

**Федеральное государственное бюджетное учреждение науки  
Институт программных систем им. А.К. Айламазяна  
Российской академии наук**

Абрамов Н.С., Емельянова Ю.Г., Талалаев А.А., Фраленко В.П.,  
Хачумов В.М., Хачумов М.В., Шишкин О.Г., Шустова М.В.

**Методологическое обеспечение для решения задач обнаружения очагов  
возгорания на основе средств технического зрения в масштабе времени,  
близком к реальному**

*Препринт*

Переславль-Залесский  
2024

Абрамов Н.С., Емельянова Ю.Г., Талалаев А.А., Фраленко В.П., Хачумов В.М., Хачумов М.В., Шишкин О.Г., Шустова М.В. Методологическое обеспечение для решения задач обнаружения очагов возгорания на основе средств технического зрения в масштабе времени, близком к реальному. Препринт. ИПС им. А.К.Айламазяна РАН, 2024, 45 с.

Представлены основные результаты, полученные в ИПС им. А.К.Айламазяна РАН в рамках выполнения проекта РФ № 22-11-20001 (2022-2024гг) «Разработка алгоритмического и программного обеспечения интеллектуальной системы обнаружения очагов возгорания на основе средств технического зрения в масштабе времени, близком к реальному» и гранта в форме субсидии из областного бюджета организациям Ярославской области. Тематика исследований соответствует приоритетному направлению «Охрана окружающей среды; мониторинг и прогнозирование состояния биотехносферы региона», поддерживаемому Правительством Ярославской области. Принимая во внимание масштабы возгораний полей и лесов в регионах РФ, включая Ярославскую область в последние годы, созданы современные алгоритмы, методы и средства мониторинга состояния полей и лесных массивов в рамках интеллектуальной технологии, в том числе с целью предупреждения, раннего обнаружения и оперативного реагирования на возникновение возгораний и пожаров. Даны рекомендации по применению методологического обеспечения профильными организациями Ярославского региона, осуществляющие мониторинг лесных территорий.

Исследование выполнено за счет гранта Российского научного фонда № 22-11-20001, <https://rscf.ru/project/22-11-20001/> и гранта в форме субсидии из областного бюджета организациям Ярославской области.

Рецензенты: д.т.н., г.н.с. ФИЦ ИУ РАН Славин О.А.  
д.т.н., зав. каф. МГТУ им. Н.Э.Баумана Алфимцев А.Н.

Abramov N.S., Yemelyanova Yu.G., Talalaev A.A., Fralenko V.P., Khachumov V.M., Khachumov M.V., Shishkin O.G., Shustova M.V. Methodological support for solving problems of detecting fires based on technical vision tools in a time scale close to real. Preprint. PSI RAS, 2024, 45 p.

The main results obtained at the Ailamazyan Program Systems Institute of Russian Academy of Sciences are presented as part of the implementation of project RSF № 22-11-20001 (2022--2024) «Development of algorithmic and software for an intelligent system for detecting fire based on technical vision a close to realtime-scale» and a grant in the form of a subsidy from the regional budget to organizations in the Yaroslavl Region. The research topics correspond to the priority direction “Environmental protection; monitoring and forecasting the state of the region’s biotechnosphere”, supported by the Government of the Yaroslavl region. Taking into account the scale of fires in fields and forests in the regions of the Russian Federation, including the Yaroslavl region in recent years, modern algorithms, methods and tools for monitoring the condition of fields and forests have been created within the framework of intelligent technology, including for the purpose of warning, early detection and rapid response to fires. the occurrence of fires and fires. Recommendations are given on the use of methodological support by specialized organizations of the Yaroslavl region that monitor forest areas.

This research was funded by Russian science foundation, grant number 22-11-20001 (<https://rscf.ru/en/project/22-11-20001/>) and a grant in the form of a subsidy from the regional budget to organizations in the Yaroslavl region.

## СОДЕРЖАНИЕ

Введение .....	4
1. Повышение качества изображений видеопотока от систем технического зрения .....	7
1.1. Деконволюция потока данных от БПЛА .....	7
1.2. Оптическая стабилизация видеопотока .....	10
2. Решение задачи раннего обнаружения возгораний .....	11
2.1. Оценка возможностей архитектуры Yolo в задаче обработки данных .....	12
2.2. Задача бинарной классификации .....	16
2.3. Выделение прямоугольных областей с очагами пожаров .....	17
2.4. Выделение границ пожаров с точностью до пикселя .....	18
2.5. Алгоритм обнаружения линии горизонта .....	19
3. Задача обнаружения людей и транспортных средств рядом с очагом возгорания .....	28
4. Проведение экспериментальных исследований .....	30
4.1. Проведение экспериментальных исследований с отсечением горизонта .....	30
4.2. Проведение экспериментальных исследований с нейронными сетями .....	31
4.3. Примеры обнаружения транспортных средств и людей в видеопотоке .....	32
4.4. Экспериментальные исследования по распознаванию транспортных средств и их номеров .....	34
5. Рекомендации по применению методов обнаружения пожаров и задымлений .....	38
Заключение .....	39
Литература .....	40

## Введение

Лесные пожары являются распространенным явлением, причинами которого являются молнии, неосторожность человека и воздействие экстремальной жары и засушливости на топливо и легковоспламеняющиеся материалы. В связи со значительным ущербом природным и человеческим ресурсам необходимы схемы раннего предупреждения, поскольку начальный этап – лучшее время для тушения пожара. Этому вопросу посвящены многочисленные работы, в которых предлагаются различные подходы.

В работе [1] предложена концепция обнаружения пожара для выявления несчастных случаев, отслеживания и предотвращения лесного пожара на месте. Основу концепции составляет сбор данных с помощью датчиков и сетевом подключении и использование облачных сервисов. Предложена модель для прогнозирования и мониторинга лесного пожара во время стихийного бедствия, отслеживания порогового значения и информирования уполномоченное лица.

