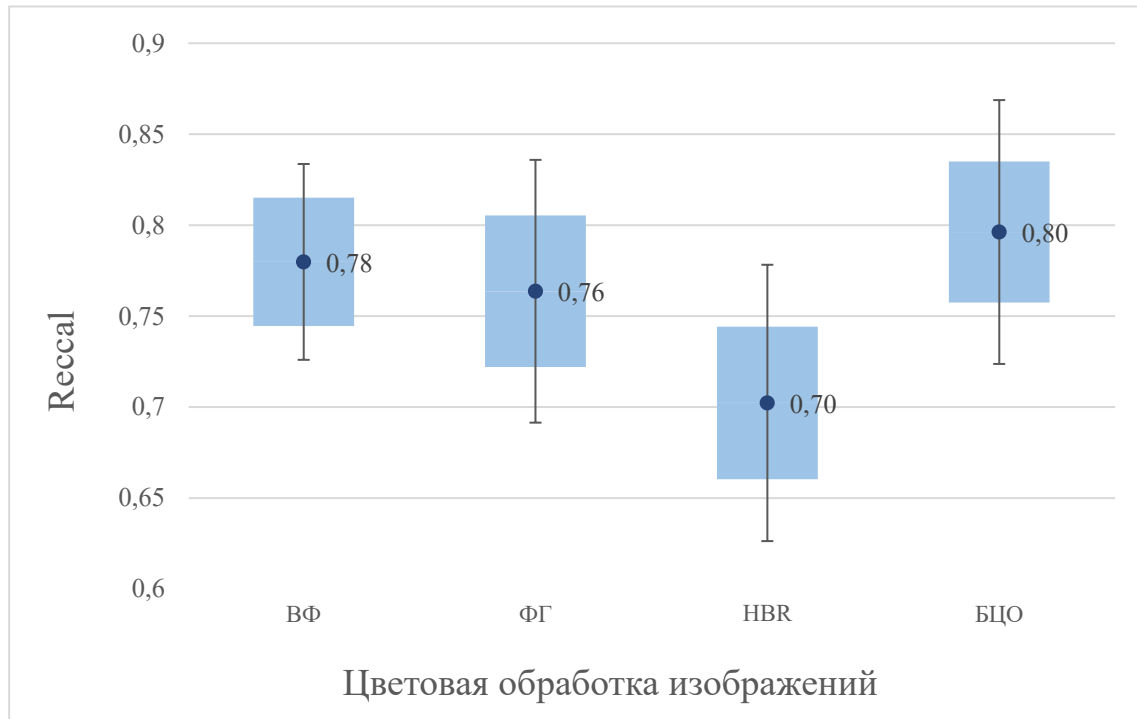
а) EfficientDet-D0



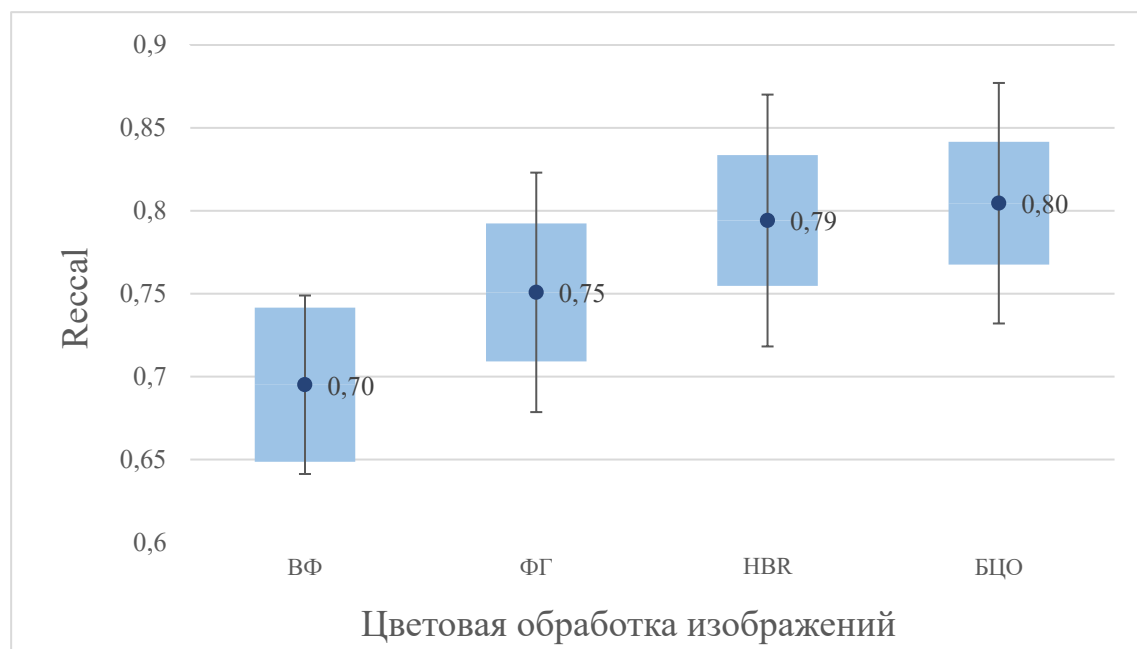
б) EfficientDet-D1



в) EfficientDet-D2



г) FasterRCNN-ResNet50



д) SSD-Mobilenet-V2

Рисунок 32 – График значимо различных параметров по метрике Recall для моделей НС.

Таблица 2 – Таблица оценки значимости различных алгоритмов предобработки данных по критерию Манна-Уитни

Метрика	Модель НС	ВФ		Гauss		HBR		БЦО
		p	\bar{x}	p	\bar{x}	p	\bar{x}	\bar{x}
Precision	EfficientDet-D0	0,67	0,70	0,16	0,81	0,09	0,85	0,64
	EfficientDet-D1	0,75	0,78	0,83	0,75	0,03	0,87	0,68
	EfficientDet-D2	0,36	0,80	0,15	0,77	0,63	0,80	0,72
	FasterRCNN-ResNet50	0,02	0,85	0,43	0,81	0,04	0,81	0,76
	SSD-Mobilenet-V2	0,81	0,8	0,46	0,83	0,018	0,76	0,81
Recall	EfficientDet-D0	0,23	0,70	0,11	0,72	0,0001	0,89	0,57
	EfficientDet-D1	0,03	0,80	0,11	0,69	0,031	0,80	0,56
	EfficientDet-D2	0,22	0,68	0,04	0,75	0,066	0,73	0,57
	FasterRCNN-ResNet50	0,57	0,77	0,85	0,76	0,039	0,70	0,79
	SSD-Mobilenet-V2	0,21	0,69	0,44	0,75	0,012	0,79	0,61
AP	EfficientDet-D0	0.67	0.69	0.16	0.74	0.09	0.88	0.64
	EfficientDet-D1	0.47	0.8	0.95	0.75	0.18	0.86	0.68
	EfficientDet-D2	0.92	0.76	0.78	0.76	0.42	0.84	0.71
	FasterRCNN-ResNet50	0.66	0.74	0.81	0.73	0.27	0.69	0.74
	SSD-Mobilenet-V2	0.22	0.8	0.36	0.79	0.38	0.78	0.85

Для сравнения результатов обнаружения использовались метрики полноты (*Recall*) и точности (*Precision*). На основе проведенного исследования выявлено,

что включение этапа предобработки при обнаружении объектов влияет на точность и полноту результатов вне зависимости от архитектуры нейронной сети. Реализованный алгоритм выделения динамических признаков и выбранная модель Efficient-D0 успешно выделяют большинство очагов возгорания на изображениях. При этом полнота составила 89 % при пороге вероятности обнаружения ≥ 35 %. На данном этапе работы системы наиболее ценным является сравнение по метрике полноты. Показатели метрики точности будут улучшены в результате фильтрации обнаруженных объектов на следующих этапах работы.

3.3. Выбор модели НС

Так как на момент трекинга требуется определить максимальное количество областей кандидатов, и метрика Recall имеет существенные статистические отличия для всех моделей НС на всех под выборках, оценка лучшей модели НС была проведена по данной метрике. Данные для оценки приведены в таблице 3.

Таблица 3 – Подготовка данных для различных моделей НС

Модель НС	Цветовая обработка
EfficientDet-D0	ВФ + HBR + ФГ
EfficientDet-D1	ВФ + HBR + ФГ
EfficientDet-D2	ВФ + ФГ
FasterRCNN-ResNet50	БЦО
SSD-Mobilenet-V2	ВФ + HBR + ФГ

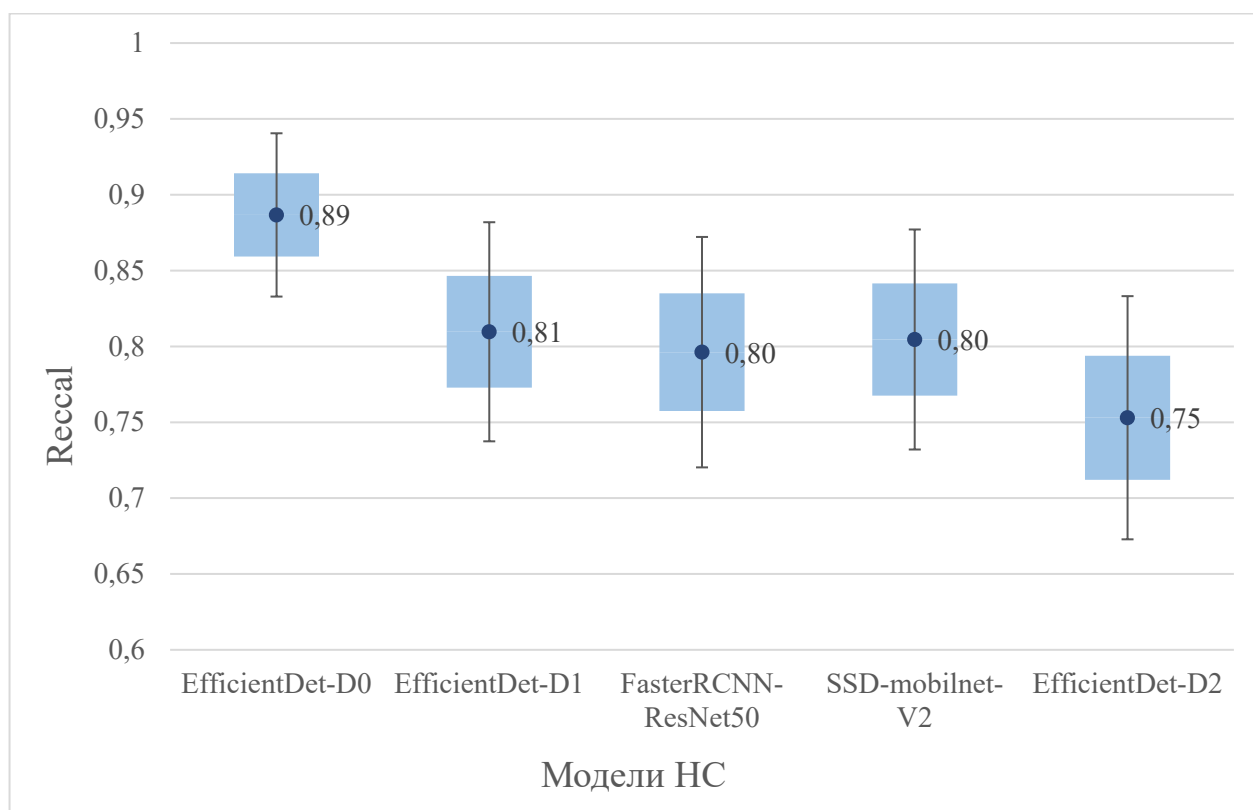


Рисунок 33 Диаграмма размаха средних значений сравнения различных моделей НС

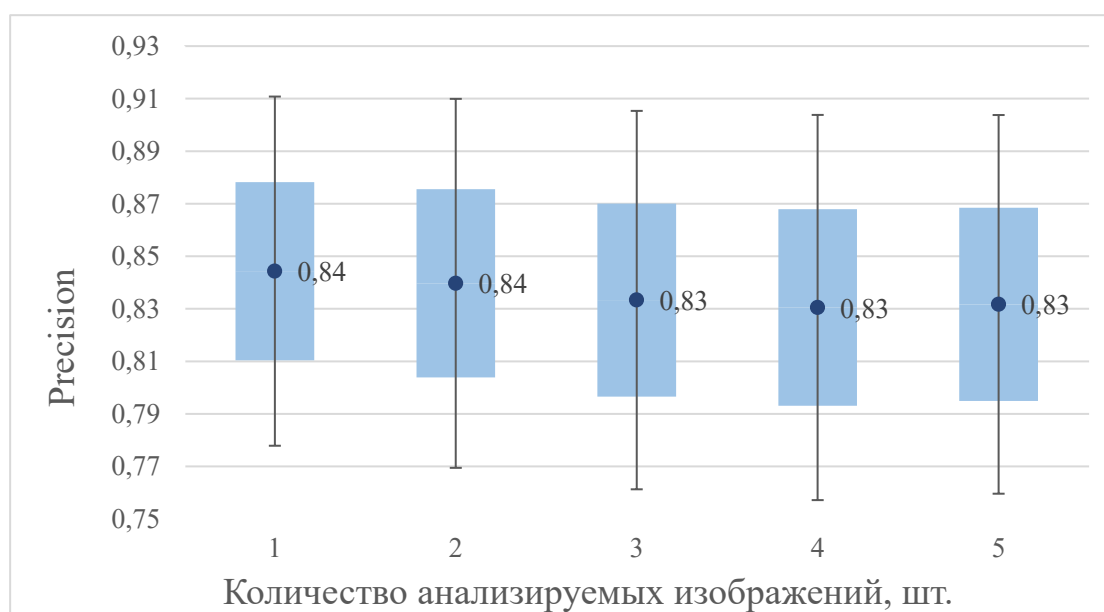
После определения наиболее значимых алгоритмов обработки для каждой модели НС был выполнен тест Краскела-Уоллиса для нахождения разницы между моделями НС. Нулевая гипотеза H_0 заключается в том, что различные модели НС имеют практически одинаковые результаты. Согласно полученным результатам, приведенным в таблице

Таблица 4, значения p имеет значение меньше уровня значимости, что говорит о том, что нулевая гипотеза может быть отклонена. На основании визуальных данных, представленных на рисунке 33, видно, что модель EfficientDet-D0 на трехкомпонентном наборе изображений с использованием авторского алгоритма вычитания фона, фильтра Гаусса и цветовой обработкой имеет лучшее значение Recall. Поэтому дальнейший анализ результатов будет оцениваться с использованием данной модели и цветовой обработки.

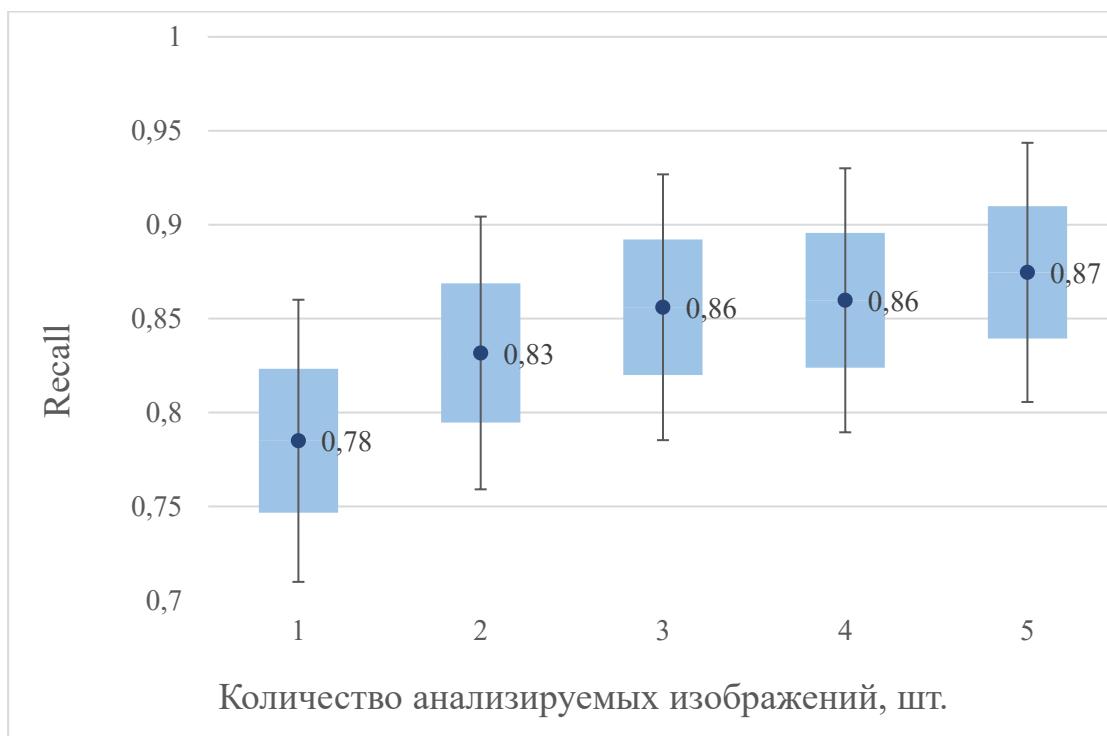
Таблица 4 – Тест Краскела-Уоллиса

Модель НС	Краскела-Уоллиса тест: $H(4, N=330) = 19,28914$ $p = 0,0007$				
	EfficientDet -D0	EfficientDet -D1	EfficientDet -D2	FasterRCNN- ResNet50	SSD- Mobilmnet-V2
Среднее значение рангов	199,16	171,58	155,49	133,63	169,64

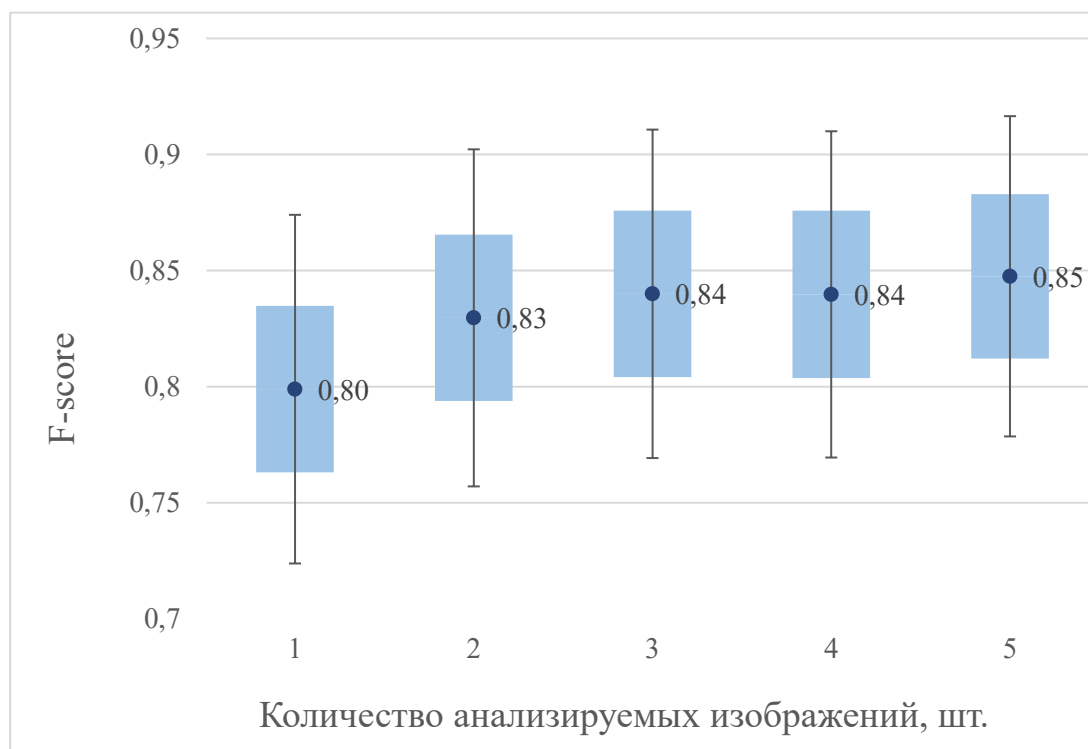
Так как мы работаем с видеофрагментами, необходимо анализировать последовательность кадров. Для этого проведем статическую проверку значимых различий по разным метрикам для последовательности кадров от 1 до 5 кадров, поступающих на анализ детектору. Для определения статистической значимости был выполнен тест Краскела-Уоллиса, нулевая гипотеза которого заключалась в том, что количество подаваемых изображений на анализ детектору имеют практически одинаковые результаты. Согласно результатам, приведенным в таблице 5, значимые отличия по средним значениям есть по метрике Recall. Согласно показателям данной метрики для анализа детектору требуется 5 изображений.