В статье [2] описана автоматическая система раннего предупреждения лесного пожара. Система использует беспилотный летательный аппарат (БПЛА), оснащенный множеством датчиков и алгоритм глубокого обучения. Датчики и алгоритм обрабатывают изображения в реальном времени с использованием предложенной модели глубокого обучения нейронной сети YOLO-подхода. YOLO-архитектуры отличаются тем, что за один шаг решают задачи выделения и классификации объектов, достигая этого тем, что обучаются поиску точных позиций целевых объектов в кадре, также возвращается информация о классе каждого объекта, значение величины уверенности искусственной нейронной сети (ИНС). Система обеспечивает хорошую точность определения местоположения лесного пожара в режиме реального времени, позволяет отслеживать лесной пожар и сообщать о нем соответствующему органу.

Другой алгоритм, на основе нейронной сети YOLO-V3, служит для обнаружения лесных пожаров на основе данных, собранных с БПЛА описан в статье [3]. Модель YOLO-V3 позволяет изучать мелкие объекты изображений пламени на этапе извлечения признаков в режиме реального времени. При этом, по сравнению с аналогами, повышается точность распознавания пламени на видеоизображении и снижается частота ошибок.

В статье [4] представлен подход к обнаружению огня на видеоданных. Зоны возможного возгорания обнаруживаются на основе низкоуровневых визуальных признаков (в общем случае это признаки, основывающиеся на цветовых и текстурных особенностях изображений, а также тем или иным способом описывающие форму целевых объектов). Используемая СНС с SPP-слоем (от англ. Spatial Pyramid Pooling) обучается на специальном наборе данных и позволяет точно идентифицировать выделенные зоны возгорания с разными соотношениями сторон и масштабами. Экспериментальные результаты показывают, что предложенный подход обеспечивает точную идентификацию пламени в разных масштабах; оценка точности (F-мера), приведенная авторами, равна 93.33%.

Новый подход для обнаружения лесных пожаров рассмотрен в статье [5]. Предлагается платформа, использующая искусственный интеллект. Используются методы компьютерного зрения для распознавания и обнаружения дыма и огня, основанные на неподвижных изображениях или видеосигналах, поступающих с камер. В этом исследовании использовались два алгоритма классификации, где первым реализован алгоритм KNN (*k-nearest neighbors algorithm*), который дает точность 98.4%, а затем для повышения точности реализован другой алгоритм случайного леса, который обеспечивает точность 99.6%. Затем используются методы обработки изображений для прогнозирования пожара и дыма на кадрах изображения при воспроизведении видео.

Отметим работу [6], в которой дан общий анализ мобильных и немобильных систем мониторинга и наблюдения для обнаружения лесных пожаров и задымлений. Отмечается, что имеется два основных подхода. Один из которых основан на машинном обучении, а другой – на извлечении цветовых характеристик изображений или видеок кадров.

Утверждается, что комбинация обоих методов используется для обнаружения дыма и пожара или только дыма, что гарантирует точность показателей более чем на 90%.

Исследования методов и алгоритмов обнаружения очагов огня на изображениях и видеопотоках, полученных средствами технического зрения, в частности, с БПЛА, были выполнены в работе [7].

Авторы аналитического обзора [8] описывают основы методов обработки изображений, особое место уделяя сверточным нейронным сетям (СНС) и перспективам их применения при решении задачи поиска огня на изображениях. Обсуждаются существующие наборы данных (датасеты) для обучения ИНС поиску огня и дыма. Авторы работы [9] оценивают различные комбинации современных архитектур глубокого обучения, функций потерь и типов обрабатываемых изображений, с целью определения параметров, оказывающих наибольшее влияние на результаты, получаемые при решении задачи семантической сегментации изображений. В обзоре [10] рассматриваются актуальные алгоритмы для обнаружения дыма, выполнена их общая классификация. Методы обработки изображений, включая подходы на основе машинного и глубокого обучения, были рассмотрены для каждого из трех типов постановки задачи: классификации изображений, сегментации изображений и задачи выделения регионов. Обсуждается производительность алгоритмов, наборы данных, используемые в исследованиях, методы оценки точности, задачи и будущие направления исследований. В исследовании [11] делается попытка проследить эволюцию подходов к решению задачи обнаружения задымления с применением методов машинного обучения, основанных на выделении характерных признаков движения, полупрозрачности, цвета, формы, текстуры и фрактальных особенностях. Предложена архитектура СНС с элементами LSTM-сети, демонстрирующая высокие показатели точности.

В работе [12] для поиска задымлений используется СНС с архитектурой CSPdarknet53 (на базе YOLO), также снабженная SPP-слоем. Оценка качества предлагаемого подхода достигает уровня 97.9% (F-мера), при этом скорость обработки данных достигает 32 кадров/с, что соответствует требованиям обнаружения в реальном времени. Алгоритм реального времени для обнаружения пожаров на основе СНС с архитектурой YOLO 5s изложен в работе [13]. Исследования показали, что модель может автоматически идентифицировать и обнаруживать пламя и дым на разных стадиях и в разных формах. Оценка точности mAP@.5 (от англ. mean Average Precision), приведенная авторами исследования, говорит об успешном обнаружении огня и задымления в 80.6% случаев, скорость обработки данных составляет 31 кадр/с. Величина mAP – основа выбора архитектуры ИНС для конкретной задачи. Очевидно, что нет смысла экспериментировать с ИНС, показавшими низкое значение mAP, то есть данный коэффициент вносит некоторое понимание в то, насколько хорошо ИНС решает поставленную задачу, поэтому он также будет применяться в данной работе для оценки полученных результатов.

Исследование [14] предлагает быстрый и эффективный метод обнаружения пожаров, основанный на использовании СНС с архитектурой MobileNetV3. Описанный метод показывает точность 90.2% на созданном авторами наборе данных, в то время как размер модели составляет всего 21.4 МБ, скорость работы может достигать 29.5 кадров/с. Результаты проведенного тестирования подтверждают высокую эффективность метода, возможность обнаружения пожаров в режиме реального времени и применимость в встраиваемых системах. Подход к распознаванию пожаров в кадре, основанный на использовании модели MobileNetV3 с глубоким обучением, предлагается в статье [15]. Экспериментальные результаты показывают, что количество параметров для предлагаемой архитектуры СНС составляет всего 2.64 млн. – по сравнению с YOLO 4 это сокращение почти на 95.87%. Подход показывает преимущества модели с меньшим количеством параметров, низкими требованиями к памяти и высокой скоростью вывода по сравнению с существующими алгоритмами. Подход, представленный в этой статье, был специально разработан для использования в качестве системы мониторинга на платформе БПЛА. В

статье [16] предлагается модель для обнаружения лесных пожаров, построенная на базе архитектуры YOLO 4, для уменьшения количества параметров и вычислений в исследовании предложено заменить «базовую сеть» YOLO 4, на сеть MobileNetV3. Результаты экспериментов показывают, что по сравнению с YOLO 4 количество параметров в предлагаемой архитектуре уменьшается на 62.78%, а скорость вывода увеличивается в 3.04 раза. При этом точность обнаружения предлагаемого алгоритма достигает 66.6% по шкале mAP@.5. Основная ценность представленного метода состоит в возможности реализации механизма обнаружения лесных пожаров и обнаружении объектов в режиме реального времени на устройствах с низкими вычислительными ресурсами.

Для определения пожаров в работе [17] предлагается проводить обработку с применением СНС, в которой полносвязный выходной слой заменен на GAP-слой (от англ. Global Average Pooling). Подобное изменение архитектуры СНС позволяет избежать эффекта переобучения сети и повышает скорость ее работы. При этом на входы ИНС подаются расширенные наборы цветовых и текстурных признаков обрабатываемых изображений. Набор входных признаков формируется за счет преобразования изображений в различные цветовые пространства и использования детектора границ Canny. Результаты тестирования показывают точность распознавания ситуации 89.92%.

В исследовании [18] продемонстрирован подход к совершенствованию процесса мониторинга с использованием БПЛА. Вместо того, чтобы полагаться на автономные компьютеры и трудоемкую постобработку изображений, авторы предлагают использовать распределенную кластерную систему и применять высокопроизводительные MapReduce-алгоритмы для обработки поступающих с БПЛА изображений. Работа MapReduce-алгоритма состоит из двух шагов: Map и Reduce. На Map-шаге происходит предварительная обработка входных данных, для этого один из компьютеров, называемый главным узлом, получает входные данные задачи, разделяет их на части и передает другим компьютерам для предварительной обработки. На Reduce-шаге происходит свертка предварительно обработанных данных: главный узел получает ответы от рабочих узлов и на их основе формирует результат – решение задачи, которая изначально формулировалась. С целью выделения на изображениях мест возгорания предлагается проводить обработку данных с использованием библиотеки HPI (от англ. Hadoop Image Processing Interface), вычисляющей «цветовое различие» (от англ. colordistance) между усредненным значением цвета эталонных и обрабатываемых изображений. Для вычисления «цветового различия» авторы предлагают использовать цветовое пространство CIELAB и метрику Евклида. Отмечается высокая скорость обработки изображений, однако данные о других качественных характеристиках не приводятся.

Авторы работы [19] адаптировали и оптимизировали методы глубокого обучения для обнаружения лесных пожаров на ранней стадии. Используются два метода типа «визуальный трансформер» (от англ. Vision Transformer, в литературе используется аббревиатура ViT) – соответственно, TransUNet и TransFire, и СНС EfficientSeg для сегментации областей лесных пожаров и точного определения их областей. «Визуальный трансформер» состоит из токенизатора, трансформера и проектора, более подробное описание принципов его работы можно найти в оригинальном исследовании [20]. Результаты показывают эффективность использования глубокого обучения и методов типа «визуальный трансформер» для классификации и сегментации кадров с лесными пожарами. Предложенная модель классификации имеет точность 85.12%. Лучшие модели семантической сегментации получили оценку F1 99.90% (для архитектуры TransUNet) и 99.82% (для архитектуры TransFire).

Как видно из обзора выше, задаче обнаружения пожаров посвящено достаточно много работ. Можно проследить тенденцию применения искусственных нейронных сетей в данной области, поскольку эти технологии обеспечивают автоматическое распознавание в режиме времени, близком к реальному и обладают высокой точности. Использование

ИНС может снизить влияние человеческого фактора и помочь лицу, принимающему решение, быстрее среагировать на возникший пожар.

Далее рассмотрим задачи предварительной обработки видеопотока, которые позволяют повысить качество распознавания областей интереса.

## **1. Повышение качества изображений видеопотока от систем технического зрения**

В настоящее время особое внимание уделяется анализу и интерпретации видеопотока от БПЛА, которые должны обеспечивать решение задач выделения и распознавания целевых объектов, позиционирования и слежения, выполняемых с борта БПЛА, в том числе как автономно, так и с Земли [21-26], для чего необходимо иметь четкие изображения без смазов, а кадры в обрабатываемых видеопоследовательностях должны сменять друг друга без резких смещений камеры. Несмотря на удачные аэродинамические и компоновочные решения, отечественные разработки по оснащению БПЛА пока проигрывают зарубежным в части качества бортовых систем технического зрения. Для обеспечения необходимых показателей требуется создание математического, алгоритмического и программного обеспечения, способного выполнять в режиме реального времени основные функции предобработки и улучшения поступающих от БПЛА кадров [27].

Для проведения экспериментов применялся процессор общего назначения Intel Core i3 8300 и графический ускоритель вычислений NVidia GTX 1080 Ti с 11 ГБ видеопамати. В качестве экспериментальных данных использовались снимки размером 1280x720 пикселей, полученные с мобильной камеры [28], максимально близкие к тем, что получаются от реальных систем технического зрения БПЛА.