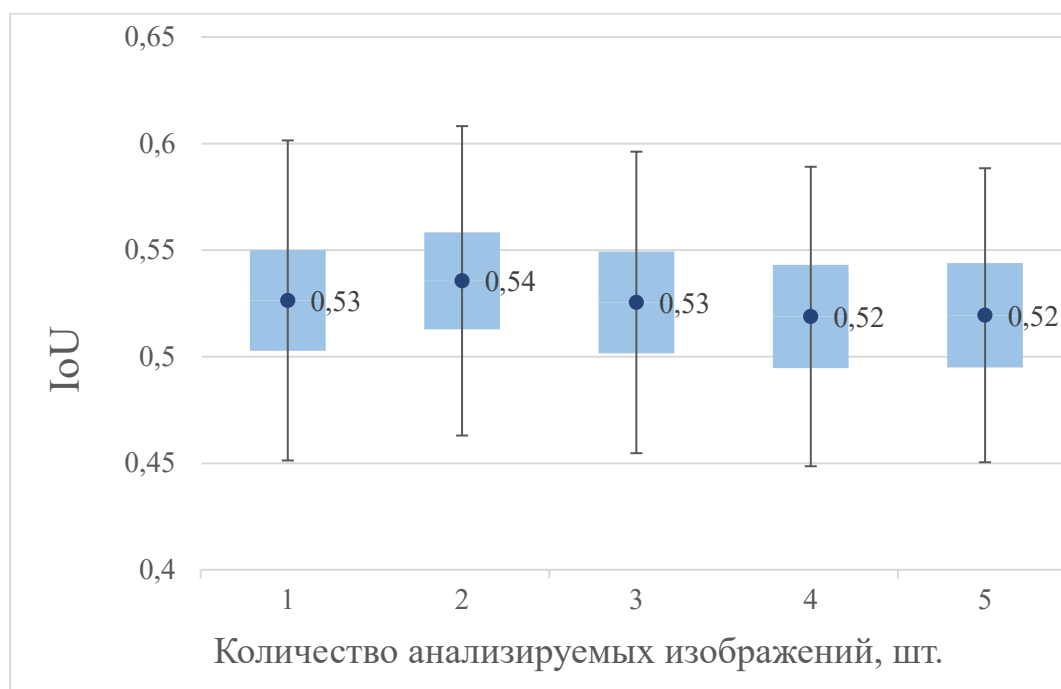
a) Precision



б) Recall



в) F-score



г) IoU

Рисунок 34 – Диаграммы размаха.

Таблица 6 – Тест Краскела-Уоллиса, сравнение значимости количества анализируемых изображений

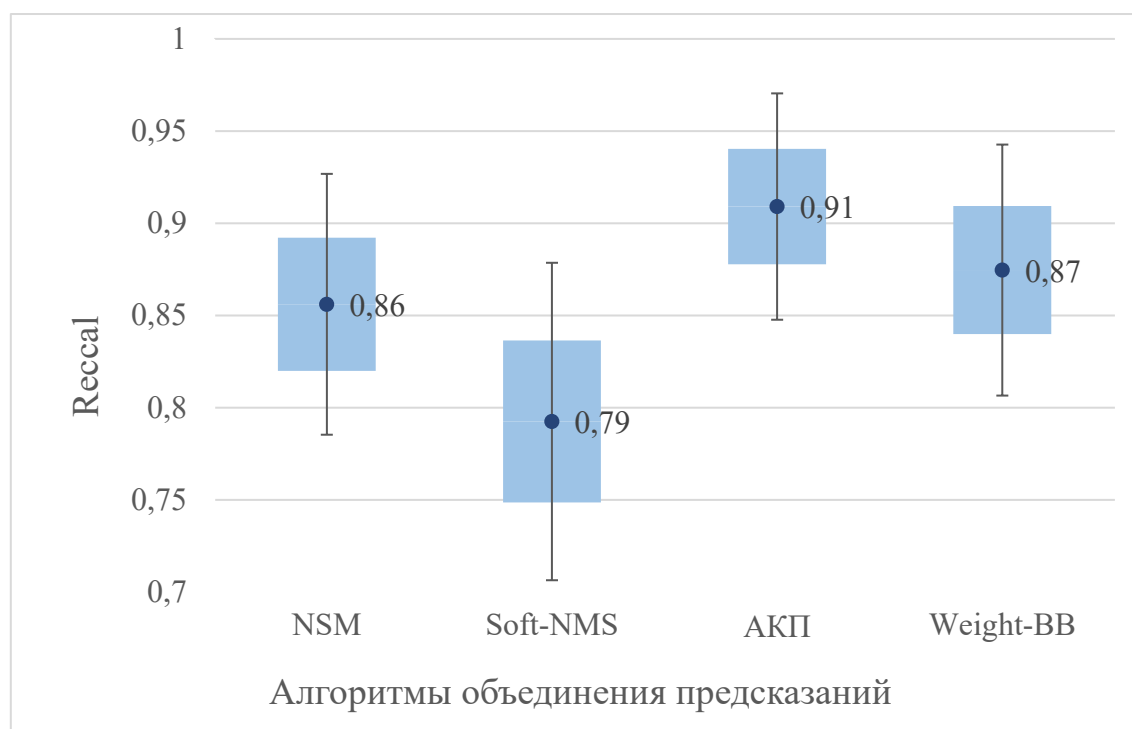
Количество картинок	mean					KW	p	Значимость
	1	2	3	4	5			
Precision	157,8	166,7	167,1	168,1	168,4	0,83	0,9	-
Recall	138,5	163,1	173,1	173,5	179,1	3,86	0,03	+
F-score	141.6	165.1	172.9	172.9	175.9	6,47	0,17	-
IoU	153	172.8	166.6	166.7	161.7	0.57	0,96	-

3.4. Результаты тестирования алгоритмов объединения предсказаний

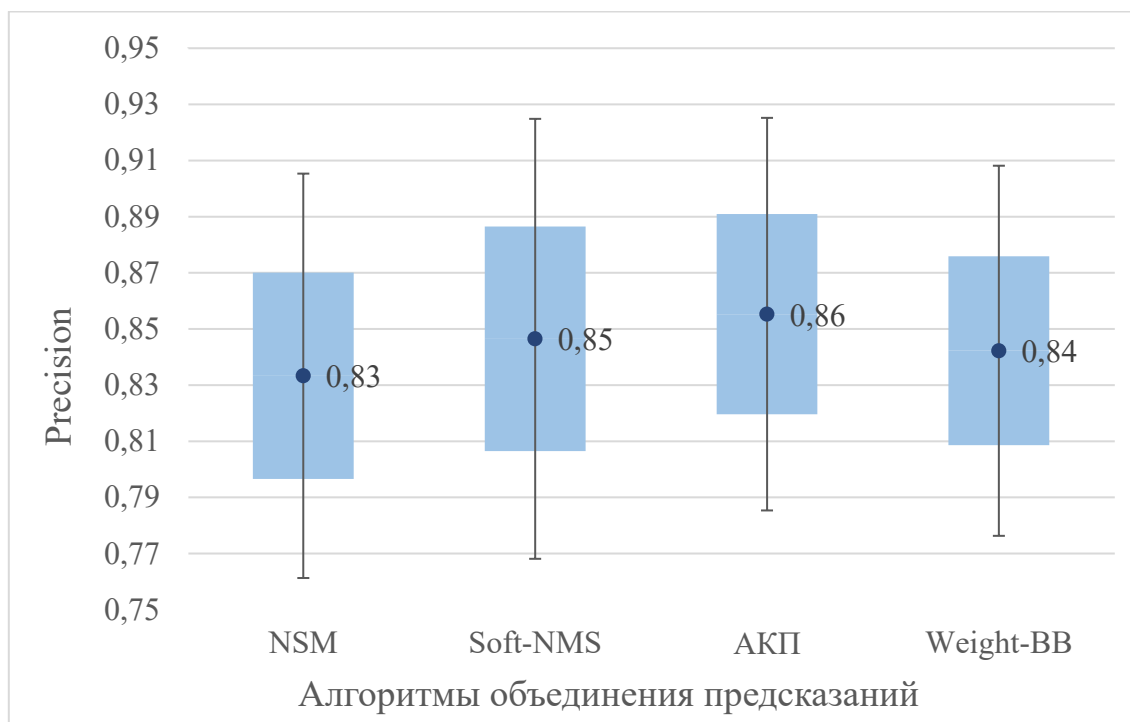
Аналогичный подход использовался для анализа выборок различных алгоритмов объединения предсказаний найденных объектов. В целях оценки значимости различий алгоритмов при разных значениях метрик использовался тест Краскела-Уоллиса. Как отмечалось выше, данный метод определяет следующее:

наблюдаются ли значимые отличия в исследуемых группах или нет. Нулевая гипотеза H_0 теста указывает на отсутствие отличий между группами.

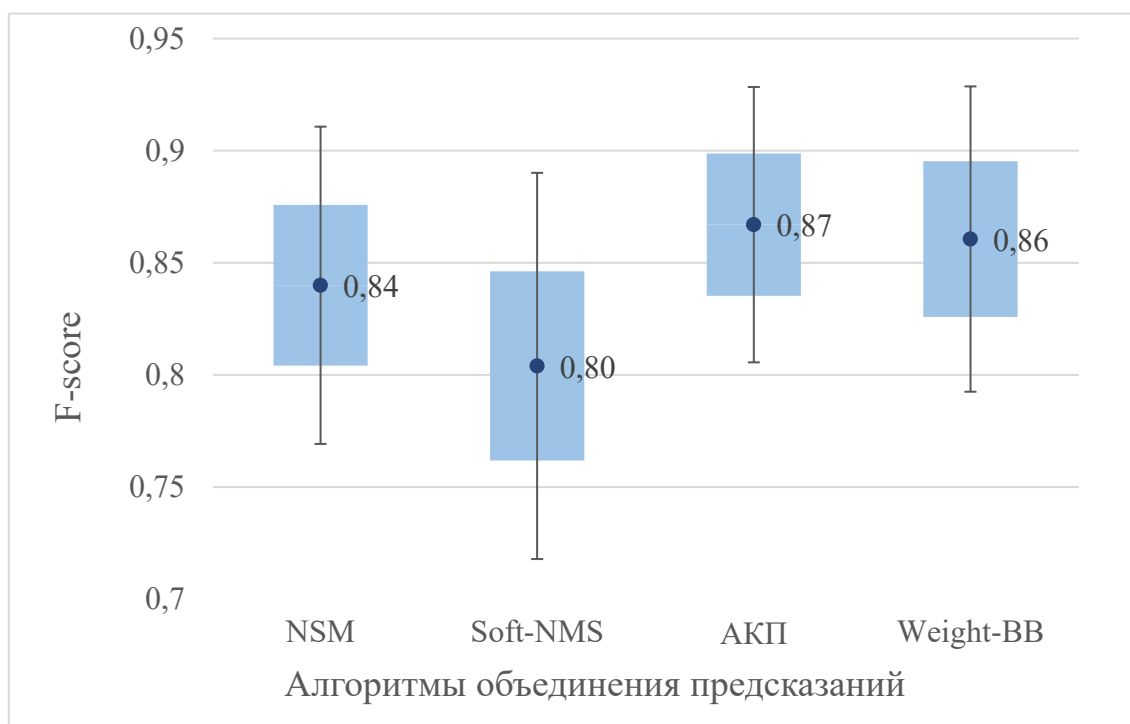
Результаты теста представлены в таблице 5 и на рисунке 35. Как видно из данных, значение p меньше выбранного уровня значимости равному 0,05 для метрики IoU. В связи с этим отвергаем нулевую гипотезу и констатируем, что одна из четырёх выборок имеет существенные отличия среднего значения. Согласно средним значениям метрики делаем вывод о том, что разработанный автором алгоритм наилучшим образом выделяет границы дымового облака в лесном массиве, при этом значительных изменений в точности и полноте обнаружений не наблюдается.



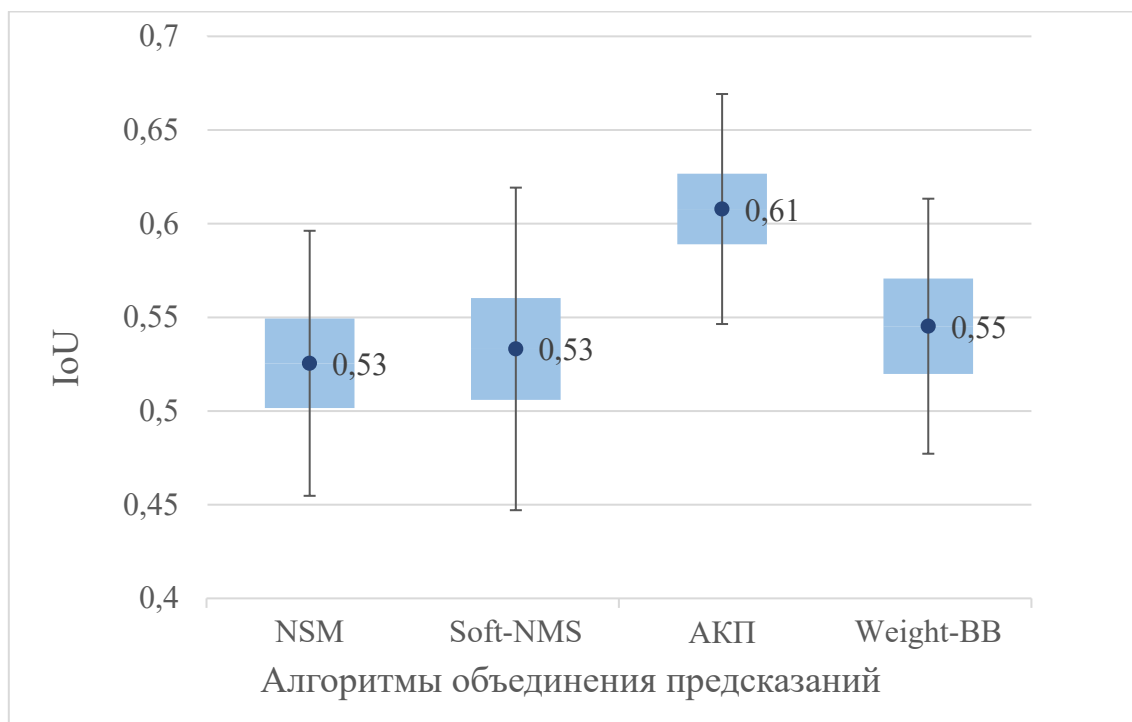
a) Recall



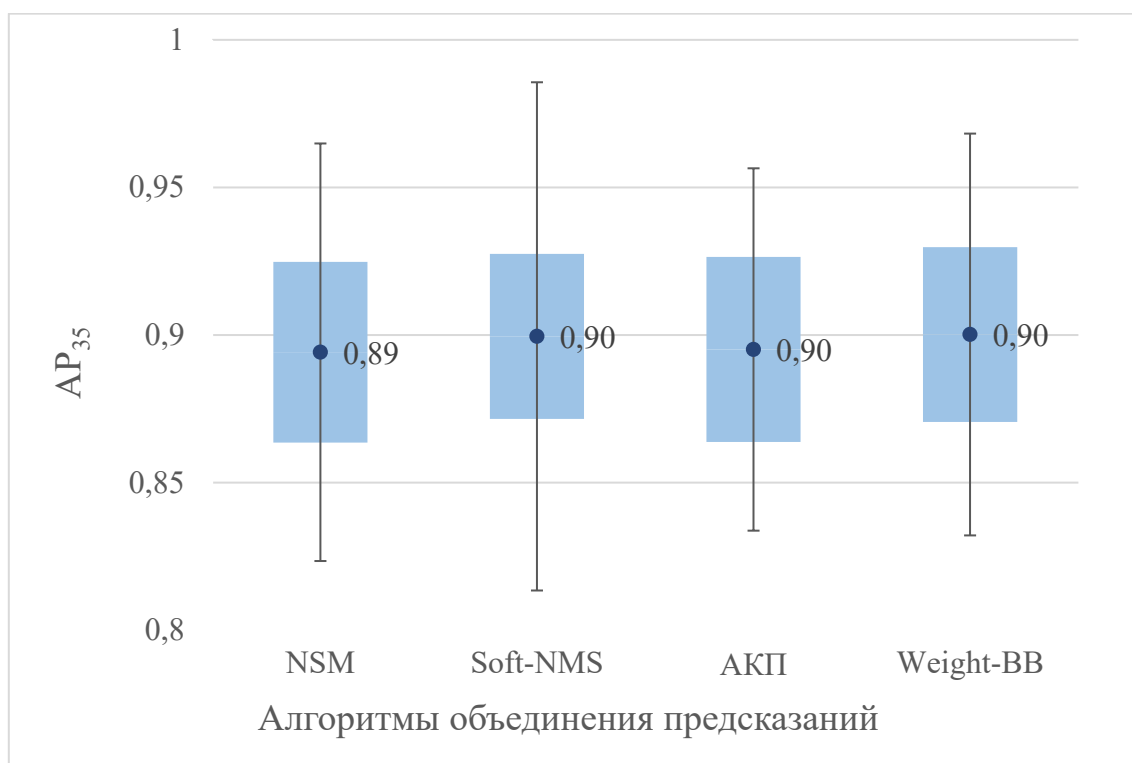
б) Precision



в) F-score



г) IoU



г) AP

Рисунок 35 – Диаграммы размаха.