### **1.1. Деконволюция потока данных от БПЛА**

На реальных изображениях от БПЛА присутствуют явные смазы, получившиеся из-за резкого смещения камеры во время получения того или иного кадра. При этом это смещение может быть нелинейным ввиду того, что движение БПЛА зависит от управляющих воздействий оператора, ветровой нагрузки, точности работы систем ориентирования на местности, в том числе Glonass и GPS. Все эти факторы могут накладываться на БПЛА одновременно, что приводит к тому, что положение камеры в пространстве становится нестабильным. Кроме того, объекты в процессе получения одного и того же кадра тоже могут двигаться, например, это автомобили, велосипеды, пешеходы, животные. То есть камера может некачественно фиксировать объект в кадре из-за 1) быстрого движения камеры или самого объекта; 2) неоптимальных настроек камеры (например, поставлена слишком большая выдержка, матрица очень медленная или перегрета). Деконволюция (от англ. «deconvolution») – способ компенсации таких факторов, приводящих к тому, что кадры видеопотока становятся смазанными. Предлагается использовать для деконволюции методы математической статистики и искусственные нейронные сети, обучаемые с помощью общедоступных датасетов.

Примеры обрабатываемых изображений приведены на рис.1. Предлагается использовать для деконволюции методы математической статистики и ИНС, обучаемые с помощью общедоступных датасетов.

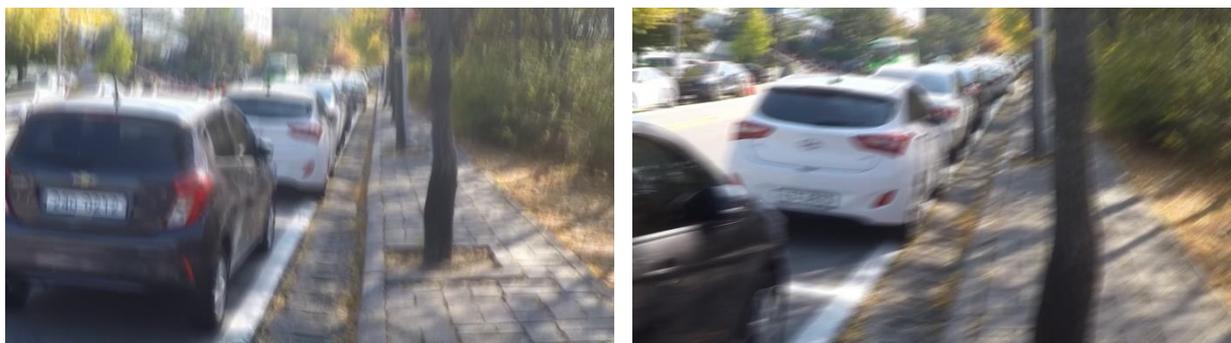
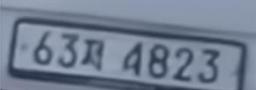
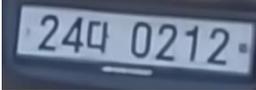
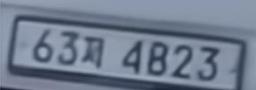
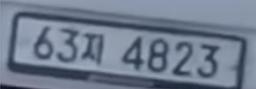
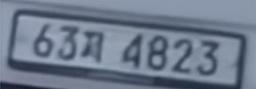
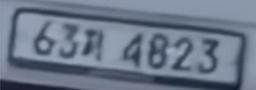
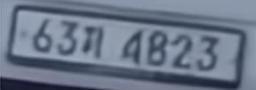
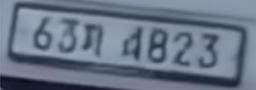
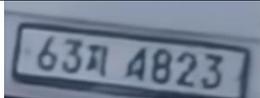


Рисунок 1 – Примеры исходных изображений

В табл.1 перечислены выявленные в результате аналитического исследования алгоритмические и программные решения, предназначенные для автоматизированного устранения смазов [29]. Приводятся сведения по среднему времени обработки отдельных кадров с помощью соответствующего решения и результаты обработки в виде фрагментов целевых областей. Отметим, что все актуальные решения реализованы в виде программного кода, использующего для вычислений графические процессоры.

Таблица 1 – Полученные результаты деконволюции

№	Метод	Время обработки, с.	Фрагмент 1		Фрагмент 2	
			PSNR	SSIM	PSNR	SSIM
1	Test-time Local Converter (TLC) [30]	13.5				
			33.012	0.939	31.186	0.929
2	Multi-Axis MLP [31]	19.5				
			30.556	0.917	31.748	0.934
3	EFNet [32]	1.10				
			35.184	0.948	35.695	0.962
4	Learning degradation [33]	3.60				
			32.549	0.932	31.739	0.934
5	Deep Generalized Unfolding (DGU) [34]	5.80				
			22.582	0.735	29.923	0.903
6	NAFNet width32 [35]	5.00				
			31.789	0.918	30.0	0.916
7	Stripformer [36]	0.03				
			28.060	0.876	30.222	0.916

8	Uformer [37]	9.40				
			32.296	0.928	31.477	0.929

БПЛА активно используются в мире для автоматизированного обследования и фиксации автомобильных номеров, поэтому подобная подзадача является актуальной. В данном случае используются изображения с автомобильными номерами. Для численной оценки были получены соответствующие отношения сигнала к шуму (Peak Signal-to-Noise Ratio, PSNR) и значения величин структурных подобий (Structure SIMilarity, SSIM), использовалась программная библиотека «skimage.metrics». Выполнено сравнение полученных изображений с изображениями без смазов (подгруппа снимков «sharp» из датасета GoPro). При полном совпадении изображений значение SSIM = 1, для абсолютно разных изображений SSIM = 0.

Экспериментальные исследования, проведенные с использованием этих и других изображений из датасета GoPro, показали, что нейронная сеть Stripformer выполняет обработку в режиме реального времени, при этом получаются достаточно качественные выходные изображения. Stripformer имеет архитектуру, основанную на идее трансформеров. «Визуальный трансформер» состоит из токенизатора, трансформера и проектора. Stripformer строит внутренние и внешние ленточные токены (strip tokens) в процессе перерасчета информативных признаков изображения в горизонтальных и вертикальных направлениях, что позволяет выявлять зашумленные фрагменты с различной ориентацией (направлением распространения) смаза. Нейронная сеть совмещает внешние и внутренние ленточные слои внимания (attention layers), чтобы определить степень и характер размытия. В дополнение к обнаружению специфических для области размытых паттернов различных ориентаций и размеров, Stripformer успешно работает даже без данных о динамике сцены и не требует огромного массива данных для обучения. Следует отметить высокое качество работы нейронной сети EFNet, которая, однако, не может использоваться в режиме реального времени. Пример работы Stripformer показан на рис.2.

Обученная оригинальная модель требует для своей работы наличия видеокарты с не менее чем 8 ГБ видеопамати. В рамках настоящего исследования выполнена небольшая модификация программного кода Stripformer, связанная с особенностями запуска модели на заявленном процессоре общего назначения с использованием библиотеки Torch, однако она работает неприемлемо долго, порядка 90 секунд на каждый кадр, что в 3000 раз дольше, чем на используемой видеокarte.



Рисунок 2 – Результат удаления смазов на снимке с помощью Stripformer

Таким образом, предложенное решение позволяет в режиме реального времени получать данные от БПЛА, очищенные от смазов.

## 1.2. Оптическая стабилизация видеопотока

При оптической реализации стабилизации видеопотока кадр обрезается со всех сторон, а оставшиеся части кадра используются для компенсации движения и формирования итогового изображения. Эта компенсация обычно выполняется с помощью ИНС и статистических методов на основе предыдущих кадров, для чего поддерживается буфер определенного размера.

В качестве тестовых в настоящей работе использовались выборки по 100 кадров, восстановленные с помощью Stripformer и подаваемые со скоростью 24 кадра в секунду. В табл.2 перечислены выявленные в результате исследования алгоритмические и программные решения, предназначенные для автоматизированной стабилизации видеопотока от БПЛА. Приводится среднее время обработки всего видеофайла. Каждое решение снабжено комментарием, отражающим приобретенный опыт его использования. Часть решений реализована с поддержкой графических ускорителей вычислений.

Таблица 2 – Экспериментальные данные по стабилизации видеопотока

№	Метод	Среднее время обработки, с.	Комментарий
1	VidGear [38]	3.91	Рамки кадров обрезаются с помощью отступа от границ и зума. Требуется инициализация стабилизатора, для чего используется объем входных данных порядка 24 кадров. Время подготовки стабилизатора к работе составляет порядка 0.70 с., то есть оно выполняется в фоновом режиме по мере поступления кадров на 30% быстрее того, как кадры поступают на обработку.
2	FuSta: Hybrid Neural Fusion [39]	1200	Используется графический ускоритель. На первом шаге запускается скрипт с генерацией данных оптического потока, который выполняет стабилизацию видео, но без отрезания краев, зума и пр. Полная обработка 100 кадров заняла 5 минут. Итоговый стабилизированный файл получился с разрешением 960x576 пикселей. На втором шаге запускается основная программа, использующая собранные данные оптического потока. Результат – набор изображений в исходном разрешении. Обработка выполняется за 15 минут.
3	MeshFlow: Minimum Latency Online Video Stabilization [40]	1260	Проведены эксперименты с двумя профилями настроек. Профиль Original: отличное качество без рамок, кадр обрезается автоматически. Профиль Constant High: обнаружены дефекты изображения в виде неестественно наклоненных объектов, например, автомобилей.
4	Video Stabilization with L1 optimal camera paths [41]	0.50	Результирующее видео имеет неудовлетворительное качество: присутствует постоянные zoom-in и zoom-out изображения, что выглядит неестественно.
5	Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths [42]	120	Итоговое изображение существенно обрезается при применении достаточно высоких уровней стабилизации. Например, при пороге в 40 пикселей изображение с 1280x720 уменьшается до 1180x620. Использование меньшего порога дает недостаточный уровень стабилизации видео. Обработка разбита на несколько этапов: 2 минуты на предобработку, затем быстрая стабилизация с использованием полученных данных.
6	Video stabilization using homography transform [43]	4.18	Итоговое изображение существенно обрезается, особенно при высоких коэффициентах стабилизации. Наилучшие результаты получены на

			realtime-стабилизации, при значении параметра $amount=20$ .
7	DUT: Learning Video Stabilization [44]	32.0	Используется графический ускоритель. Метод опирается на несколько предобученных нейронных сетей. В основе подхода – самообучение на нестабильных видеоданных. Изображение автомобиля на итоговом видео немного отклоняется в стороны, однако в целом получившееся видео можно назвать качественным.
8	Deep Iterative Frame Interpolation for Full-frame Video Stabilization [45]	100	Используется графический ускоритель. Итоговый видеопоток потерял плавность перехода между кадрами.
9	PWStableNet [46]	8.00	Используется графический ускоритель. Результирующее видео с «рывками».

Наилучшие результаты в задаче стабилизации получены с использованием библиотеки VidGear, в которой накапливается массив особых точек на заданном числе кадров с дальнейшим его обновлением в режиме реального времени. Накопленные данные используются для оценки движения камеры БПЛА с дальнейшей компенсацией ее нестабильности. Реализация соответствует подходу, используемому в библиотеке OpenCV [47]. При этом в целом обработка происходит в конвейерном режиме.

Проведенные исследования позволили получить решение двух важнейших задач улучшения видеопотока от БПЛА: удаление смазов и стабилизация видео. Нейронная сеть Stripformer покадрово удаляет смазы, затем обработанные данные сразу попадают в буфер модуля стабилизации на базе библиотеки VidGear, с помощью которого осуществляется стабилизация видеопотока. Итоговое решение позволяет осуществлять полный цикл обработки видеопотока с задержкой не более чем на 0.03 секунды за счет применения конвейерного метода обработки данных. Обработка выполняется в режиме реального времени.