Таблица 5 – Тест Краскела-Уоллиса, сравнение результатов объединения предсказаний

Метрика	Среднее значение рангов				KW	p	Значимость
	NMS	Soft-NMS	АКП	WBF			
Precision	141	130	137	121	3,64	0.3	-
Recall	122	129	143	134	4.53	0.2	-
F-score	131	133	126	138	1.09	0.7	-
IoU	128	119	151	130	5,92	0.11	+
AP ₃₅	126	131	138	133	1.25	0.74	—

При тестировании алгоритма кластеризации прогнозов обнаруженных объектов на первом этапе был осуществлен подбор оптимального количества кадров. Для каждого кадра обнаруженные объекты записываются в массив предсказаний объектов возгорания, представленных *координатами углов* ограничивающей рамки и *вероятностью* принадлежности области, ограниченной рамкой, к искомому классу. В результате тестирования было определено количество кадров для анализа равное 4. Применение алгоритма на четырех кадрах повысило показания полноты на 3 % при сохранившихся результатах по метрике точности. При этом реализация алгоритма позволила получить высокую эффективность объединения предсказаний обнаружения дыма, увеличив среднюю точность локализации найденных объектов на изображении на 7 % и составила по метрике IoU 62 % (рисунок 36).

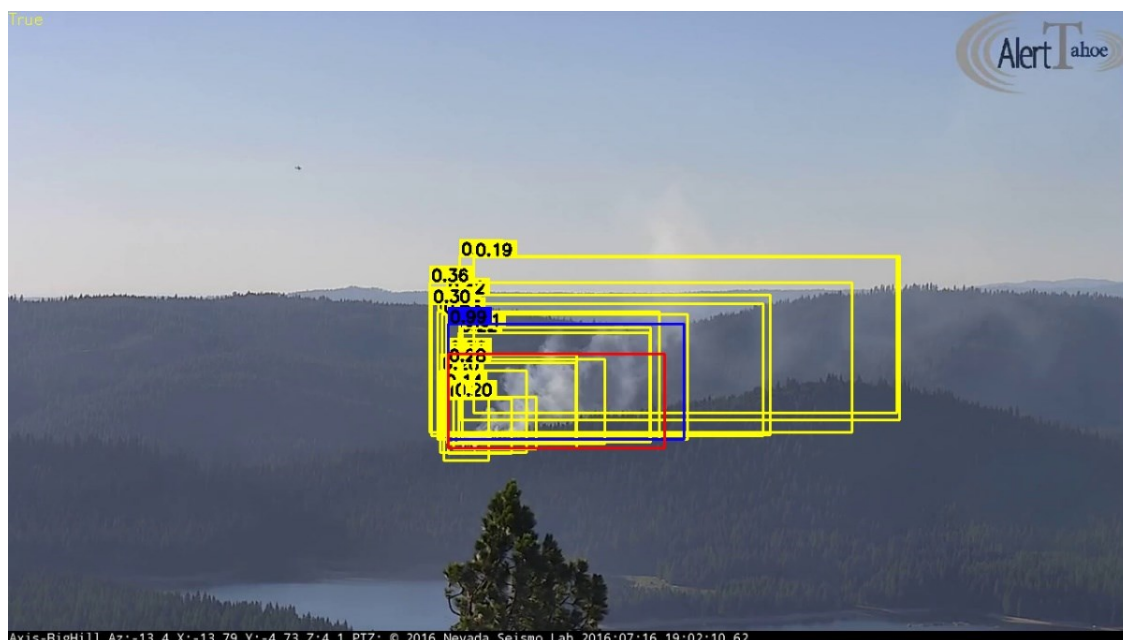


Рисунок 36 – Пример работы алгоритма кластеризации площадей «АКП» (боксы жёлтого цвета результаты предсказаний, бокс красного цвета – размеченный бокс, бокс синего цвета – работа алгоритма)

3.5. Результаты тестирования классификатора

В данном параграфе будет проведена оценка точности различных моделей сверточных нейронных сетей, являющихся частью гибридной модели нейронной сети для классификации пожароопасных объектов. Для определения оптимальной модели НС были выбраны параметры, представленные в таблице 6 – Параметры обучения гибридных нейросетей. Для обучения был собран датасет, состоящий из 36000 изображений с пожарами и 29000 изображений без пожаров.

Таблица 6 – Параметры обучения гибридных нейросетей

Параметр сети	Модель
Имя модели НС	VGG19, MobileNet-V2, ResNet50, Inception-V3, EfficientNet-B0
Кол. анализируемых изображений, шт.	2,3,4,5,6

Продолжение таблицы №6

Размер входного изображения, пикселей	64×64, 128×128, 256×256 для модели Inception-V3(150×150, 299×299)
Коэффициент скорости обучения	0,0001
Метод оптимизации	Adam
Функция активации	LeakyReLU
Функция потерь	Бинарная кросс энтропия (BCE)
Тип рекуррентной сети	GRU
Размер слоя GRU	128
Слой Dropout	0.5
Размер полносвязного слоя (FC)	512

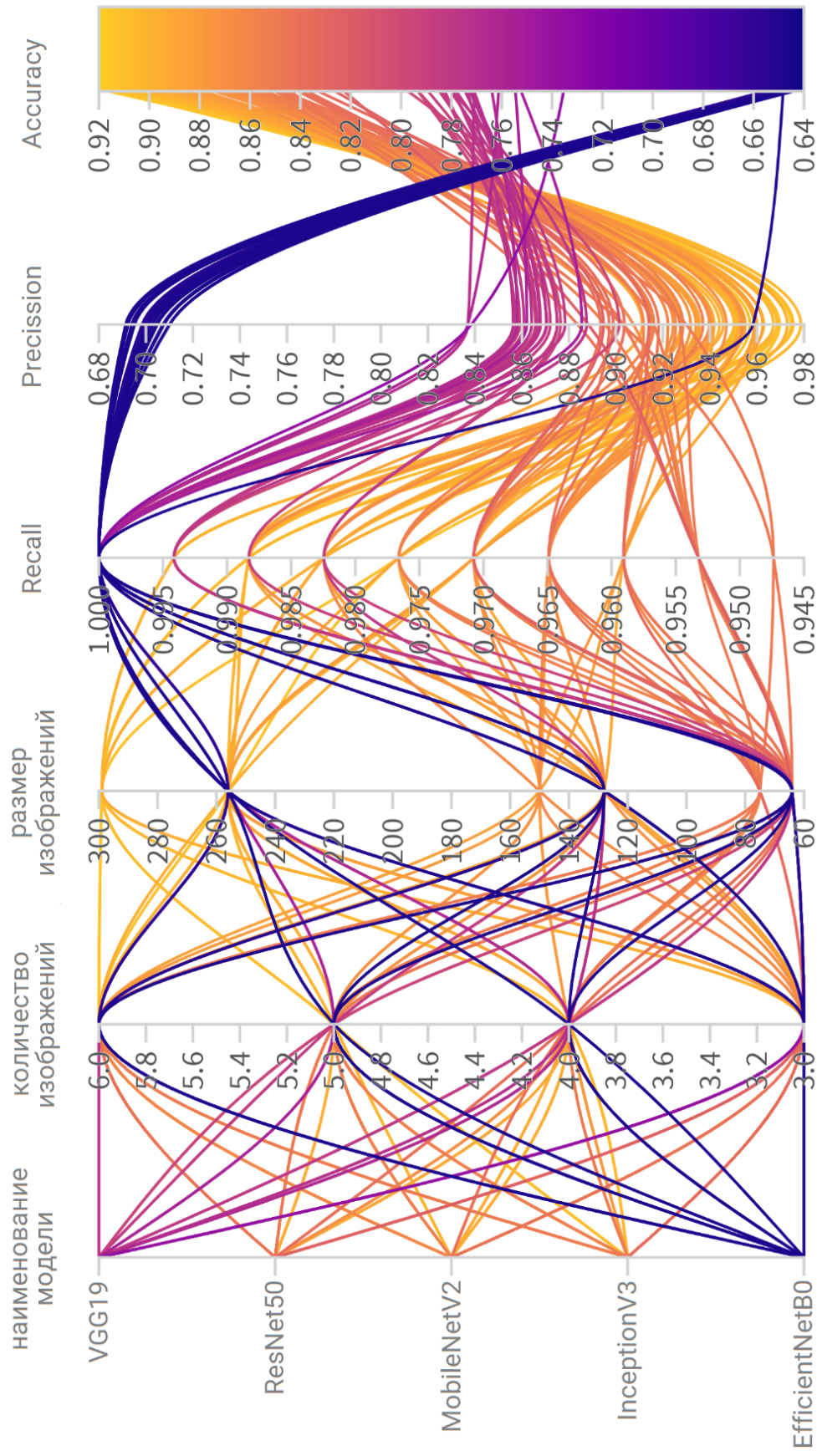


Рисунок 37 – График сравнения моделей НС

Таблица 7 – Результаты обучения моделей НС

Модель	Recall	Precision	Accuracy	F-score
VGG19	0,82	0,87	0,79	0,84
MobileNet-V2	0,94	0,92	0,91	0,93
ResNet50	0,94	0,91	0,90	0,92
Inception-V3	0,94	0,93	0,92	0,94
EfficientNet-B0	0,98	0,65	0,65	0,78

Результаты оценки моделей по метрике F-score представлены на рисунке 38. На представленном графике видно, что по данному показателю наилучший результат имеет модель Inception-V3, потому что показатель метрике F-score составил 94%. Для данной модели НС были проанализированы результаты влияния размера входного изображения на точность сети, см. рисунок 39. На представленном графике виден восходящий тренд прироста точности от увеличения размера изображения, при этом количество изображений не влияет на точность обнаружения. Для определения количества анализируемых кадров в последовательности обратимся к графику на рисунке 40. Исходя из результатов, представленном на данном графике, можно проследить восходящий тренд для метрики Recall и для метрики Precision в диапазоне от 2 до 4 изображений в последовательности. Последующие увеличения количества анализируемых кадров в последовательности не дает значимого увеличения точности по этим двум метрикам, поэтому для анализа будет использовано 4 кадра.

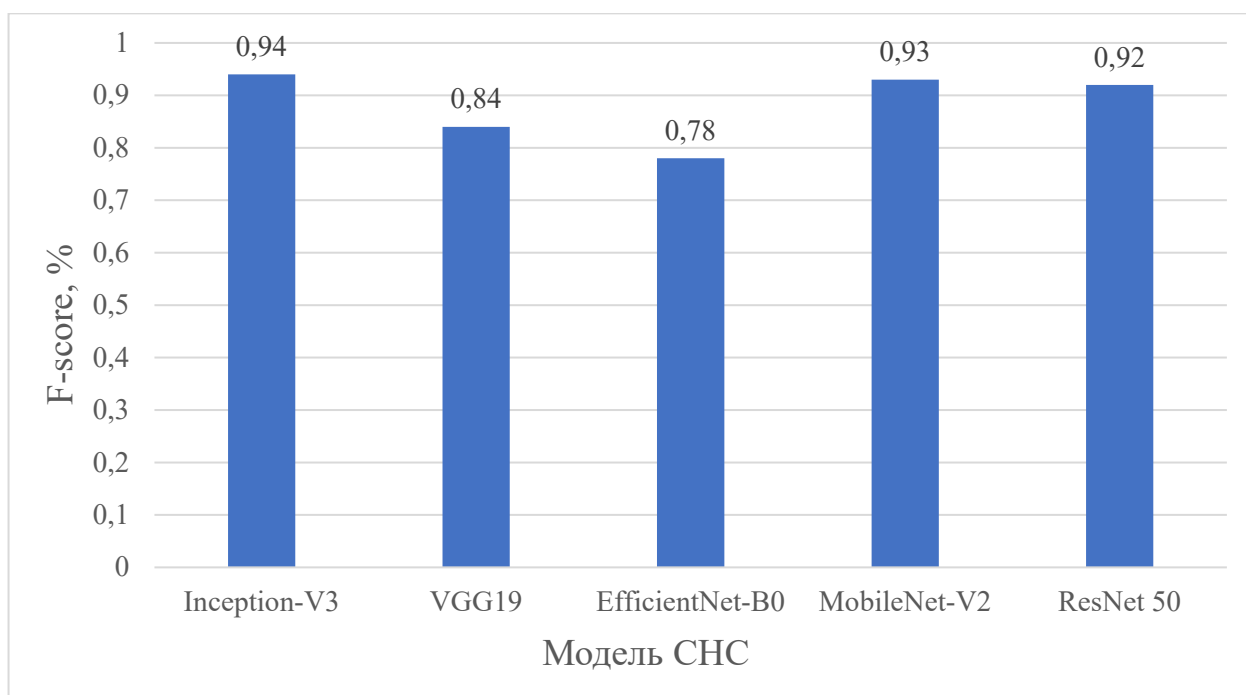


Рисунок 38 – Сравнение моделей НС по метрике F-score

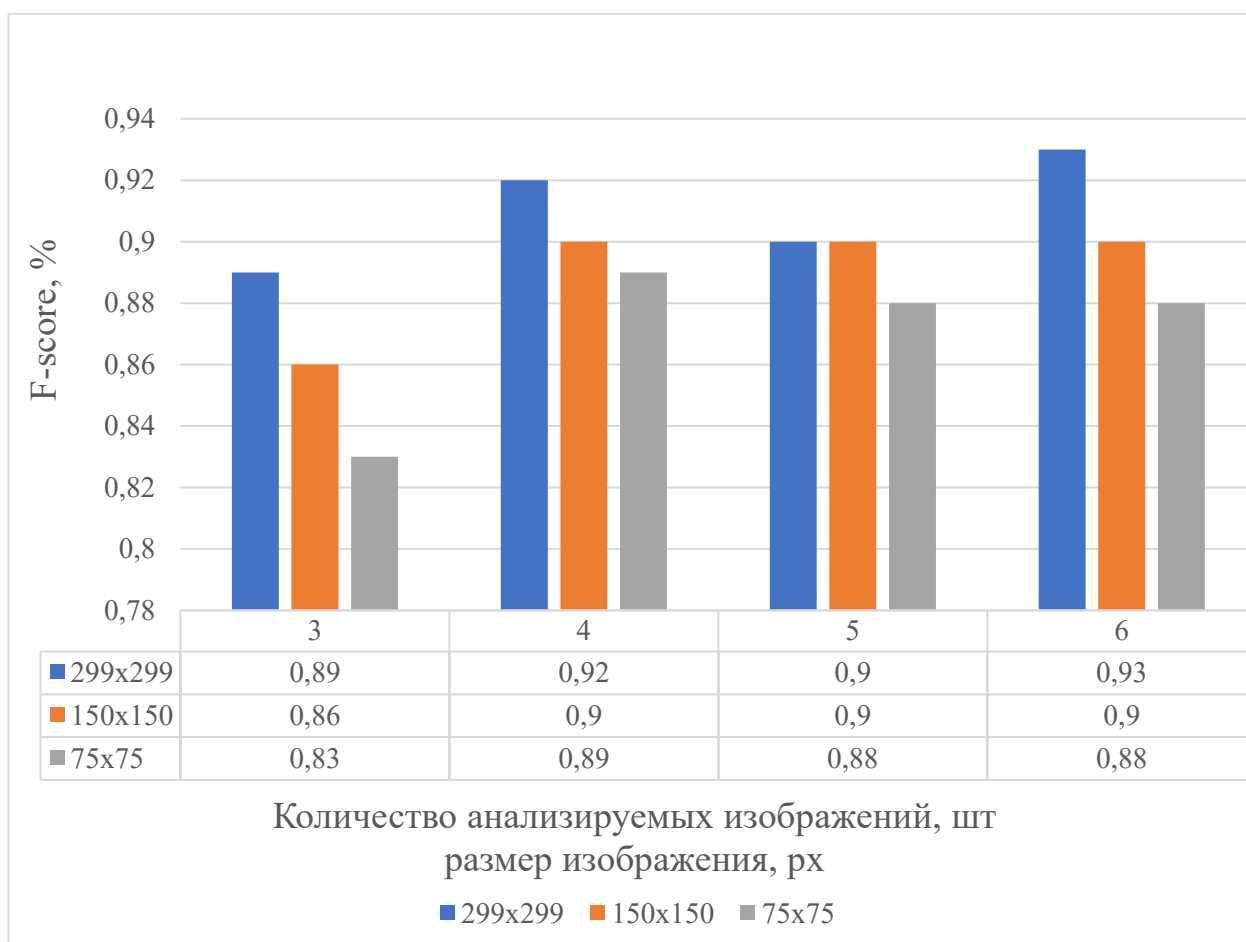


Рисунок 39 – Гистограмма оценки влияния анализируемого размера изображения на показатель метрики F-score модели Inception-V3

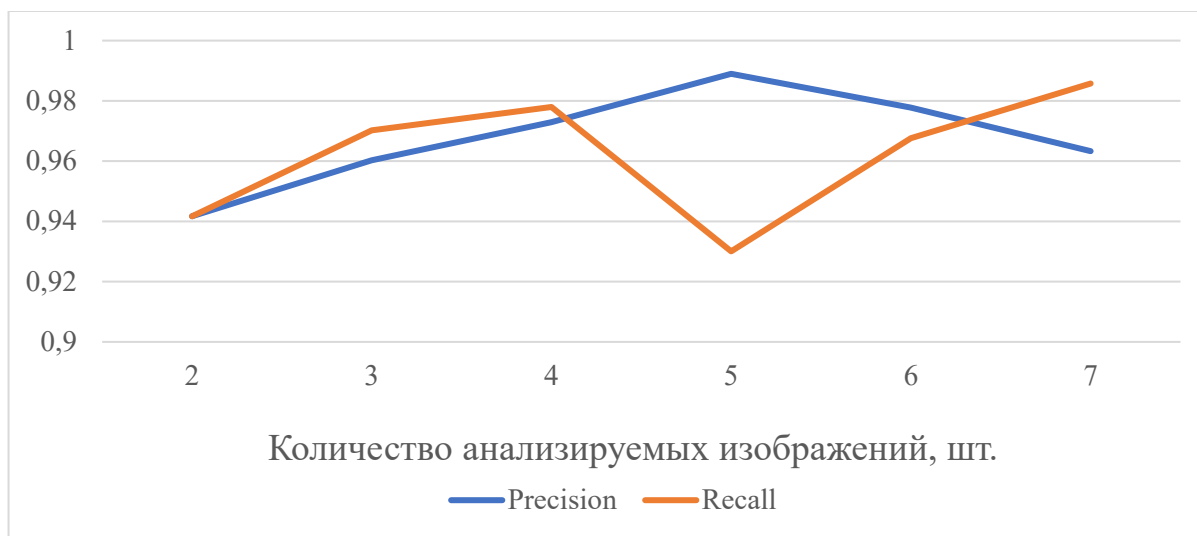


Рисунок 40 – График зависимостей метрик Precision, Recall от количества анализируемых изображений

С целью определения наиболее результативной архитектуры модели НС был проведен перебор гиперпараметров, представленных в таблице 8.