## 2. Решение задачи раннего обнаружения возгораний

Негативные последствия пожаров на социо-эколого-экономические сферы предлагается парировать за счет своевременного реагирования на возникновение огня в наблюдаемой зоне. Для этого предлагается использовать хорошо себя зарекомендовавшие в решении задач классификации технологии искусственных нейронных сетей для поиска огня и дыма по изображениям и видеопотокам.

На рис.3 показана схема решения задачи обнаружения возгораний [48].

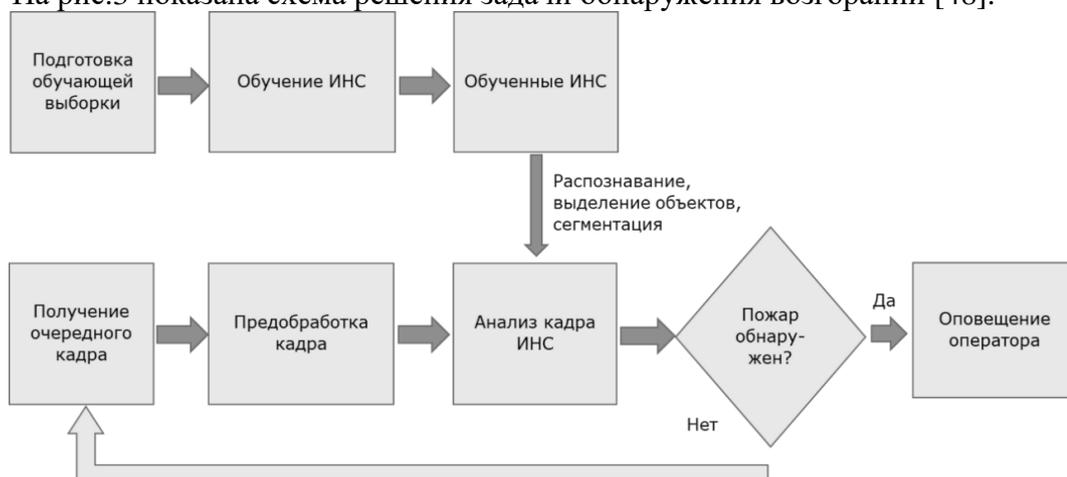


Рисунок 3 – Схема решения задачи классификации огня в видеопотоке

Предлагаемый подход основан на выделении признаков и использовании классификаторов с применением искусственных нейронных сетей сверточного типа. Процесс обучения ИНС (верхний ряд схемы) выполняется один раз, далее покадрово

выполняется анализ видеопотока (нижний ряд схемы) обученной сетью, которая выполняет классификацию видеокadra: пожар/нет пожара.

Далее рассмотрим разные версии и модификации ИНС семейства Yolo для решения задачи обнаружения пожаров.

### 2.1. Оценка возможностей архитектуры Yolo в задаче обработки данных

Для оценки возможностей актуальной версии программного пакета Ultralytics [49] по локализации целевых объектов (зон с огнем в лесном массиве) был проведен ряд экспериментов в различных режимах обработки поступающей информации: выделение целевых объектов с помощью неориентированных и ориентированных прямоугольных областей, а также в режиме сегментации, что соответствует режимам detect, obb и segment). Исследованы Yolo 5, Yolo 8, Yolo 8 Ghost, Yolo 8 Rtdetr, Yolo 9 и оригинальная архитектура Baidu [50]. Тестирование выполнялось в режиме с половинной точностью (параметр *half=True*).

Работа искусственных нейронных сетей проверялась на датасете, состоящем из данных от БПЛА, снятых в лесном массиве: оригинальные изображения; изображения, поврежденные с помощью смазов; изображения, восстановленные с помощью ИНС Stripformer [51]; в качестве экспериментальных данных использовался датасет FLAME [52] (данные для экспериментов с сегментацией очагов огня, полученные непосредственно с БПЛА); всего 1518 обучающих изображений и 379 тестовых; параметр *imgsz*, определяющий масштабирование данных, задан как 960; пример разных видов разметок приведен на рис. 4.



Рисунок 4 – Пример разметки

Результаты решения задачи выделения областей с огнем с помощью оригинальной архитектуры Yolo 8 в различных режимах работы представлены в табл.3-5.

Таблица 3 – Выделение объектов прямоугольными областями средствами Yolo 8l

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.773	0.703	0.761	0.364	0.73	0.301
Смазанные	0.755	0.659	0.725	0.347	0.70	0.240
Восстановленные	0.775	0.704	0.763	0.365	0.74	0.269

Среднее время обработки одного кадра: 22.6 мс.

Таблица 4 – Выделение ориентированными прямоугольными областями Yolo 8l

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.872	0.796	0.869	0.523	0.83	0.361
Смазанные	0.878	0.759	0.847	0.508	0.81	0.272
Восстановленные	0.853	0.802	0.861	0.522	0.83	0.224

Среднее время обработки одного кадра: 23.0 мс.

Таблица 5 – Результаты работы Yolo 8l, выделяющей объекты с помощью сегментации

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.807	0.714	0.787	0.450	0.76	0.238
Смазанные	0.760	0.678	0.752	0.431	0.72	0.140
Восстановленные	0.793	0.721	0.786	0.448	0.76	0.174

Среднее время обработки одного кадра: 27.5 мс.

Настоящая оценка выполняется в том числе с помощью достигаемых в различных режимах значениях F1-меры. В последней колонке табл.3-5 указано значение порога уверенности Yolo 8, при котором эта F1-мера была достигнута.

Для Yolo 8, выполняющей сегментацию, отдельно можно привести данные по точности, полноте, mAP50 и mAP50-95 непосредственно для генерируемых результирующих бинарных масок с разметкой целевых объектов (см. табл.6).