Таблица 8 – Исследуемые гиперпараметры обучения

Параметр сети	Модель
Размер входного изображения, px	150×150, 299×299
Тип рекуррентной сети	GRU, LSTM
Размер слоя РНС	256, 512, 1024
Размер полносвязного слоя (FC)	256, 512, 1024

Итоговый вид предложенной гибридной архитектуры НС представлен на рисунке 41.

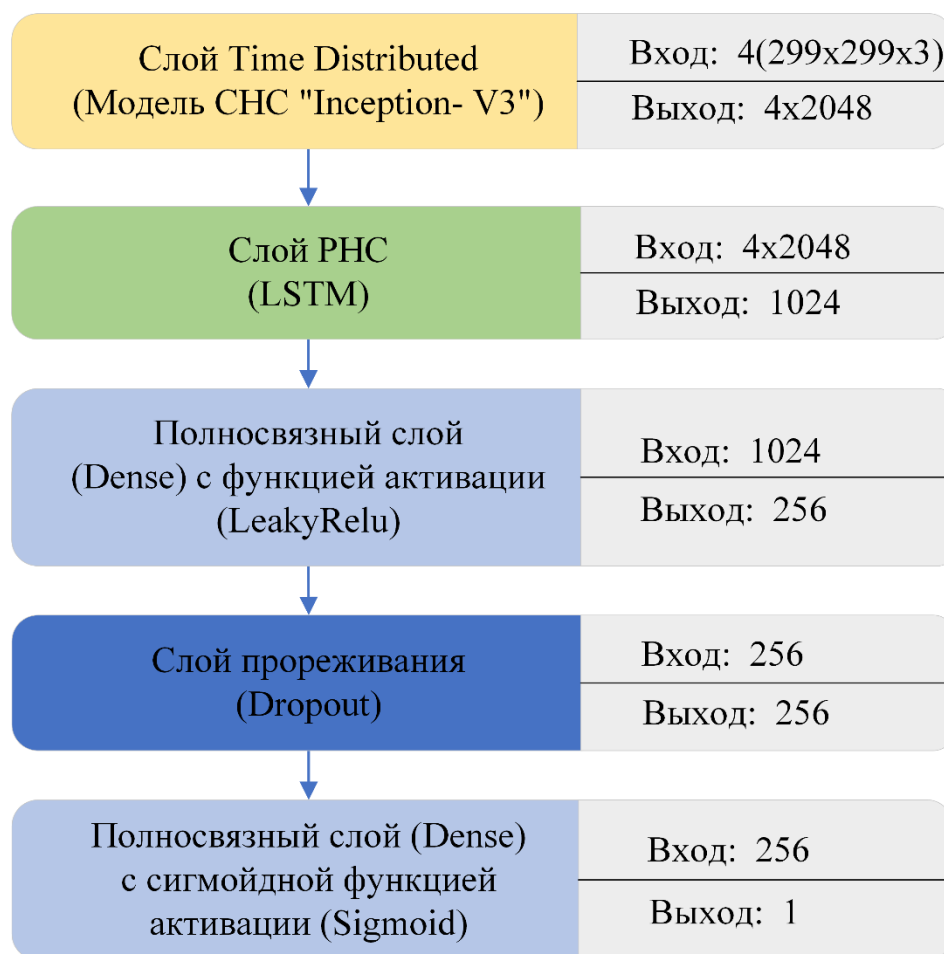


Рисунок 41 – Гибридная архитектура НС, основанная на СНС и РНС.

Результаты перебора гиперпараметров представлены на изображении

Ошибка! Источник ссылки не найден..

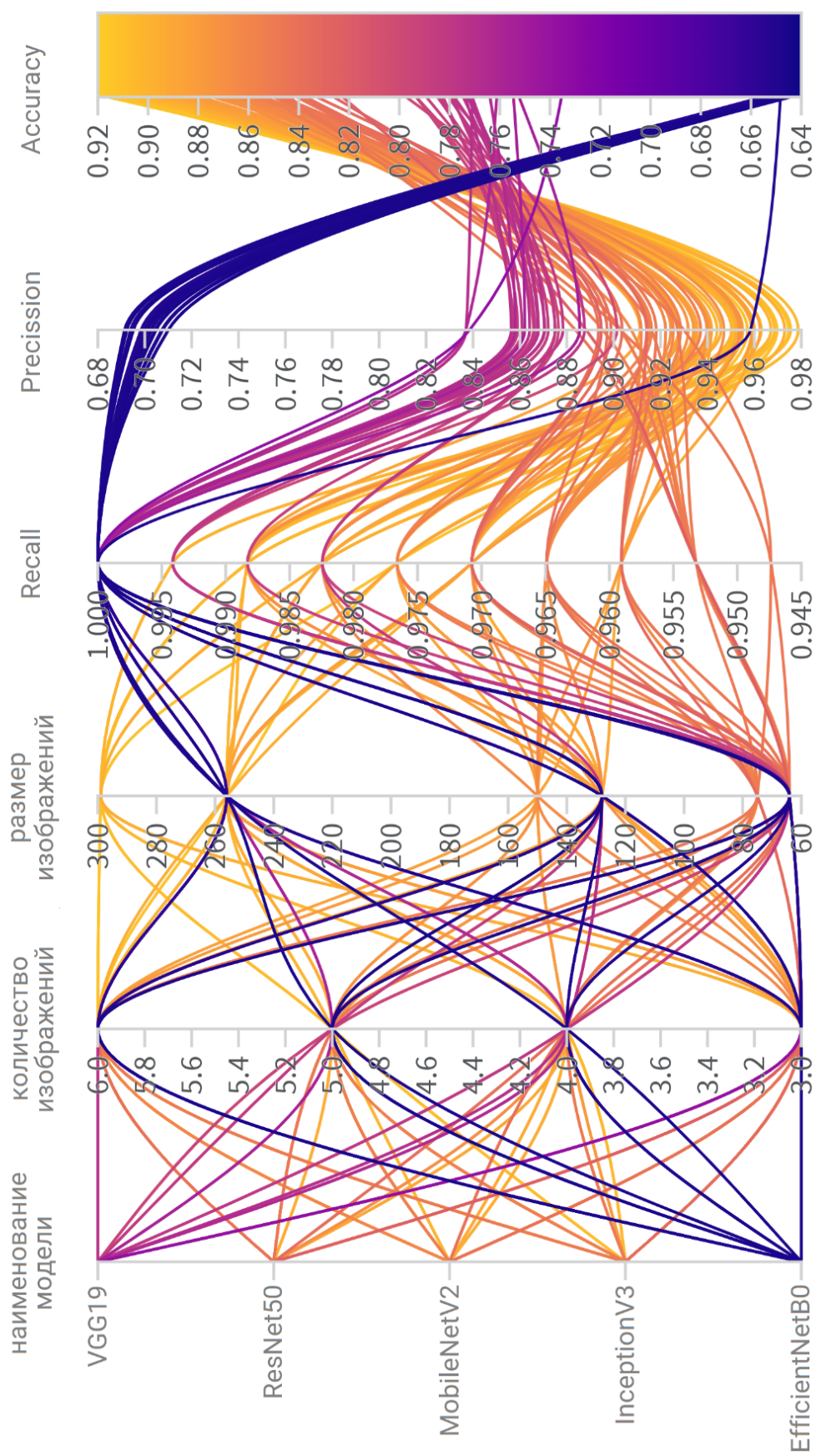


Рисунок 42 – График результатов обучения и перебора гиперпараметров гибридной нейронной сети

Тестовый набор данных содержал 169 видео с пожарами и 74 видео без пожаров. Метрики оценки классификации обнаруженных объектов представлены в таблице 9

Таблица 9 – Метрики оценки качества обнаруженных объектов

Метрика	Характеристики
TP	164 шт.
TN	72 шт.
FP	7 шт.
FN	5 шт.
Precision	95 %
Recall	97 %
Accuracy	96 %
F-score	96 %
AP ₃₅	95 %
Скорость обработки одного видео	0,7 секунды

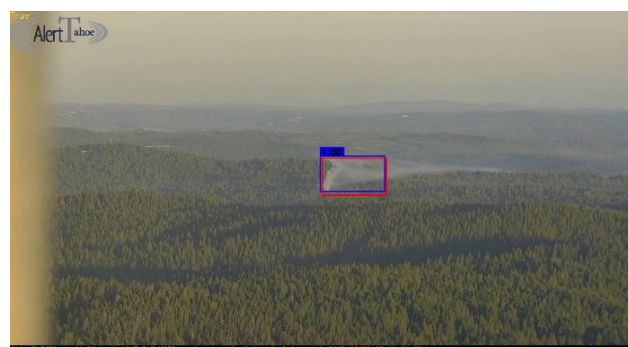
3.6. Результаты тестирования технологии раннего обнаружения пожароопасных объектов в лесном массиве

В данном параграфе проведено сравнение результатов авторской технологии обнаружения очага возгорания в лесном массиве с аналогами систем на базе видеозаписей HPWREN [123]. В целях оценки работоспособности и надежности предложенных алгоритмов проведено сравнение полученных результатов с аналогичными системами обнаружения, чьи характеристики есть в открытом доступе. Результаты сравнения представлены в таблице 10. Предложенная в диссертации технология обнаружения очага возгорания в лесном массиве демонстрирует более высокую точность обнаружения и меньшее количество ложных срабатываний.

Таблица 10 – Сравнение подходов обнаружения очага возгорания в лесном массиве

Подход		Precision, %	Recall, %	Accuracy, %
1	Система обнаружения лесных пожаров Fuego [12]	91,4	86,0	91,2
2	Система обнаружения лесных пожаров SmokeyNet [125]	90,9	76,1	83,6
3	Система обнаружения лесных пожаров Multimodal SmokeyNet [126]	87,0	71,0	80,0
4	Предложенная технология	95,4	97,2	95,6

На рисунке 43 представлены примеры обнаружения очага возгорания, где бокс красного цвета – это разметка, а бокс синего цвета, обнаруженный алгоритмом.



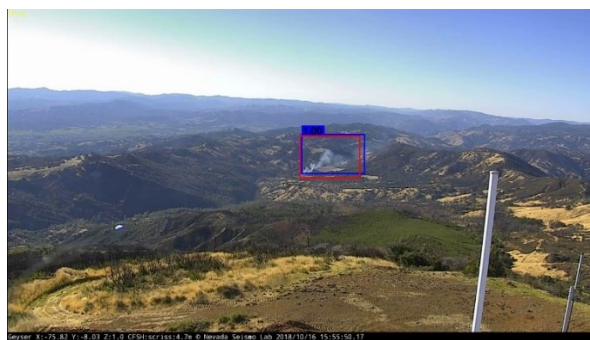


Рисунок 43 – Примеры положительных обнаружений

На рисунке 44 представлены примеры обнаружения областей кандидатов и их фильтрация гибридной нейронной сетью.

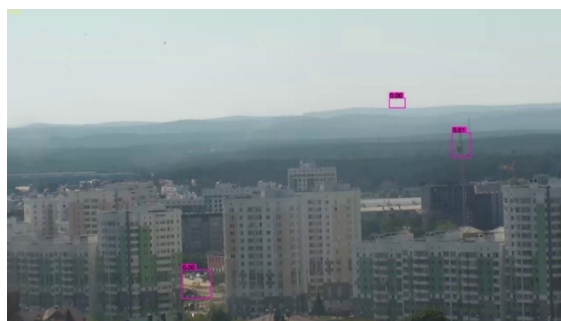
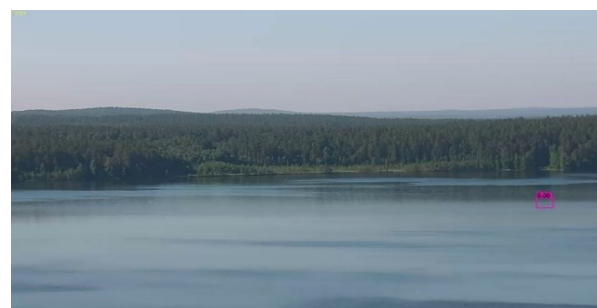
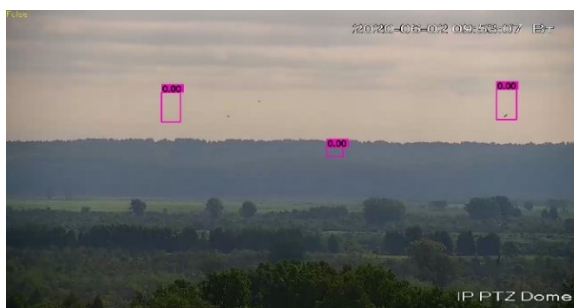
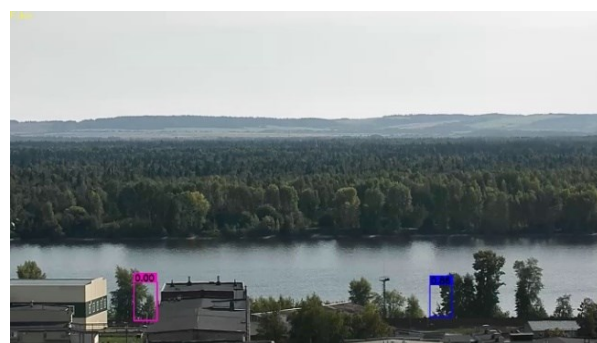


Рисунок 44 – Примеры работы системы: розовые боксы – это области кандидаты, обнаруженные детектором и отфильтрованные как «нет пожара» классификатором, синие боксы – это области кандидаты, классифицированные как «есть пожар»

3.7. Оценка работоспособности предложенных алгоритмов при решении различных задач

Представленные алгоритмы в технологии обнаружения объектов в видеоряде на основе машинного обучения обладают свойствами универсальности и были использованы и апробированы при решении задач в области медицины, робототехники, мониторинга местности. В данном разделе представлены примеры применения алгоритмов, а именно: выделения динамических признаков объектов, обнаружения объектов, фильтрация обнаруженных объектов на основе гибридной архитектуры НС.

3.7.1 Выделение динамических признаков для отслеживания перемещений дистального конца катетера

В работах [130–132] и ГЗ "Наука" №17.8205.2017/БЧ был применен алгоритм выделения динамических признаков с целью сегментации для отслеживания перемещений дистального конца катетера в составе лабораторного стенда. Принцип работы алгоритма распознавания формы непрерывного робота представлен несколькими этапами, показанными на рисунке 45. На первом этапе алгоритм одновременно получает изображения с двух камер. Две камеры создают стереоскопический эффект и позволяют получать информацию о пространственном положении робота. Этап предобработки основан на применении алгоритма выделения динамических признаков, описанный в пункте 2.2, предназначенный для улучшения качества сегментации. Входные данные представлены изображениями 256x256 пикселей.



Рисунок 45 – Принцип работы алгоритма распознавания образа гибкого манипулятора

Применение алгоритма выделения динамических признаков на этапе предобработки изображения позволило убрать шумы и за счет вычитания усредненного кадра выделить динамические структуры. Результат реализации алгоритма выделения динамических признаков представлен на изображении 46.



Рисунок 46 – Дистанционная карта маски

Сегментация выполняется с использованием RGB трешхолда. Как только двоичные маски получены, алгоритм скелетирует их, используя метод дистанционного преобразования, а затем вычисляет ограничивающие рамки. Скелетонизация области интереса (в частности дистального конца катетера) требуется для нахождения осевой линии робота на каждом изображении. Чтобы не обрабатывать все изображение, алгоритм определяет ограничивающие рамки области интереса (см. рисунок 47).

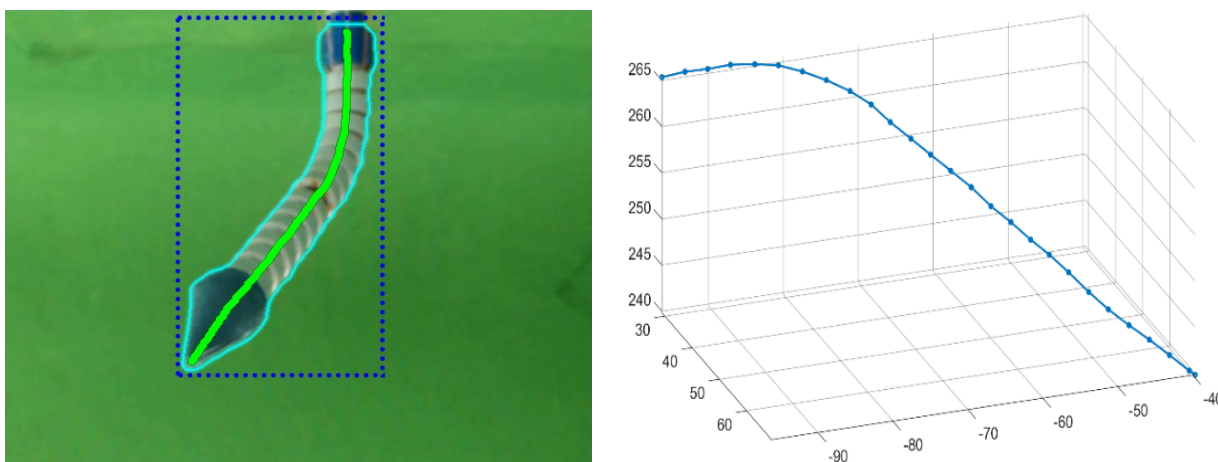


Рисунок 47 – Построение скелета с помощью алгоритма

После проведения предварительной обработки и сегментации следует процедура триангуляции. Для сопоставления пар точек используется метод градиентного спуска. Эта реализация позволяет алгоритму определять пары соответствующих точек на обоих скелетах. Целью применения метода

градиентного спуска является минимизация ошибки триангуляции, что отражено на Рисунок 48 между двумя лучами. Первый луч фиксируется, а направление второго выбирается таким образом, чтобы погрешность их триангуляции была самой низкой. Метод градиентного спуска анализирует текущий и соседние пиксели скелета и итеративно перемещается вдоль него, пока не будет достигнут минимум целевой функции. Начальное приближение метода является результатом минимизации предыдущей пары лучей. Эта реализация позволяет алгоритму минимизировать количество итераций и ускорить процесс триангуляции.