Таблица 6 – Результаты работы Yolo 8l, выделяющей объекты с помощью сегментации (данные по генерируемым бинарным маскам)

Данные	Точность	Полнота	mAP50	mAP50-95
Оригинальные	0.820	0.722	0.801	0.399
Смазанные	0.761	0.674	0.746	0.361
Восстановленные	0.793	0.721	0.788	0.383

В табл.7 приведены результаты модели Yolo 8l Ghost, которая также эффективна по точности и полноте, как и оригинальная, но работает быстрее, согласно исследованию [53]: в оригинальной модели Yolo 8 некоторые из слоев свертки были заменены на слои GhostConv и C3Ghost, обладающие более высоким быстродействием. Сам же слой детекции остался без изменений.

Таблица 7 – Выделение объектов прямоугольными областями средствами Yolo 8l Ghost

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.780	0.721	0.771	0.416	0.75	0.264
Смазанные	0.779	0.680	0.748	0.398	0.73	0.172
Восстановленные	0.786	0.716	0.766	0.410	0.75	0.243

Среднее время обработки одного кадра: 20.3 мс.

В табл.8 приведены результаты модели Yolo 8l Rtdetr, в которой оригинальный слой Yolo 8 для детекции объектов заменен на слой детекции из трансформера RT-DETR, разработанного компанией Baidu. Сама же архитектура Yolo 8l до слоя детекции не менялась. RT-DETR использует возможности Vision Transformers (ViT) для эффективной обработки разномасштабных признаков за счет развязки внутримасштабного взаимодействия и межмасштабного слияния.

Таблица 8 – Выделение объектов прямоугольными областями средствами Yolo 8l Rtdetr

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.741	0.613	0.667	0.332	0.67	0.407
Смазанные	0.694	0.590	0.621	0.297	0.64	0.361
Восстановленные	0.721	0.610	0.661	0.326	0.66	0.375

Среднее время обработки одного кадра: 48.9 мс.

В табл.9 представлены тесты оригинальной архитектуры от Baidu (трансформер RT-DETR), без использования слоев от Yolo 8, а именно – L-модификация [54], режим detect.

Таблица 9 – Результаты работы L-модификации ИНС от Baidu

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.838	0.742	0.770	0.396	0.79	0.512
Смазанные	0.788	0.662	0.687	0.333	0.72	0.447
Восстановленные	0.832	0.715	0.750	0.378	0.77	0.531

Среднее время обработки одного кадра: 43.7 мс.

В табл.10 результаты запуска L-модификации ИНС от Baidu в режиме obb.

Таблица 10 – Результаты работы L-модификации ИНС от Baidu в режиме obb

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.853	0.831	0.880	0.552	0.84	0.046
Смазанные	0.858	0.752	0.834	0.515	0.80	0.040
Восстановленные	0.859	0.813	0.873	0.548	0.84	0.073

Среднее время обработки одного кадра: 23.4 мс.

В табл.11 результаты запуска L-модификации ИНС от Baidu в режиме segment.

Таблица 11 – Результаты работы L-модификации ИНС от Baidu в режиме segment

Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
Оригинальные	0.824	0.712	0.786	0.444	0.76	0.363
Смазанные	0.800	0.651	0.743	0.416	0.72	0.286
Восстановленные	0.797	0.723	0.783	0.443	0.76	0.292

Среднее время обработки одного кадра: 26.0 мс.

В табл.12 приведены значения точности, полноты, mAP50 и mAP50-95 непосредственно для генерируемых результирующих бинарных масок с разметкой целевых объектов для этой архитектуры ИНС. Достижимые значения F1-мер хуже, чем в режиме obb. Однако, точно такие же, как и в случае с оригинальной архитектурой Yolo 8.

Таблица 12 – Результаты работы ИНС, выделяющей объекты с помощью сегментации (данные по генерируемым бинарным маскам для L-модификации ИНС от Baidu)

Данные	Точность	Полнота	mAP50	mAP50-95
Оригинальные	0.792	0.672	0.756	0.361
Смазанные	0.767	0.616	0.702	0.328
Восстановленные	0.798	0.660	0.746	0.349

Результаты по генерируемым бинарным маскам очень близки к тем, что показывает оригинальная архитектура Yolo 8l.

Дополнительно исследована архитектура Yolo 9c в режимах detect, obb и segment (см. табл.13).

Таблица 13 – Результаты работы Yolo 9c в режимах detect, obb и segment

Режим	Данные	Точность	Полнота	mAP50	mAP50-95	F1	Порог уверенности
detect	Оригинальные	0.778	0.733	0.779	0.415	0.76	0.222
	Смазанные	0.784	0.674	0.748	0.396	0.73	0.270
	Восстановленные	0.794	0.722	0.775	0.412	0.76	0.306
obb	Оригинальные	0.858	0.783	0.853	0.519	0.82	0.325
	Смазанные	0.842	0.765	0.834	0.501	0.80	0.119
	Восстановленные	0.854	0.802	0.858	0.527	0.83	0.182
segment	Оригинальные	0.763	0.670	0.728	0.386	0.71	0.259
	Смазанные	0.751	0.624	0.699	0.371	0.68	0.194
	Восстановленные	0.750	0.686	0.733	0.394	0.72	0.220

Среднее время обработки одного кадра: 19.6 мс.

Для Yolo 9c, выполняющей сегментацию, отдельно можно привести данные по точности, полноте, mAP50 и mAP50-95 непосредственно для генерируемых результирующих бинарных масок с разметкой целевых объектов (см. табл.14).

Таблица 14 – Результаты работы Yolo 9c, выделяющей объекты с помощью сегментации (данные по генерируемым бинарным маскам)

Данные	Точность	Полнота	mAP50	mAP50-95
Оригинальные	0.793	0.696	0.758	0.368
Смазанные	0.805	0.628	0.728	0.343
Восстановленные	0.771	0.699	0.752	0.366

Таким образом, следует констатировать, что наилучшие результаты получены при работе L-модификации ИНС от Baidu и Yolo 8l с ориентированными прямоугольными областями, однако, первая показывает очень низкую уверенность в ответах, что говорит о том, что класс «огонь» очень слабо отделился от фона. Отметим, что поиск зон с огнем осуществляется одинаково успешно как на оригинальных данных, так и на восстановленных. Применение ИНС Stripformer позволяет существенно снизить эффект от смазов, возникающих при съемке с БПЛА. ИНС Yolo 8 Ghost позволяет существенно снизить время обработки без чувствительного снижения качества локализации. Модификация Yolo 8 Rtdetr показала наихудшие результаты. Сравнение ИНС Yolo 8l, L-модификации ИНС от Baidu и Yolo 9c в режиме сегментации показывает, что первые две работают с одинаковым качеством, а Yolo 9c существенно отстает, однако самая быстрая из них.