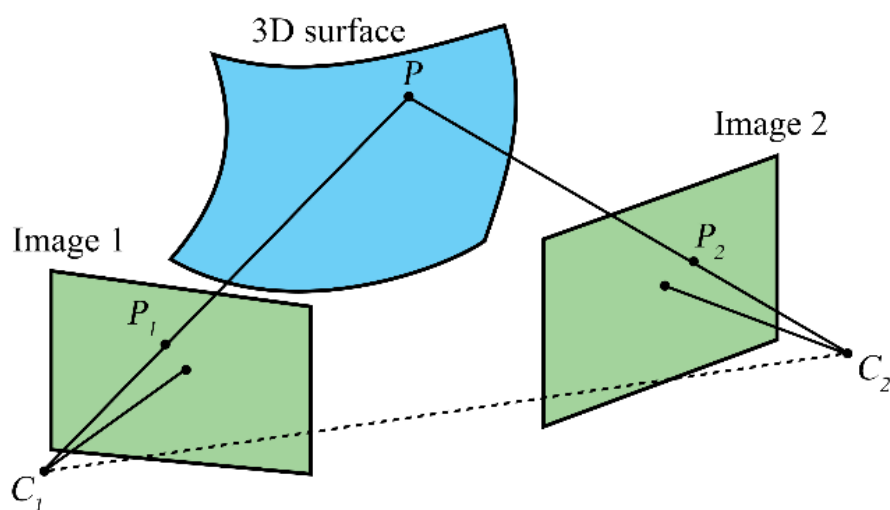


Рисунок 48 – Схема работы алгоритма перепроецирования

Описанный способ находит соответствие между точками первого и второго скелетов. Таким образом, мы можем найти пары точек на двух скелетах, триангулировать их и получить набор точек, соответствующих скелету робота в трёхмерном пространстве. Для того чтобы получить максимально возможное количество пар точек, алгоритм применяется ко всем пикселям скелета в обеих масках. Использование данного подхода позволяет улучшить результаты трекинга при сложных формах изгиба робота.

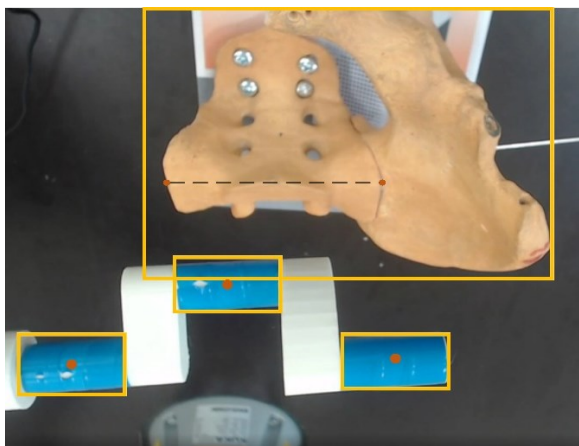
Для полученного набора точек необходимо определить правильный порядок. Для этого используется евклидова метрика. Процесс сортировки начинается с базы робота. На первом шаге алгоритм берет точку с наименьшей Z-координатой. Следующая точка будет ближайшей к выбранной точке. Этот процесс повторяется

итеративно, пока не будет найден точный порядок для всех наборов точек. В результате работы алгоритма получаем отсортированную последовательность, описывающую осевую линию робота.

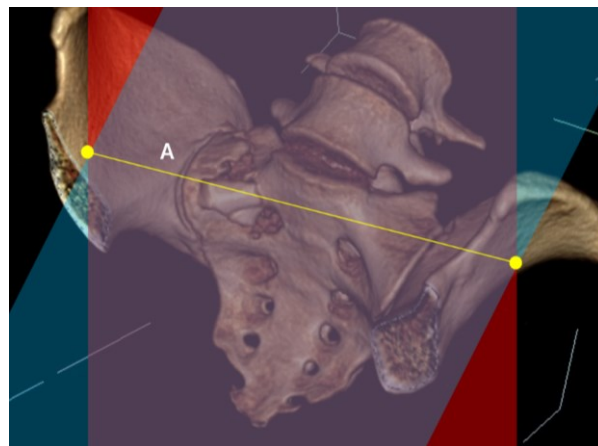
Таким образом, поиск оптимального набора точек сводится к интегрированию ранее полученной последовательности по длине центральной линии. Началом интегрирования является точка на центральной линии с наименьшей координатой Z_0 , из имеющихся двух массивов точек. Стоит отметить, что линейная интерполяция активно используется для повышения точности позиционирования. В результате интегрирования определяются точные координаты узловых точек, представляющих выходные данные алгоритма обнаружения.

3.7.2 Обнаружение объектов в решение задачи навигации роботов

В работе [133] применялись алгоритмы обнаружения объектов для распознавания и вычисления координат объектов. На рисунке 49 представлен пример отслеживания навигационной штанги и макета тазовой кости человека. Для перемещения штанги используется робот KUKA lbr iiwa 7 R800, в паре с программным обеспечением для удаленного управления с компьютера, используемого для обнаружения объектов. Навигационная штанга представляет собой разновидность коленчатого вала, состоящего из трех валов, предназначением которых является отслеживание угла поворота фланца робота. Система реализована для обнаружения штанги и определенной позиции тазовой кости с целью передачи координат объектов в систему.



а)



б)

Рисунок 49 – Обнаружение объектов с целью вычисления вектора для фиксации тазовых костей. а) пример макетирования; б) 3d моделирование объекта.

В качестве входных данных для системы являются координаты навигационной штанги, закрепленной на работе в стартовой позиции. В стартовой позиции центры валов образуют плоскость, которая в свою очередь образует перпендикуляр относительно плоскости пола. Камера устанавливается в двух положениях и образует нормаль к анатомическим точкам на тазовой кости.

Алгоритм работы системы состоит из следующих шагов:

1. Штанга закрепляется на манипуляторе и помещается в зону видимости видеокамеры или другого средства контроля зоны вмешательства (устройства слежения), например, рентгеновского аппарата или С-дуги.
2. Робот перемещается в плоскости пола до пересечения центральной оси навигационной штанги и нормали.
3. На изображении с помощью алгоритма обнаружения объектов выделяются валы с расчетом координат центра каждого из валов.
4. Поворот навигационной штанги осуществляется манипулятором до образования на изображении прямой, определяемой по координатам центров выделенных валов. Полученные координаты поворота сохраняются в памяти робота.

5. Повторение пунктов 2-4 алгоритма для нового положения камеры.

3.7.3 Фильтрация обнаруженных объектов с помощью гибридной архитектуры НС в задаче оценки парковочного пространства

Результаты работы алгоритма фильтрации обнаруженных объектов внедрены в ООО "Спотпаркинг" (г. Томск) в составе системы для обнаружения автомобилей на парковочном пространстве. Система анализа парковочного пространства позволяет определить габариты автомобилей на основе реализации методов сегментации изображений и оценки занятости парковочного пространства с помощью гибридной архитектуры НС, что в свою очередь дает возможность с высокой точностью оценить занятость парковочного места. Схема системы анализа парковочного пространства представлена на рисунке 50.

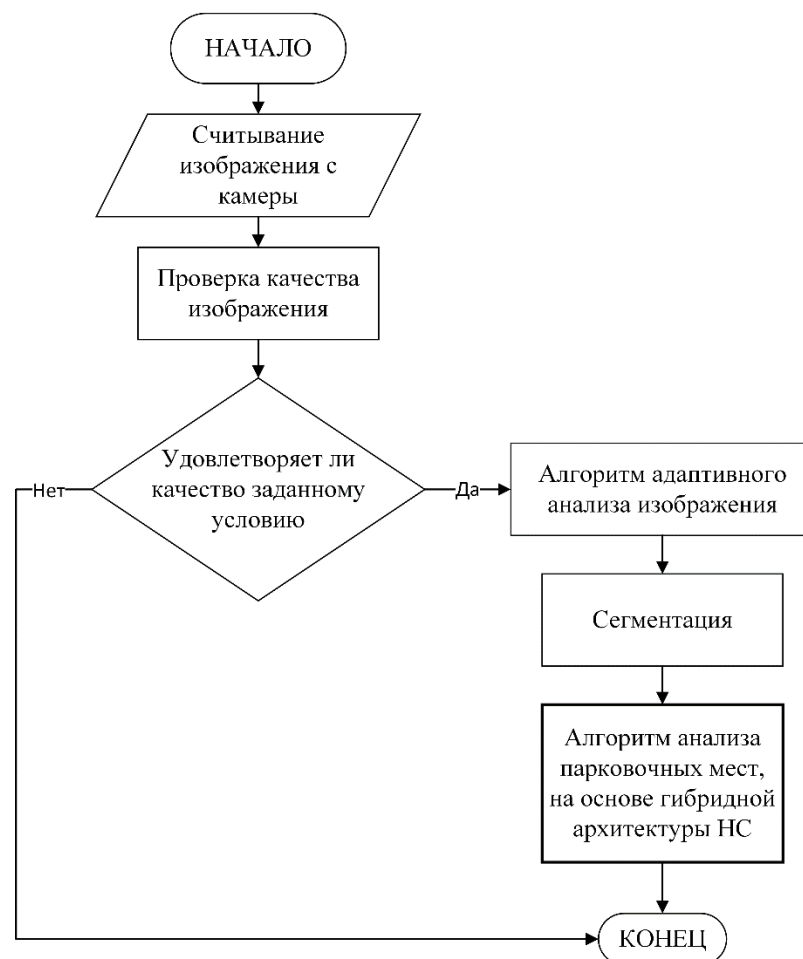


Рисунок 50 – Блок-схема алгоритма работы системы анализа парковочного пространства

Разметка парковочных мест на изображениях, полученных с камер видеонаблюдения, выполняется в ручном режиме. На изображение наносятся полигоны, состоящие из 4-х и более точек. Создание маркировки парковочного пространства выполняется при помощи web-платформы Supervisely [124].

Перед обработкой изображения ИНС кадр проходит проверку на “качество”. Данная проверка подразумевает определение условий видимости, а именно различимость объектов на кадре. Для выполнения данной проверки в работе предложено использовать алгоритм обнаружения границ, где для обнаружения границ реализован фильтр двумерной свертки с ядром:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Считается, что изображение обладает удовлетворительным (достаточным) качеством при условии, что процент присутствия границ на изображении выше 1,2%. Данное значение получено эмпирически при анализе кадров низкой видимости (сильный дождь, туман, снегопад и т.п.).

Для решения задачи сегментации парковочного пространства был выбран подход сегментации на основе экземпляров. Сегментация экземпляра является расширенным типом обнаружения объектов, которое генерирует карту сегментации для каждого обнаруженного экземпляра объекта. Сегментация экземпляра обрабатывает отдельные объекты как отличные сущности, независимо от класса объектов. В отличие от этого семантическая сегментация рассматривает все объекты того же класса как принадлежащие одной сущности. В качестве модели была выбрана Mask R-CNN – это метод, является модификацией Faster R-CNN путем добавления ветвей для предсказания объектов маски параллельно с существующей ветвью для обнаружения ограничивающей рамки (бокса). Для создания модели сегментации был выбран открытый фреймворк глубокого машинного обучения Pytorch [9]. Для более быстрого создания модели ИНС использована библиотека Segmentation Models [10]. Данная библиотека

предоставляет возможность быстрого создания моделей ИНС с использованием предварительно обученных энкодеров и располагает большим выбором архитектур моделей сегментации.

Для анализа парковочного пространства используется следующий алгоритм:

1. Определение парковочного места.

- Модель ИНС обрабатывает изображения с камер и представляет результат в виде изображения сегментационной маски и ограничивающей рамки.

2. Классификация парковочного пространства:

- Оценка занятости парковочного места согласно результатам обнаружения моделью Mask R-CNN;
- Выделение области парковочного места согласно заранее составленной полигональной разметки;
- Оценка занятости парковочного места на основе оценивания размеров и ориентации занимаемого пространства.

Результат обнаружения парковочных мест представлен в виде ограничивающей рамки для каждого объекта в кадре. Полученные результаты обнаружения используются для формирования последовательности фрагментов изображений (патчей), для последующей классификации гибридной нейронной сетью для оценки занятости парковочного пространства. Архитектура гибридной НС описана в пункте 2.4.3.

Для получения более точной оценки занятости парковочного места производится оценка не только наличия или отсутствия автомобиля на парковочном месте, но и оцениваются размеры и ориентация занимаемого пространства. С этой целью каждое анализируемое парковочное место представлено в виде квадратной матрицы 5×5 , инициализируемой нулями. Если вся область занята автомобилем, осуществляется преобразование матрицы в единичную. Если данная матрица содержит единицу или единицы не более чем в

одном крайнем столбце или строке, то место не будет считаться занятым. Во всех прочих ситуациях происходит оценка занимаемой площади парковочного места, и в случае, если это значение превышает 30%, то место считается занятым. Данный алгоритм оценки занятости парковочного места демонстрирует высокую точность. Также следует отметить, что дополнительный анализ краев парковочных мест делает систему «жесткой разметки» более гибкой, так как при смещении машин относительно заданной разметки этот факт будет учтен. Уточним, что под системой «жесткой разметки» подразумевается ручное создание полигональной сетки парковочных мест на изображении. Алгоритм оценки занятости парковочного места приведен на рисунке 51.

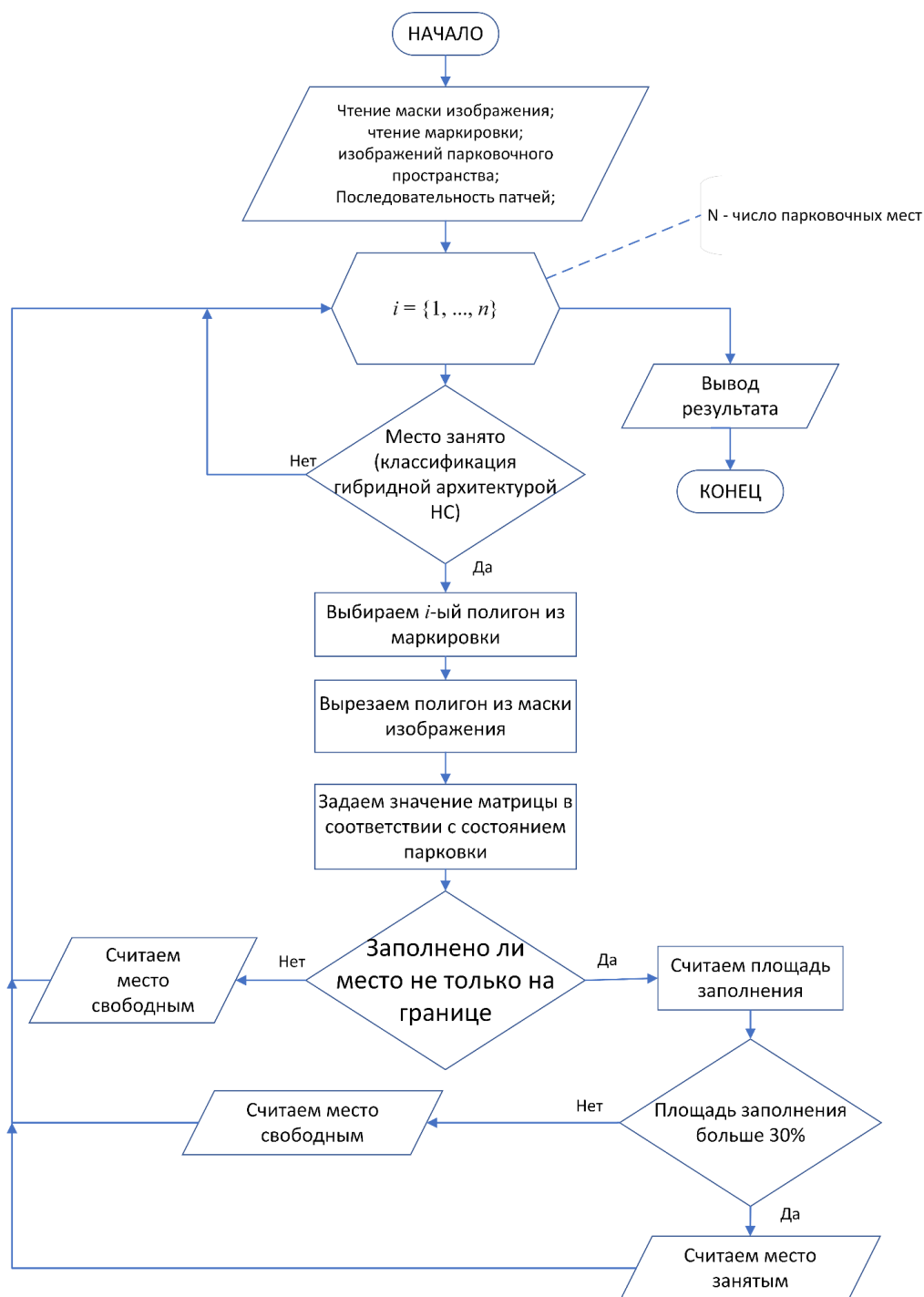


Рисунок 51 – Блок схема работы алгоритма классификации парковочного пространства

Результат работы системы приведен на рисунках 52-54, где под буквой: а) представлено исходное изображение; б) исходное изображение с наложением сегментационной маски (сегментированные объекты закрашены голубым цветом); в) - результирующее изображение, демонстрирующее классификацию

парковочных мест на предмет занятости (красным цветом отмечены занятые парковочные места, зеленым свободные парковочные места).



(a)

(б)

(в)

Рисунок 52 – Пример входного изображения (съемка с средней высоты с присутствием солнечного блика)



(a)

(б)

(в)

Рисунок 53 – Пример входного изображения (съемка с большой высоты, с присутствием тени большого размера)



(a)

(б)

(в)

Рисунок 54 – Пример входного изображения (съемка с низкой высоты)

3.8. Выводы по третьей главе

1. Получены результаты тестирования предложенных алгоритмов в рамках технологии раннего обнаружения очага возгорания в лесном массиве.
2. Реализован алгоритм выделения динамических признаков и выбрана модель детектора, которые успешно выделяют большинство очагов возгорания на изображениях; точность по метрике Recall составила 89 % при пороге вероятности обнаружения в 35 %.
3. Алгоритм кластеризации прогнозов показал высокую эффективность объединения предсказаний обнаружений, увеличив среднюю точность локализации найденных объектов на изображении на 6 %.
4. Реализован алгоритм фильтрации обнаруженных областей кандидатов. В основе алгоритма использована гибридная модель НС, основанная на сверточной и рекуррентной моделях нейросетей. В результате перебора параметров нейросети была сформирована гибридная архитектура НС для классификации обнаруженных объектов, а также выявлено количество изображений, необходимое для анализа. Применение классификатора увеличило показатель F-score на 9 %.
5. Сравнение полученных результатов с аналогичными системами обнаружения, характеристики которых представлены в открытом доступе, показало, что предложенная технология выделения объектов с динамическими признаками демонстрирует более высокую точность обнаружения и меньшее количество ложных срабатываний для задачи обнаружения лесных пожаров. Реализованные алгоритмы обладают свойствами универсальности, что подтверждено использованием и апробацией для решения задач в области медицины, робототехники и мониторинга местности.

ЗАКЛЮЧЕНИЕ

1. Предложена технология анализа изображений в видеоряде позволяющей достичь высокого качества обнаружения объектов, на основе предложенных алгоритмов локализации и классификации с учетом статических и динамических данных. Предложенные в технологии алгоритмы могут быть применены с целью анализа изображений для широкого круга практических задач.
2. Реализован алгоритм выделения динамических признаков на последовательности кадров, основанный на алгоритмах цветовой обработки изображений и операции вычитания фона. При реализации алгоритма осуществляется вычисление средней интенсивности пикселей кадров последовательности с последующим вычислением разности ключевого кадра с одним из исходных кадров. Использование разработанного алгоритма для модели обнаружения объектов EfficientDet-D0 позволило повысить значения метрики полноты (Recall) на 31 %, а метрики точности (Precision) на 21 %.
3. Реализован алгоритм, позволяющий объединять результаты нейросети в итоговое решение. Алгоритм выделяет плотные группы объектов данных, на основе оценки отношения площадей пересечения между соседними объектами, что отличает от других алгоритмов, базирующихся на фиксированном значении расстояния между объектами. Использование разработанного алгоритма позволяет повысить точность локализации на 6 % относительно альтернативных подходов.
4. Предложен алгоритм классификации обнаруженных объектов, состоящий из алгоритма подготовки последовательности изображений для анализа гибридной архитектурой нейронной сети. Использование гибридной архитектуры позволяет анализировать статические характерные для дыма признаки с учетом временной составляющей. Применение предложенного классификатора позволило увеличить показатель F-score на 9 %.

5. Результаты диссертационной работы были использованы при выполнении НИР грант УМНИК №16068ГУ/2020, грант РФФИ № 20-37-90055, грант ГЗ "Наука" №FFSWW-2020-0014. Внедрены в организациях в ООО "НТП КИ-БЕРЦЕНТР" (г. Томск), в ООО "Спотпаркинг" (г. Томск) и в учебном процессе в отделении информационных технологий Инженерной школы информационных технологий и робототехники ТПУ.

СПИСОК СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ

CNN convolutional neural network, сверточная нейронная сеть

DNN deep neural network, глубокая нейронная сеть

DL deep learning, глубокое обучение

FPS frames per second, частота кадров в секунду

IoU intersection over union, степень перекрытия двух участков изображений

mAP mean average precision, средняя точность

VGG19 вариант архитектуры сверточной нейронной сети, разработанный Visual Geometry Group

pooling нелинейное уплотнение карты признаков

softmax обобщение логистической функции для многомерного случая

MSE mean square error, среднеквадратическая ошибка

BPG backpropagation, метод обратного распространения ошибки

CE cross-entropy, кросс энтропия

SGD stochastic gradient descent, стохастический градиентный спуск

ADAM adaptive moment estimation, алгоритм градиентной оптимизации стохастических целевых функций первого порядка

SSD single shot multiBox detector, детектор объектов

YOLO you only look once detector, детектор объектов

Faster R-CNN детектор объектов

Mask R-CNN детектор объектов, включающий сегментацию экземпляров

FC fully connected, полносвязная сеть

FCN fully convolutional network, полностью сверточная сеть

NMS алгоритм немаксимального подавления, алгоритм обработки изображений, используемый для устранения дублирования объектов.

SOFT-NMS алгоритм мягкого немаксимального подавления, алгоритм обработки изображений, используемый для устранения дублирования объектов.

WBF, Weighted Boxes Fusion, алгоритм обработки изображений, используемый для устранения дублирования объектов.

FP False Positive, ошибка первого рода

FN False Negative, ошибка второго рода

TN True Negative, истинная отрицательная оценка

TP True Positive, истинная положительная оценка

Leaky ReLU Rectified Linear Unit, нелинейная функция активации

CPU central processing unit, центральный процессор

GPU graphic processing unit, графический процессор, как правило используется пакет NVIDIA CUDA для его программирования, в частности библиотеки cuBLAS и cuDNN

Система раннего обнаружение пожаров на основе камер наружного видеонаблюдения – это комплекс технических средств и программных решений, которые позволяют обнаруживать начальные стадии лесных пожаров, когда его размер не превышает 1000 м², что позволяет быстро потушить его и предотвратить распространение на большую площадь леса. Для обнаружения возгораний и передачи сигнала на диспетчерскую панель или систему оповещения используются видеокамеры, установленные на вышках, зданиях, мачтах и других объектах, что позволяет охватить большую территорию и обеспечить более эффективное обнаружение элементов огня и дыма относящиеся к пожару.

СПИСОК ЛИТЕРАТУРЫ

1. Li M., Huang B., Tian G. A comprehensive survey on 3D face recognition methods // Eng. Appl. Artif. Intell. 2022. Vol. 110. P. 104669.
2. Danilov V. V et al. Use of semi-synthetic data for catheter segmentation improvement // Comput. Med. Imaging Graph. Elsevier, 2023. Vol. 106. P. 102188.
3. Danilov V. V, Kolpashchikov D.Y., Laptev N. V. Automatic control of a continuous robot using the FABRIK algorithm // Model. Optim. Inf. Technol. 2019. Vol. 7, № 4. P. 1–2.
4. Poliak M., Jurecki R., Buckner K. Autonomous vehicle routing and navigation, mobility simulation and traffic flow prediction tools, and deep learning object detection technology in smart sustainable urban transport systems // Contemp. Readings Law Soc. Justice. Addleton Academic Publishers, 2022. Vol. 14, № 1. P. 25–40.
5. Laptev N. et al. Visualization System for Fire Detection in the Video Sequences // sv-journal.org. 2021. Vol. 13, № 2. P. 1–9.
6. Макунина Я.С., Никончук А.В. АНАЛИЗ ВОЗНИКНОВЕНИЯ ЛЕСНЫХ ПОЖАРОВ В РОССИИ // ИННОВАЦИИ В ХИМИКО-ЛЕСНОМ КОМПЛЕКСЕ: ТЕНДЕНЦИИ И ПЕРСПЕКТИВЫ РАЗВИТИЯ. 2021. P. 24–26.
7. Mahmoud M.A.I., Ren H. Forest Fire Detection Using a Rule-Based Image Processing Algorithm and Temporal Variation // Math. Probl. Eng. / ed. Bianco V. Hindawi, 2018. Vol. 2018. P. 7612487.
8. Çetin A.E. et al. Video fire detection–review // Digit. Signal Process. Elsevier, 2013. Vol. 23, № 6. P. 1827–1843.
9. Töreyn B.U. et al. Fire detection in infrared video using wavelet analysis // Opt. Eng. SPIE, 2007. Vol. 46, № 6. P. 67204.
10. Yuan C., Liu Z., Zhang Y. Fire detection using infrared images for UAV-based forest fire surveillance // 2017 International Conference on Unmanned Aircraft Systems (ICUAS). 2017. P. 567–572.
11. Stipaničev D. et al. Forest fire protection by advanced video detection system-Croatian experiences // Third TIEMS Workshop-Improvement of Disaster

Management System, Trogir. 2006.

12. Govil K. et al. Preliminary results from a wildfire detection system using deep learning on remote camera images // Remote Sens. Multidisciplinary Digital Publishing Institute, 2020. Vol. 12, № 1. P. 166.

13. Агеев В.Г. et al. ПРИМЕНЕНИЕ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ ДЛЯ МОНИТОРИНГА ЛЕСНЫХ И СТЕПНЫХ ПОЖАРОВ // Научный вестник НИИГД Респиратор. Государственный научно-исследовательский институт горноспасательного дела ..., 2019. № 3. P. 7–18.

14. Алчинов Е.И. et al. МОНИТОРИНГ ЛЕСНЫХ ПОЖАРОВ С ИСПОЛЬЗОВАНИЕМ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ // Редакционная коллегия ФШ Хафизов (отв. редактор) АВ Пермяков. 2018. P. 230.

15. Ксенофонтов Ю.Г. Использование современных информационных технологий в системах пожарного мониторинга лесных массивов // Актуальные проблемы обеспечения пожарной безопасности и защиты от чрезвычайных ситуаций. 2020. P. 211–217.

16. Kinaneva D. et al. Early Forest Fire Detection Using Drones and Artificial Intelligence // 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2019. P. 1060–1065.

17. Мельников Д.Б., Октябрьский В.В. ТЕПЛОВЫЕ ИНФРАКРАСНЫЕ КАНАЛЫ МНОГО-И ГИПЕРСПЕКТРАЛЬНОЙ АППАРАТУРЫ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ // Труды Военно-космической академии имени АФ Можайского. федеральное государственное бюджетное военное образовательное учреждение ..., 2016. № 651. P. 106–112.

18. Каткова О.Н., Карпова О.В. Мониторинг лесных пожаров по спутниковым данным // Инновации природообустройства и защиты окружающей среды. 2019. P. 412–417.

19. БЕНМУССА А. МОДЕЛИ И МЕТОДЫ АНАЛИЗА ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ МОНИТОРИНГА ЛЕСНЫХ ПОЖАРОВ // Дни науки студентов Владимирского государственного университета имени

Александра Григорьевича и Николая Григорьевича Столетовых. 2020. P. 52–57.

20. Chowdary V., Gupta M.K. Automatic forest fire detection and monitoring techniques: a survey // *Intelligent Communication, Control and Devices*. Springer, 2018. P. 1111–1117.

21. Bu F., Gharajeh M.S. Intelligent and vision-based fire detection systems: A survey // *Image Vis. Comput.* Elsevier, 2019. Vol. 91. P. 103803.

22. Xu R. et al. A Forest Fire Detection System Based on Ensemble Learning // *Forests*. 2021. Vol. 12, № 2.

23. Toan N.T. et al. A deep learning approach for early wildfire detection from hyperspectral satellite images // *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*. 2019. P. 38–45.

24. Rashkovetsky D. et al. Wildfire Detection From Multisensor Satellite Imagery Using Deep Semantic Segmentation // *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2021. Vol. 14. P. 7001–7016.

25. Park M. et al. Wildfire-Detection Method Using DenseNet and CycleGAN Data Augmentation-Based Remote Camera Imagery // *Remote Sens.* 2020. Vol. 12, № 22.

26. Oh S.H. et al. Early Wildfire Detection Using Convolutional Neural Network // *Frontiers of Computer Vision* / ed. Ohyama W., Jung S.K. Singapore: Springer Singapore, 2020. P. 18–30.

27. Jindal P. et al. Real-Time Wildfire Detection via Image-Based Deep Learning Algorithm // *Soft Computing: Theories and Applications* / ed. Sharma T.K. et al. Singapore: Springer Singapore, 2021. P. 539–550.

28. Khryashchev V., Larionov R. Wildfire Segmentation on Satellite Images using Deep Learning // *2020 Moscow Workshop on Electronic and Networking Technologies (MWENT)*. 2020. P. 1–5.

29. Qiao L., Zhang Y., Qu Y. Pre-processing for UAV Based Wildfire Detection: A Loss U-net Enhanced GAN for Image Restoration // *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*. 2020. P. 1–6.

30. Barmpoutis P. et al. Fire Detection from Images Using Faster R-CNN and Multidimensional Texture Analysis // ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019. P. 8301–8305.
31. Zhang G. et al. A Self-Adaptive Wildfire Detection Algorithm with Two-Dimensional Otsu Optimization // Math. Probl. Eng. / ed. Zdunek R. Hindawi, 2020. Vol. 2020. P. 3735262.
32. Jiao Z. et al. A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3 // 2019 1st International Conference on Industrial Artificial Intelligence (IAI). 2019. P. 1–5.
33. Zhang J. et al. ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition // IEEE Access. 2021. Vol. 9. P. 10858–10870.
34. Sinha D., Kumari R., Tripathi S. Semisupervised Classification Based Clustering Approach in WSN for Forest Fire Detection // Wirel. Pers. Commun. 2019. Vol. 109, № 4. P. 2561–2605.
35. Pan H. et al. Additive neural network for forest fire detection // Signal, Image Video Process. 2020. Vol. 14, № 4. P. 675–682.
36. Li Z., Mihaylova L., Yang L. A deep learning framework for autonomous flame detection // Neurocomputing. Elsevier, 2021. Vol. 448. P. 205–216.
37. Kinaneva D. et al. An artificial intelligence approach to real-time automatic smoke detection by unmanned aerial vehicles and forest observation systems // 2020 International Conference on Biomedical Innovations and Applications (BIA). 2020. P. 133–138.
38. Sowah R.A. et al. Hardware module design and software implementation of multisensor fire detection and notification system using fuzzy logic and convolutional neural networks (CNNs) // J. Eng. Hindawi, 2020. Vol. 2020.
39. Novák M. et al. Multisensor UAV system for the forest monitoring // 2020 10th International Conference on Advanced Computer Information Technologies (ACIT). IEEE, 2020. P. 293–296.
40. Di Biase V., Laneve G. Geostationary Sensor Based Forest Fire Detection

and Monitoring: An Improved Version of the SFIDE Algorithm // Remote Sens. 2018. Vol. 10, № 5.

41. Günay O. et al. Fire detection in video using LMS based active learning // Springer. 2010. Vol. 46, № 3. P. 551–577.

42. Gubbi J., Marusic S., Palaniswami M. Smoke detection in video using wavelets and support vector machines // Fire Saf. J. Elsevier, 2009. Vol. 44, № 8. P. 1110–1115.

43. Qi X., Ebert J. A computer vision based method for fire detection in color videos // Int. J. imaging. Citeseer, 2009. Vol. 2, № S09. P. 22–34.

44. journal T.C.-E., 2010 undefined. Fast and efficient method for fire detection using image processing // Wiley Online Libr. 2010. Vol. 32, № 6. P. 881–890.

45. Lee B., Han D. Real-time fire detection using camera sequence image in tunnel environment // International Conference on Intelligent Computing. Springer, 2007. P. 1209–1220.

46. Kolesov I. et al. Fire and smoke detection in video with optimal mass transport based optical flow and neural networks // 2010 IEEE International Conference on Image Processing. IEEE, 2010. P. 761–764.

47. Töreyn B.U. et al. Computer vision based method for real-time fire and flame detection // Pattern Recognit. Lett. 2006. Vol. 27, № 1. P. 49–58.

48. Yuan C. Automatic Fire Detection Using Computer Vision Techniques for UAV-based Forest Fire Surveillance. Concordia University, 2017.

49. Alamgir N. Computer vision based smoke and fire detection for outdoor environments. Queensland University of Technology, 2020.

50. Васюков В.Н., Зайцева А.Ю., Бондаренко В.В. СИСТЕМА РАННЕГО ОБНАРУЖЕНИЯ ЛЕСНЫХ ПОЖАРОВ - АРХИТЕКТУРА И АЛГОРИТМЫ // Доклады АН ВШ РФ. Novosibirsk State Technical University, 2015. № 2. P. 43–56.

51. Васюков В.Н., Зайцева А.Ю. АЛГОРИТМЫ АНАЛИЗА ИЗОБРАЖЕНИЙ ДЛЯ СИСТЕМЫ РАННЕГО ОБНАРУЖЕНИЯ ЛЕСНЫХ ПОЖАРОВ // Доклады Академии наук высшей школы Российской Федерации.

Федеральное государственное бюджетное образовательное учреждение высшего ..., 2014. № 4 (25). Р. 57–69.

52. Зайцева А.Ю., Васюков В.Н. Алгоритм сегментации изображений для системы противопожарного мониторинга // ISBN 978-5-7638-2687-6© Сибирский федеральный университет, 2014© Институт инженерной физики и радиоэлектроники СФУ, 2014. 2014. Р. 165.

53. Зайцева А.Ю., Васюков В.Н. Имитационная модель для генерирования последовательностей изображений дымовых облаков в процессе развития на фоне лесного массива. 2019.

54. Слезин К.А., Нурутдинов Г.Н., Ивановский М.А. ПОСТРОЕНИЕ ГИС ДЛЯ МОДЕЛИРОВАНИЯ ЛЕСНЫХ ПОЖАРОВ // Информационно-сенсорные системы в теплофизических исследованиях. 2018. Р. 312–316.

55. Слезин К.А., Рыжков А.П., Севенюк В.В. ИСПОЛЬЗОВАНИЕ НЕЙРОНЕЧЕТКОЙ КЛАССИФИКАЦИИ ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИКИ КОНТУРОВ ЛЕСНЫХ ПОЖАРОВ // Информационно-сенсорные системы в теплофизических исследованиях. 2018. Р. 317–320.

56. Громов Ю.Ю., Слезин К.А. Построение ГИС для оценки динамики контуров лесных пожаров // Приборы и системы. Управление, контроль, диагностика. Общество с ограниченной ответственностью Издательство Научтехлитиздат, 2018. № 4. Р. 16–28.

57. Громов Ю.Ю., Слезин К.А., Ивановский М.А. Нейронечеткий классификатор моделей интеллектуальной геоинформационной системы моделирования контуров пожаров // Промышленные АСУ и контроллеры. Общество с ограниченной ответственностью Издательство Научтехлитиздат, 2018. № 10. Р. 15–21.

58. Cappellini V., Mattii L., Mecocci A. An intelligent system for automatic fire detection in forests // Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). Springer Verlag, 1989. Vol. 399 LNCS. P. 351–364.

59. Chen T. et al. An early fire-detection method based on image processing //

ieeexplore.ieee.org.

60. Dimitropoulos K. et al. Flame detection for video-based early fire warning for the protection of cultural heritage // Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 2012. Vol. 7616 LNCS. P. 378–387.
61. Çelik T., Demirel H. Fire detection in video sequences using a generic color model // Fire Saf. J. 2009. Vol. 44, № 2. P. 147–158.
62. Marbach G., Loepfe M., Brupbacher T. An image processing technique for fire detection in video images // Fire Saf. J. 2006. Vol. 41, № 4. P. 285–289.
63. Yamagishi H., 1999 J.Y.-M.P. of, 1999 undefined. Fire flame detection algorithm using a color camera // ieeexplore.ieee.org.
64. Gonzalez R.C. Digital image processing. Pearson education india, 2009.
65. Ning C., Fei D. Flame Object Segmentation by an Improved Frame Difference Method // 2012 Third International Conference on Digital Manufacturing Automation. 2012. P. 422–425.
66. Vijayalakshmi S.R., Muruganand S. Smoke detection in video images using background subtraction method for early fire alarm system // 2017 2nd International Conference on Communication and Electronics Systems (ICCES). 2017. P. 167–171.
67. Islam M.R. et al. Smoke Object Segmentation and the Dynamic Growth Feature Model for Video-Based Smoke Detection Systems // Symmetry (Basel). 2020. Vol. 12, № 7.
68. Shuai L. et al. A novel smoke detection algorithm based on Fast Self-tuning background subtraction // 2016 Chinese Control and Decision Conference (CCDC). 2016. P. 3539–3543.
69. Filonenko A., Hernández D.C., Jo K.-H. Fast Smoke Detection for Video Surveillance Using CUDA // IEEE Trans. Ind. Informatics. 2018. Vol. 14, № 2. P. 725–733.
70. Dimitropoulos K., Tsalakanidou F., Grammalidis N. Flame Detection for Video-based Early Fire Warning Systems and 3D Visualization of Fire Propagation // Proc. IASTED Int. Conf. Comput. Graph. Imaging, CGIM 2012. 2012. P. 209–216.

71. Zhang Z. et al. An improved probabilistic approach for fire detection in videos // Springer. Kluwer Academic Publishers, 2014. Vol. 50, № 3. P. 745–752.
72. Avgerinakis K., ... A.B.-... workshop on multi, 2012 undefined. Smoke detection using temporal HOGHOF descriptors and energy colour statistics from video // researchgate.net. 2014.
73. Mueller M. et al. Optical flow estimation for flame detection in videos // ieeexplore.ieee.org.
74. Chen J., He Y., Wang J. Multi-feature fusion based fast video flame detection // Build. Environ. 2010. Vol. 45, № 5. P. 1113–1122.
75. Barmpoutis P., ... K.D.-2013 10th I., 2013 undefined. Real time video fire detection using spatio-temporal consistency energy // ieeexplore.ieee.org.
76. Emmy Prema C., Vinsley S.S., Suresh S. Efficient Flame Detection Based on Static and Dynamic Texture Analysis in Forest Fire Detection // Fire Technol. 2018. Vol. 54, № 1. P. 255–288.
77. Tao H., Lu X. Smoke vehicle detection based on multi-feature fusion and hidden Markov model // J. Real-Time Image Process. Springer, 2020. Vol. 17, № 3. P. 745–758.
78. Wu X. et al. An adaptive threshold deep learning method for fire and smoke detection // ieeexplore.ieee.org.
79. Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks // Adv. Neural Inf. Process. Syst. 2012. № 25. P. 1097–1105.
80. Sharma J. et al. Deep convolutional neural networks for fire detection in images // Commun. Comput. Inf. Sci. Springer Verlag, 2017. Vol. 744. P. 183–193.
81. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // arXiv Prepr. arXiv1409.1556. 2014.
82. He K. et al. Deep residual learning for image recognition // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2016. Vol. 2016-Decem. P. 770–778.

83. Zhang Q. et al. Deep convolutional neural networks for forest fire detection // atlantis-press.com. 2016.
84. Muhammad K. et al. Convolutional neural networks based fire detection in surveillance videos // ieeexplore.ieee.org.
85. Szegedy C. et al. Going deeper with convolutions // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. P. 1–9.
86. Muhammad K., Ahmad J., Baik S.W. Early fire detection using convolutional neural networks during surveillance for effective disaster management // Neurocomputing. Elsevier B.V., 2018. Vol. 288. P. 30–42.
87. Dunning A., international T.B.-2018 25th I., 2018 undefined. Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection // ieeexplore.ieee.org.
88. Sousa M.J., Moutinho A., Almeida M. Wildfire detection using transfer learning on augmented datasets // Expert Syst. Appl. Elsevier Ltd, 2020. Vol. 142.
89. Krizhevsky A., ... I.S.-A. in neural, 2012 undefined. Imagenet classification with deep convolutional neural networks // proceedings.neurips.cc.
90. Toulouse T. et al. Computer vision for wildfire research: An evolving image dataset for processing and analysis // Fire Saf. J. Elsevier Ltd, 2017. Vol. 92. P. 188–194.
91. Barmpoutis P. et al. Fire detection from images using faster R-CNN and multidimensional texture analysis // ieeexplore.ieee.org.
92. Lin G. et al. Smoke Detection on Video Sequences Using 3D Convolutional Neural Networks // Fire Technol. Springer New York LLC, 2019. Vol. 55, № 5. P. 1827–1847.
93. Jadon A. et al. FireNet: a specialized lightweight fire & smoke detection model for real-time IoT applications // arxiv.org.
94. Dai J. et al. R-FCN: Object detection via region-based fully convolutional networks // Advances in Neural Information Processing Systems. Neural information processing systems foundation, 2016. P. 379–387.
95. Girshick R. et al. Rich feature hierarchies for accurate object detection and

semantic segmentation // Proceedings of the IEEE conference on computer vision and pattern recognition. 2014. P. 580–587.

96. Girshick R. Fast R-CNN // 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2015. P. 1440–1448.

97. He K. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition // IEEE Trans. Pattern Anal. Mach. Intell. IEEE, 2015. Vol. 37, № 9. P. 1904–1916.

98. Lin T.-Y. et al. Focal Loss for Dense Object Detection // 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017. Vol. 42, № 2. P. 2999–3007.

99. Liu W. et al. SSD: Single shot multibox detector // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, 2016. Vol. 9905 LNCS. P. 21–37.

100. Redmon J. et al. You Only Look Once: Unified, Real-Time Object Detection.

101. Uijlings J.R.R. et al. Selective search for object recognition // Int. J. Comput. Vis. Springer, 2013. Vol. 104. P. 154–171.

102. Ren S. et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

103. Neubeck A., Van Gool L. Efficient non-maximum suppression // 18th International Conference on Pattern Recognition (ICPR'06). IEEE, 2006. Vol. 3. P. 850–855.

104. Bodla N. et al. Soft-NMS--improving object detection with one line of code // Proceedings of the IEEE international conference on computer vision. 2017. P. 5561–5569.

105. Solovyev R., Wang W., Gabruseva T. Weighted boxes fusion: Ensembling boxes from different object detection models // Image Vis. Comput. Elsevier, 2021. Vol. 107. P. 104117.

106. Nguyen H. Van, Pham T.X., Le C.N. Real-time long short-term glance-based

fire detection using a CNN-LSTM neural network // *Int. J. Intell. Inf. Database Syst.* Inderscience Publishers (IEL), 2021. Vol. 14, № 4. P. 349–364.

107. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network // *Phys. D Nonlinear Phenom.* Elsevier, 2020. Vol. 404. P. 132306.

108. Shi X. et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting // *Adv. Neural Inf. Process. Syst.* 2015. Vol. 28.

109. Liu W. et al. SSD: Single Shot MultiBox Detector // *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*). Springer Verlag, 2016. Vol. 9905 LNCS. P. 21–37.

110. Tan M., Pang R., Le Q. V. EfficientDet: Scalable and efficient object detection // *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 2020. P. 10778–10787.

111. Tan M., Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks // *International conference on machine learning.* PMLR, 2019. P. 6105–6114.

112. Tan M., Pang R., Le Q. V. EfficientDet: Scalable and Efficient Object Detection. 2019.

113. Sandler M. et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks // *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* IEEE, 2018. P. 4510–4520.

114. Szegedy C. et al. Rethinking the inception architecture for computer vision // *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016. P. 2818–2826.

115. TensorFlow [Electronic resource]. URL: <https://www.tensorflow.org/> (accessed: 04.11.2021).

116. PyTorch [Electronic resource]. URL: <https://pytorch.org/> (accessed: 10.05.2022).

117. models/research/object_detection at master · tensorflow/models · GitHub

[Electronic resource]. URL: https://github.com/tensorflow/models/tree/master/research/object_detection (accessed: 10.05.2022).

118. Weights & Biases – Developer tools for ML [Electronic resource]. URL: <https://wandb.ai/site> (accessed: 10.05.2022).

119. Nevada Seismological Laboratory [Electronic resource]. URL: <https://www.youtube.com/user/nvseismolab/about> (accessed: 02.12.2020).

120. Nevada Seismological Laboratory, University of Nevada [Electronic resource]. URL: <http://www.seismo.unr.edu/> (accessed: 02.12.2020).

121. Wildfire Observers and Smoke Recognition [Electronic resource]. URL: <http://wildfire.fesb.hr> (accessed: 19.11.2020).

122. Perm forest fire center [Electronic resource]. URL: https://www.youtube.com/channel/UCsKn1hQgGh5n7NGoqLNoh_Q/videos (accessed: 19.11.2020).

123. Cameras from various HPWREN related sites [Electronic resource]. URL: <http://hpwren.ucsd.edu/cameras/> (accessed: 16.04.2022).

124. Supervisely - Web platform for computer vision. Annotation, training and deploy [Electronic resource]. URL: <https://supervise.ly/> (accessed: 02.12.2020).

125. Dewangan A. et al. FIgLib & SmokeyNet: Dataset and Deep Learning Model for Real-Time Wildland Fire Smoke Detection // Remote Sens. MDPI, 2022. Vol. 14, № 4. P. 1007.

126. Baldota S. et al. Multimodal Wildland Fire Smoke Detection // arXiv Prepr. arXiv2212.14143. 2022.

127. Car Object Detection | Kaggle [Electronic resource]. URL: <https://www.kaggle.com/datasets/sshikamaru/car-object-detection> (accessed: 09.01.2023).

128. DroneDetectionThesis/Drone-detection-dataset: Dataset containing IR, visible and audio data to be used to train drone detection systems. [Electronic resource]. URL: <https://github.com/DroneDetectionThesis/Drone-detection-dataset> (accessed:

09.01.2023).

129. Moving Obstacle Detection in Highly Dynamic Scenes [Electronic resource]. URL: <https://data.vision.ee.ethz.ch/cvl/aess/dataset/> (accessed: 09.01.2023).

130. Патент на изобретение № 2721769. Стенд для контроля контурных перемещений гибкого манипулятора/ Лаптев Н.В., Колпащиков Д.Ю., Гергет О.М. и др. // Оpubл. 22.05.2020, Бюл. № 15. Заявка № 2019127060 от 28.08.2019 г. Решение о выдаче патента от 28.08.2019.

131. Manakov R.A. et al. Visual shape and position sensing algorithm for a continuum robot // IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021. Vol. 1019, № 1. P. 12066.

132. Данилов В.В., Колпащиков Д.Ю., Лаптев Н.В. Автоматическое управление непрерывным роботом с использованием алгоритма FABRIK // Моделирование, оптимизация и информационные технологии, 2019. Vol. 7, № 4. P. 1–2.

133. Лаптев Н.В., Кравченко А.А., Гергет О.М. Разработка библиотеки планирования и управления движениями коллаборативного робота Kuka iiwa // Информационные и математические технологии в науке и управлении. Федеральное государственное бюджетное учреждение науки, 2022. № 2 (26). P. 161–168.

ПРИЛОЖЕНИЕ А

Акты внедрения результатов диссертационной работы

TOMSK
POLYTECHNIC
UNIVERSITY



ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)



"УТВЕРЖДАЮ"

Директор ИШИТР, к.т.н.,

А.С. Фадеев

» _____ 2023г.

Акт

о внедрении в учебный процесс результатов
диссертации Лаптева Н.В. на тему: "Обнаружение объектов видеоряда на основе
технологии машинного обучения", представленной на соискание ученой степени
кандидата технических наук

Комиссия в составе: председателя – руководителя отделения информационных технологий на правах кафедры Инженерной школы информационных технологий и робототехники (ИШИТР), к.т.н., доцента Шерстнева В.С., и членов - к.т.н., доцента Погребного А.В., и к.т.н., доцента Кочегурова А.И. - составили настоящий акт в том, что результаты диссертационной работы Лаптева Н.В.:

- метод обнаружения объектов видеоряда с динамическими признаками;
- алгоритм объединения предсказаний с видеопоследовательности;
- программное обеспечение обнаружения объектов в видеоряде на основе машинного обучения;

используются при проведении практических и лабораторных занятий по дисциплинам «Технология обработки информации» и «Теория распознавания образов» для магистрантов направлений 09.04.01 «Информатика и вычислительная техника» и 09.04.02 «Информационные системы и технологии» в Томском политехническом университете. Результаты обеспечивают возможность применения инновационных технологий преподавания и повышают качество учебного процесса.

Руководитель отделения
информационных технологий на
правах кафедры ИШИТР

Шерстнев В.С.

Доцент ОИТ

Погребной А.В.

Доцент ОИТ

Кочегуров А.И.

"УТВЕРЖДАЮ"

Генеральный директор, к.т.н.

А.Х. Оздиев

" " 2023 г.

А К Т
об внедрении результатов
кандидатской диссертации Лаптева Н.В. в IT-компанию ООО
"Спотпаркинг"

Комиссия в составе: Челпанов М. С., менеджер ООО "Спотпаркинг", Чернова С. А, бэкенд-разработчик ООО "Спотпаркинг", Гмырь В. ML-инженер ООО "Спотпаркинг" – составили настоящий акт о том, что следующие результаты диссертационной работы Лаптева Н.В. используются в мобильном приложении для поиска свободных парковочных мест автомобилей:

- метод выделения объектов на изображениях видеопоследовательности
- алгоритм объединения предсказаний с последовательности изображений

Метод выделения объектов на изображениях видеопоследовательности позволяет выполнить оценку занятости парковочного пространства, посредством классификации на более высоком уровне за счет смещения разметки парковочных мест с привязкой к временной составляющей.

Менеджер



Челпанов М. С.

Бэкенд-разработчик

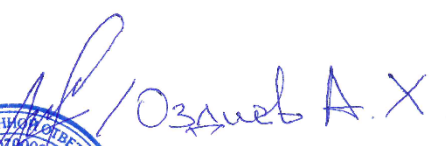


Чернова С. А.

ML-инженер



Гмырь В.

«УТВЕРЖДАЮ»

Директор ООО «ИТП КИБЕРЦЕНТР»

Плещков А.Г.

2023 г.



А К Т
об внедрении результатов
кандидатской диссертации Лаптева Н.В.
в ООО «ИТП КИБЕРЦЕНТР»

Комиссия в составе: Осипенко Д.В., инженер-программист, Погребной В.Ю., инженер, Анохин А.Ю., инженер – составили настоящий акт о том, что результаты диссертационной работы Лаптева Н.В.:

- модель нейронной сети для обнаружения пожароопасных объектов;
- алгоритм объединения предсказаний с последовательности изображений;
- модель гибридной нейронной сети для классификации данных предсказаний.

используются в ПО: «Программное обеспечение информационно-телекоммуникационной системы интеллектуального видеонаблюдения и аналитики лесопожарной обстановки» (ИТС «ИВА»)(Включено в Единый реестр российских программ для электронных вычислительных машин и баз данных в соответствии с Приказом Минкомсвязи РФ от 12.07.2021 №710 (Приложение 1, №пп.14, реестровый №242585) для обнаружения возгораний на ранней стадии в автоматическом режиме с выдачей тревожных сигналов оператору. Применение системы позволило автоматизировать процесс обнаружения и значительно снизить нагрузку по мониторингу леса и обнаружению возгораний.

Инженер-программист

Инженер

Инженер

Д.В. Осипенко

В.Ю. Погребной

А.Ю. Анохин

ПРИЛОЖЕНИЕ Б

Свидетельства о регистрации интеллектуальной деятельности

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022611227

**Обнаружение пожароопасных объектов в лесном
массиве на основе динамических признаков**

Правообладатель: *федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский
политехнический университет» (RU)*

Авторы: *Лаптев Никита Витальевич (RU), Лаптев
Владислав Витальевич (RU), Гергет Ольга Михайловна
(RU)*



Заявка № 2022610413

Дата поступления 13 января 2022 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 21 января 2022 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022619805

Программа для ЭВМ «Многофункциональный мобильный комплекс мониторинга, ситуационного анализа и оповещения труднодоступных объектов с использованием беспилотных летательных аппаратов и ГИС-технологий (ММК МАО)»

Правообладатель: *Общество с ограниченной ответственностью «ИНКОМ» (RU)*

Авторы: *Сонькин Дмитрий Михайлович (RU), Плешков Алексей Геннадьевич (RU), Изумнов Артём Олегович (RU), Осипенко Денис Витальевич (RU), Слезенко Николай Сергеевич (RU), Погребной Вадим Юрьевич (RU), Волегова Любовь Владимировна (RU), Кравченко Андрей Александрович (RU), Лаптев Никита Витальевич (RU), Лаптев Владислав Витальевич (RU), Колпащиков Дмитрий Юрьевич (RU), Гергет Ольга Михайловна (RU), Данилов Вячеслав Владимирович (RU)*



Заявка № 2022618392

Дата поступления 05 мая 2022 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 26 мая 2022 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022684945

Объединение прогнозов обнаруженных областей
задымления

Правообладатель: *федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский политехнический университет» (RU)*

Авторы: *Лаптев Никита Витальевич (RU), Гергет Ольга Михайловна (RU), Колпащиков Дмитрий Юрьевич (RU), Кравченко Андрей Александрович (RU), Лаптев Владислав Витальевич (RU)*



Заявка № 2022685130

Дата поступления 20 декабря 2022 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 20 декабря 2022 г.

Руководитель Федеральной службы
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b80077e14c40f0a94edbd24145d5c7
Владелец: **Зубов Юрий Сергеевич**
Действителен с 2.03.2022 по 26.05.2023

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



ПАТЕНТ

НА ИЗОБРЕТЕНИЕ

№ 2721769

СТЕНД ДЛЯ КОНТРОЛЯ КОНТУРНЫХ ПЕРЕМЕЩЕНИЙ
ГИБКОГО МАНИПУЛЯТОРА

Патентообладатели: *федеральное государственное автономное образовательное учреждение высшего образования "Национальный исследовательский Томский политехнический университет", (RU)*

Авторы: *Лантев Никита Витальевич (RU), Колпащиков Дмитрий Юрьевич (RU), Гергет Ольга Михайловна (RU), Манаков Роман Аркадьевич (RU), Кравченко Андрей Александрович (RU), Данилов Вячеслав Владимирович (RU), Скирневский Игорь Петрович (RU)*

Заявка № 2019127060

Приоритет изобретения 28 августа 2019 г.

Дата государственной регистрации в

Государственном реестре изобретений

Российской Федерации 22 мая 2020 г.

Срок действия исключительного права

на изобретение истекает 28 августа 2039 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

ПРИЛОЖЕНИЕ Г

Фрагмент исходного кода основных модулей программного комплекса

```

import json
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
import tensorflow as tf
import time
import logging
from network_impl import Net
logging.getLogger('tensorflow').disabled = True
import cv2
import wandb
import numpy as np
import pandas as pd
from ensemble_boxes import *
from tqdm import tqdm
from os import listdir
from os.path import splitext
from typing import List
from union_boxes import main
def bb_intersection_over_union(boxA, boxB):
    xA = max(boxA[0], boxB[0])
    yA = max(boxA[1], boxB[1])
    xB = min(boxA[2], boxB[2])
    yB = min(boxA[3], boxB[3])
    interArea = abs(max((xB - xA), 0)) * max((yB - yA), 0)
    if interArea == 0:
        return 0
    boxAArea = abs((boxA[2] - boxA[0]) * (boxA[3] - boxA[1]))
    boxBArea = abs((boxB[2] - boxB[0]) * (boxB[3] - boxB[1]))
    iou = interArea / float(boxAArea + boxBArea - interArea)
    return iou

```

```

def prepare_data_for_post_proc(full_boxes, full_scores, s): # Отсечка данных по трешхолду
    boxes_to_process = []
    scores_to_process = []
    for box, score in zip(full_boxes, full_scores):
        if score > s:
            boxes_to_process.append(np.array(box).tolist())
            scores_to_process.append(score)
    lb_list = [0] * len(scores_to_process)
    return boxes_to_process, scores_to_process, lb_list

def draw_boxes(frame, full_boxes, full_scores, boxes, gr_boxes, conf_score, s, sz, conf_lstm_cutoff): #
Функция рисования боксов
    negative = frame.copy()
    for box, score in zip(full_boxes, full_scores): # Закомментировать для скрытия найденных боксов
        if (score > s):
            x_min = box[1] * sz[0]
            y_min = box[0] * sz[1]
            x_max = box[3] * sz[0]
            y_max = box[2] * sz[1]
            txt_size = cv2.getTextSize("{:.2f}".format(score), cv2.FONT_HERSHEY_SIMPLEX, 0.6, 2)
            negative = cv2.rectangle(negative, (int(x_min), int(y_min) - txt_size[0][1] - 5),
                                    (int(x_min) + txt_size[0][0] + 5, int(y_min)),
                                    (0, 255, 255), -1)
            negative = cv2.rectangle(negative, (int(x_min), int(y_min)), (int(x_max), int(y_max)),
                                    (0, 255, 255),
                                    2)
            negative = cv2.putText(negative, "{:.2f}".format(score),
                                   (int(x_min) + 2, int(y_min) - 2),
                                   cv2.FONT_HERSHEY_SIMPLEX, 0.6,
                                   (0, 0, 0), 2)
    for box, score in zip(boxes, conf_score):
        if (score > conf_lstm_cutoff):
            negative = cv2.rectangle(

```



```

        negative,
        (int(box[0]), int(box[1])), (int(box[2]), int(box[3])),
        (255, 0, 10),
        2
    )
    txt_size = cv2.getTextSize("{:.2f}".format(score), cv2.FONT_HERSHEY_SIMPLEX, 0.6, 2)
    negative = cv2.rectangle(negative, (int(box[0]), int(box[1]) - txt_size[0][1] - 5),
                             (int(box[0]) + txt_size[0][0] + 5, int(box[1])),
                             (255, 0, 10), -1)
    negative = cv2.putText(negative, "{:.2f}".format(score),
                           (int(box[0]) + 2, int(box[1]) - 2),
                           cv2.FONT_HERSHEY_SIMPLEX, 0.6,
                           (0, 0, 0), 2)
else:
    negative = cv2.rectangle(
        negative,
        (int(box[0]), int(box[1])), (int(box[2]), int(box[3])),
        (200, 0, 255),
        2)
    txt_size = cv2.getTextSize("{:.2f}".format(score), cv2.FONT_HERSHEY_SIMPLEX, 0.6, 2)
    negative = cv2.rectangle(negative, (int(box[0]), int(box[1]) - txt_size[0][1] - 5),
                             (int(box[0]) + txt_size[0][0] + 5, int(box[1])),
                             (200, 0, 255), -1)
    negative = cv2.putText(negative, "{:.2f}".format(score),
                           (int(box[0]) + 2, int(box[1]) - 2),
                           cv2.FONT_HERSHEY_SIMPLEX, 0.6,
                           (0, 0, 0), 2)
for box in gr_boxes: # закомментировать for для отключения истинных боксов
    negative = cv2.rectangle(
        negative,
        (int(box[0]), int(box[1])), (int(box[2]), int(box[3])),

```

```

        (0, 0, 255),
        2 )

    return negative

def check_images_smoke(model,
                        score_thresh: float,
                        frame: List[float]) -> (bool, List[float], List[float]):
    predictions = model(np.array([frame], dtype=np.uint32))
    boxes = predictions['detection_boxes'].numpy()
    boxes = np.squeeze(boxes)
    scores = predictions['detection_scores'].numpy()
    scores = np.squeeze(scores)
    flag = False
    res_boxes, res_scores = [], []
    for score, box in zip(scores, boxes):
        if score >= score_thresh:
            flag = True
            res_boxes.append(box)
            res_scores.append(score)
    return flag, np.array(res_boxes), np.array(res_scores), frame

def convert_nn_out_to_coords(boxes, sz):
    for i in range(0, len(boxes)):
        temp_box = boxes[i].copy()
        temp_box[1] *= sz[0]
        temp_box[0] *= sz[1]
        temp_box[3] *= sz[0]
        temp_box[2] *= sz[1]
        boxes[i][0] = temp_box[1]
        boxes[i][1] = temp_box[0]
        boxes[i][2] = temp_box[3]
        boxes[i][3] = temp_box[2]

```

```
return boxes
```

```
def predict_confirm(model, image_for_predict, bbox, conf):
```

```
    if model == None:
```

```
        return None
```

```
    conf_class = []
```

```
    for box, scor in zip(bbox, conf):
```

```
        test_im = []
```

```
        i = 0
```

```
        for img_src in image_for_predict:
```

```
            frame = img_src[int(box[1]):int(box[3]), int(box[0]):int(box[2])]

```

```
            frame = cv2.resize(frame, (300, 300), cv2.INTER_CUBIC)

```

```
            frame = frame / 255.0

```

```
            if ( i < 4):
```

```
                test_im.append(frame)

```

```
                i += 1

```

```
        test_im = np.expand_dims(test_im, axis=0)

```

```
        model_output = model.predict(test_im)

```

```
        conf_class.append(model_output[0][0])

```

```
    return conf_class
```

```
def test_on_union(use_algoritm_union_boxes, ODPredict, model_class = None): # Тестирование выборки и
объединение боксов
```

```
    list = ['threshold', 'TP', 'FP', 'FN', 'TN', 'Precision', 'Recall', 'F-score', 'Accuracy', 'mIOU']

```

```
    list_per_video_stats = ['threshold', 'TP', 'FP', 'FN', 'TN', 'IOU']

```

```
    csvData = pd.DataFrame(columns=list)

```

```
    conf_lstm_cutoff = 0.8

```

```
    for s in tqdm(range(5, 60, 5), desc=f"Process various cut-off rates on
{dict_box_processing[use_algoritm_union_boxes]} post processing"):

```

```
        s /= 100

```

```
        A, FN, TP, FP, Precision, Recall, F1 = 0, 0, 0, 0, 0, 0, 0

```

```
        TN = 0

```

```

boxes_array = 0
mIOU = 0.0

for videoData in ODPredict:

    full_boxes, full_scores, video_path, ann_path, frames_idx, sz, frames, gr_boxes = videoData

    # Формируем массив боксов

    if use_algorithm_union_boxes == 1: # Вставить Димин код (Union_box)

        obj_1, obj_2, obj_3, obj_4, obj_5 = main([

            [
                scor,
                0.5,
                box[1] * sz[0],
                box[0] * sz[1],
                box[3] * sz[0],
                box[2] * sz[1],
                0.0,
                0.0,
            ] for box, scor in zip(full_boxes, full_scores)],

            s)

        if len(obj_1) > 0:

            boxes = [[x1, y1, x2, y2] for x1, y1, x2, y2 in zip(obj_1, obj_2, obj_3, obj_4)]

        else:

            boxes = []

    elif use_algorithm_union_boxes == 2:

        boxes_to_process, scores_to_process, lb_list = prepare_data_for_post_proc(full_boxes, full_scores,

        if len(scores_to_process) > 0:

            boxes, _, _ = nms([boxes_to_process], [scores_to_process], [lb_list], weights=[1], iou_thr=0.15)

        else:

            boxes = []

    elif use_algorithm_union_boxes == 3:

        boxes_to_process, scores_to_process, lb_list = prepare_data_for_post_proc(full_boxes, full_scores,

        sigma = 0.1

```

```

skip_box_thr = 0.0001
if len(scores_to_process) > 0:
    boxes, _, _ = soft_nms([boxes_to_process], [scores_to_process], [lb_list], weights=[1],
                           iou_thr=0.15, sigma=sigma, thresh=skip_box_thr)
else:
    boxes = []
elif use_algorithm_union_boxes == 4:
    skip_box_thr = 0.0001
    boxes_to_process, scores_to_process, lb_list = prepare_data_for_post_proc(full_boxes, full_scores,
    if len(scores_to_process) > 0:
        boxes, _, _ = weighted_boxes_fusion([boxes_to_process], [scores_to_process], [lb_list],
        weights=[1], iou_thr=0.15, skip_box_thr=skip_box_thr)
    else:
        boxes = []
else:
    boxes_to_process, _, _ = prepare_data_for_post_proc(full_boxes, full_scores, s)
    boxes = convert_nn_out_to_coords(boxes_to_process.copy(), sz)
if use_algorithm_union_boxes >= 2:
    boxes = convert_nn_out_to_coords(boxes, sz)
true_boxes = []
iou_arr = []
full_scores_class = predict_confirm(model_class, frames, boxes, full_scores)
check = False
for gr_box in gr_boxes:
    if sum(gr_box) != 0:
        for box, score_lstm in zip(boxes, full_scores_class):
            y_true = [gr_box[0], gr_box[1], gr_box[2], gr_box[3]]
            y_pred = [box[0], box[1], box[2], box[3]]
            if bb_intersection_over_union(y_true, y_pred) >= iou or bb_intersection_over_union(y_pred,
y_true) >= iou:
                if score_lstm > conf_lstm_cutoff:
                    true_boxes.append(box)

```

```

        check = True

        iou_arr.append(bb_intersection_over_union(y_true, y_pred))

    if check is not True and len(gr_boxes) > 0:

        FN += 1

        if s == 0.05:

            negative = draw_boxes(frames[1], full_boxes, full_scores, boxes, gr_boxes, full_scores_class, s, sz,
                                conf_lstm_cutoff)

            if not os.path.exists(

                f'metrics/{name}/FN_detection_{dict_box_processing[use_algorithm_union_boxes]}'):

                os.makedirs(f'metrics/{name}/FN_detection_{dict_box_processing[use_algorithm_union_boxes]}')

            cv2.imwrite(f'metrics/{name}/FN_detection_{dict_box_processing[use_algorithm_union_boxes]}/{os.path.basename(ann_path)}.png', negative)

            if float(args.show_all_boxes) > 0:

                s_t = float(args.show_all_boxes)

                if abs(s_t - s) < 0.005:

                    negative = draw_boxes(frames[1], full_boxes, full_scores, boxes, gr_boxes, full_scores_class, s, sz,
                                        conf_lstm_cutoff)

                    if len(gr_boxes) == 0 and len(boxes) == 0:

                        check = True

                    negative = cv2.putText(negative, str(check),

                                            (0, 15),

                                            cv2.FONT_HERSHEY_SIMPLEX, 0.6,

                                            (0, 255, 255), 1)

                    if not os.path.exists(

                        f'metrics/{name}/ALL_detection_{dict_box_processing[use_algorithm_union_boxes]}'):

                            os.makedirs(f'metrics/{name}/ALL_detection_{dict_box_processing[use_algorithm_union_boxes]}')

                    cv2.imwrite(

                        f'metrics/{name}/ALL_detection_{dict_box_processing[use_algorithm_union_boxes]}/{os.path.basename(ann_path)}.png',

                            negative)

            if len(iou_arr) > 0:

                mIOU += np.mean(iou_arr)

```

```

    boxes_array += len(iou_arr)
if len(true_boxes) > 0:
    TP += 1
FP += len([x for x in full_scores_class if x >= conf_lstm_cutoff]) - len(true_boxes)
if len(gr_boxes) == 0:
    if len([x for x in full_scores_class if x >= conf_lstm_cutoff]) == 0:
        TN += 1

# Считаем метрики
if boxes_array > 0:
    mIOU = mIOU / boxes_array
else:
    mIOU = 0
try:
    Precision = TP / (TP + FP)
except:
    Precision = 0
try:
    Recall = TP / (TP + FN)
except:
    Recall = 0
try:
    F1 = 2 * Precision * Recall / (Precision + Recall)
except:
    F1 = 0
try:
    A = (TP + TN) / (TP + TN + FP + FN)
except:
    A = 0
log_csv = pd.DataFrame([
    [
        s,

```

```

    TP,
    FP,
    FN,
    TN,
    Precision,
    Recall,
    F1,
    A,
    mIOU
]
],
    columns=list)

csvData = csvData.append(log_csv, ignore_index=True)

csvData.to_csv(f'metrics/{name}/metrics_{dict_box_processing[use_aloritm_union_boxes]}.csv')

import argparse

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument('-u', '--union_boxes', default=1, type=int)
    parser.add_argument('-e', '--encoder', default='efficientdet_d0', type=str)
    parser.add_argument('-t', '--type', default=3, type=int)
    parser.add_argument('-c', '--img_count', default=4, type=int)
    parser.add_argument('-s', '--show_all_boxes', default='0.05', type=str)
    args = parser.parse_args()
    wandb_config = dict(union_boxes=args.union_boxes,
                        encoder=args.encoder,
                        type=args.type,
                        img_count=args.img_count,)
    dict_type = {0: 'none', 1: 'difference', 2: 'gauss', 3: 'gray'}
    dict_box_processing = {0: 'none', 1: 'union', 2: 'nms', 3: 'soft-nms', 4: 'weighted-boxes-fusion'}
    name = f'boxes_img_{args.img_count}_{args.encoder}_smoke_detection_{dict_type[args.type]}'#
    dict_type[args.type]

    if not os.path.exists(f'metrics/{name}'):

```



```

    os.makedirs(f'metrics/{name}')

from datetime import datetime

today = datetime.today()

video_list_raw = listdir('DataForTesting/video') #
video_list = []

for pathToVideo in video_list_raw:

    pwd = os.path.splitext(pathToVideo)[0]

    video_list.append(pwd)

video_list.sort()

if args.type == 1:

    preprocessing = 'difference'

elif args.type == 2:

    preprocessing = 'gausse'

elif args.type == 3:

    preprocessing = 'gray'

else:

    preprocessing = None

model_dir =
f'Smoke_detection/inference_models_new/{args.encoder}_smoke_detection_{args.type}/saved_model'

video_dir = 'DataForTesting/video'

ann_dir = 'DataForTesting/ann'

frame_count_on_video = 7

iou = 0.12

print(f'\033[32mInit model {name}')

model = tf.saved_model.load(model_dir)

inference_time = []

images_predict_count = args.img_count

use_algoritm_union_boxes = args.union_boxes

ODPredict = None

start = time.time()

inference_time.append(start)

for idx_, video_name in enumerate(tqdm(video_list, desc="Process video...")):

```

```

video_path = os.path.join(video_dir, video_name + '.mp4')
ann_path = os.path.join(ann_dir, video_name + '.mp4.json')
if os.path.exists(video_path) and os.path.exists(ann_path):
    # Получаем кадры из видео
    images = []
    frame_for_show = []
    cap = cv2.VideoCapture(video_path)
    sz = (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
          int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    frames_idx = np.array(np.linspace(0, frame_count - frame_count_on_video, frame_count_on_video),
dtype=int)
    if preprocessing == 'gray':
        MFrame = np.zeros((sz[1], sz[0]))
    else:
        MFrame = np.zeros((sz[1], sz[0], 3))
    counter_temp = 0
    for i, frame_idx in enumerate(frames_idx):
        cap.set(1, frame_idx)
        _, frame = cap.read()
        frame_for_show.append(frame.copy())
        if _ != False:
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            if preprocessing == 'gausse':
                frame = cv2.GaussianBlur(frame, (5, 5), 0)
            if preprocessing == 'gray':
                frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                frame = cv2.GaussianBlur(frame, (5, 5), 0)
            MFrame = MFrame + frame
            images.append(frame)
    else:
        print(f'WARNING: {video_name} problem video')

```

```

frames_idx = np.delete(frames_idx, i)
MFrame = MFrame / len(images)
if images_predict_count == 1:
    images_predict = [
        images[3],
    ]
elif images_predict_count == 2:
    images_predict = [
        images[0],
        images[-1],
    ]
    frame_for_show = [
        frame_for_show[0],
        frame_for_show[-1]
    ]
elif images_predict_count == 3:
    images_predict = [
        images[0],
        images[3],
        images[-1],
    ]
    frame_for_show = [
        frame_for_show[0],
        frame_for_show[3],
        frame_for_show[-1]
    ]
elif images_predict_count == 4:
    images_predict = [
        images[0],
        images[2],
        images[4],

```

```

        images[-1],
    ]
    frame_for_show = [
        frame_for_show[0],
        frame_for_show[2],
        frame_for_show[4],
        frame_for_show[-1]
    ]
elif images_predict_count == 5:
    images_predict = [
        images[0],
        images[2],
        images[3],
        images[4],
        images[-1], ]
    frame_for_show = [
        frame_for_show[0],
        frame_for_show[2],
        frame_for_show[3],
        frame_for_show[4],
        frame_for_show[-1]
    ]
else:
    images_predict = []
    full_boxes = None
    full_scores = None
    counter_temp=0
    for img_pred in images_predict:
        if preprocessing is not None:
            img_pred = np.abs(MFrame - img_pred)
        if preprocessing == 'gray':

```

```

img_pred = cv2.cvtColor(img_pred.astype('uint8'), cv2.COLOR_GRAY2RGB)
_, boxes, scores, predict_images = check_images_smoke(model=model,
                                                         score_thresh=0.01,
                                                         frame=img_pred)

counter_temp += 1
inference_time.append(time.time())

if full_boxes is None:
    full_boxes = boxes
else:
    try:
        full_boxes = np.vstack((full_boxes, boxes))
    except:
        continue

if full_scores is None:
    full_scores = scores
else:
    try:
        full_scores = np.hstack((full_scores, scores))
    except:
        continue

# Ищем аннотированные боксы
gr_full_boxes = []

try:
    f = open(ann_path)
    ann_data = json.load(f)
    for id in frames_idx:
        ann_frame = ann_data['frames'][id]
        for fig in ann_frame['figures']:
            if fig['geometryType'] == 'rectangle':
                rectangle = fig['geometry']['points']['exterior']
                x1 = rectangle[0][0]

```

```

y1 = rectangle[0][1]
if rectangle[1][0] < x1:
    x1, x2 = rectangle[1][0], rectangle[0][0]
else:
    x2 = rectangle[1][0]
if rectangle[1][1] < y1:
    y1, y2 = rectangle[1][1], rectangle[0][1]
else:
    y2 = rectangle[1][1]
gr_full_boxes.append([x1, y1, x2, y2])
obj_1, obj_2, obj_3, obj_4, _ = main([
    [
        1.0,
        0.5,
        box[0],
        box[1],
        box[2],
        box[3],
        0.0,
        0.0,
    ] for box in gr_full_boxes],
    0)
gr_boxes = []
for x1, y1, x2, y2 in zip(obj_1, obj_2, obj_3, obj_4):
    gr_boxes.append([x1, y1, x2, y2])
except:
    gr_boxes = []
if ODPredict is None:
    ODPredict = [
        (full_boxes, full_scores, video_path, ann_path, frames_idx, sz, frame_for_show, gr_boxes)]
else:

```

```

try:
    ODPredict.append(
        (full_boxes, full_scores, video_path, ann_path, frames_idx, sz, frame_for_show, gr_boxes))
except:
    continue

for i in range(len(inference_time)-1, 0, -1):
    inference_time[i] -= inference_time[i-1]
if len(inference_time) > 1:
    inference_time[0] -= start
print(f'\033[32m inference time on {images_predict_count} images: {np.mean(inference_time[1:])}')
del model

Net = Net(None, 'adam', 0.0001, None) # Обращение к Network_impl
model = Net.load_model(
    '/media/viacheslav/Project/Smoke_klassification/models_classifai/smoke_classification#test_time=04-06-
22.53/')

if (use_algorithm_union_boxes == -1):
    for use_algorithm_union_boxes in range(0, 5):
        test_on_union(use_algorithm_union_boxes, ODPredict, model)
else:
    test_on_union(use_algorithm_union_boxes, ODPredict, model)
print("\033[32mAlgorithm complited!")

```