Полученные результаты показали эффективность применения программного пакета Ultralytics для решения задач локализации целевых объектов. Нейронная сеть Stripformer успешно удаляет смазы кадров. В случае с локализацией мест с огнем наилучшие результаты получены при работе Yolo 8l с ориентированными прямоугольными областями, эта же ИНС может эффективно применяться и для сегментации.

## 2.2. Задача бинарной классификации

В данном разделе описаны эксперименты по бинарной классификации изображения. Рассматривается два класса – «огонь есть», «огня нет». В обучающей выборке для решения задачи бинарной классификации кадров с БПЛА 39375 изображений (25018 с огнем и 14357 без), в тестовой – 8617 изображений (5137 с огнем и 3480 без огня). В исследованиях на эпохах обучения применялись сгенерированные пакеты по 32 изображения. Использовалась аугментация со случайным отражением изображений по горизонтали и случайными поворотами в пределах 15 градусов.

Всего было проведено четыре группы экспериментов, в каждой исследовались авторские модификации нейронных сетей. В первой – модификация ИНС Xception v1 [55]; во второй – Shufflenet v2 [56]; в третьей – LW-Fire [57]; в четвертой – проведены эксперименты с использованием предобученных на базе ImageNet актуальных нейронных сетей из библиотеки Keras [58], эти сети использовались как бэкбоны (от англ. backbone). Бэкбон – основная (базовая) сеть, служащая для извлечения признаков из поступающего на вход изображения. Данная часть сети является вариативной, в ее основу могут входить классификационные нейросети, такие как ResNet, VGG, EfficientNet и др., с удалением последнего выходного слоя. В качестве инструментов для оценки применялась полнота, нормализованные точность и F1-мера [59].

В первой группе экспериментов вместо первых двух слоев свертки Xception v1 используется один слой с 128 картами признаков (в оригинальной архитектуре – два слоя с 32 и 64 картами признаков соответственно), настроечный вектор с указанием числа карт признаков во внутренних подслоях инициализирован как (8, 12, 18, 24) вместо (128, 256, 512, 728), крайний слой SeparableConv2D взят с 128 картами признаков вместо 1024. Во второй группе экспериментов в качестве основы была взята архитектура shufflenetv2\_x2.0, для обучения использовался Adam-оптимизатор с L1- и L2-регуляризациями (оптимизаторы – важный компонент архитектуры нейронных сетей; они играют важную роль в процессе тренировки нейронных сетей, помогая им делать более точные прогнозы; оптимизаторы определяют оптимальный набор параметров модели, таких как вес и смещение, чтобы при решении конкретной задачи модель выдавала наилучшие результаты). В третьей группе экспериментов слои свертки LW-FIRE были взяты с 512, 128, 256, 256, 256 и 256 картами признаков вместо 32, 64, 128, 128, 128 и 128. Полносвязные слои взяты с 1024, 512, 256, 128 и 64 нейронами вместо 256, 128, 64, 32 и 16. Результаты четвертой группы экспериментов с использованием моделей InceptionV3, InceptionResNetV2, Xception v1, DenseNet201, NASNetMobile, MobileNetV3Large, EfficientNetV2S и ResNetRS152: нормализованные F1-меры 62%, 57%, 78%, 75%, 70%, 56%, 57% и 59% соответственно. То есть лучшие результаты у Xception и DenseNet201, что говорит о том, что признаки, извлекаемые этими ИНС наиболее адекватны решаемой задаче. К указанным бэкбонам подключалась полносвязная ИНС с 1024, 512, 256, 128, 64 и 1 нейроном соответственно, после каждого слоя (кроме последнего) вставлены dropout-слои, необходимые для минимизации эффекта переобучения. Результаты проведенных исследований показали (см. табл.15), что известные архитектуры ИНС показывают неудовлетворительные результаты в условиях сложных данных, максимально близких к тем, что мы имеем на практике. Даже тонкая настройка архитектур не позволила получить значение нормализованной F1-меры более 90%.

Таблица 15 – Сравнение разных архитектур ИНС в задаче классификации кадров

Версия ИНС	Нормализованная точность, %	Полнота, %	Нормализованная F1-мера, %
Xception v1	83.0	83.0	83.0
Shufflenet v2	71.4	70.9	71.2
<b>LW-Fire</b>	<b>86.0</b>	<b>85.4</b>	<b>85.7</b>
Предобученная Xception v1 + полносвязная часть	77.2	78.82	78.0
Предобученная DenseNet201 + полносвязная часть	75.1	74.90	75.0

### 2.3. Выделение прямоугольных областей с очагами пожаров

В данном разделе описаны эксперименты по выделению очагов огня с помощью прямоугольников. Исходные изображения от БПЛА (датасет FLAME, файл 9) размером 3840x2160 были разделены на девять непересекающихся частей, каждая такая часть размером 1280x720 пикселей. Информация о зонах с источниками огня извлечена из png-масок, подготовленных разработчиками исходного датасета для сегментации (FLAME, файл 10), затем с помощью алгоритма маркировки связанных компонент были извлечены данные о зонах с огнем: сведения о размерах и координаты сохранены в текстовые файлы. Далее фрагменты снимков с информацией о точной локации мест с огнем подавались на ИНС. Чтобы из одного датасета сформировать и тестовые изображения, и обучающие, выполнялось разбиение обработанных данных пакетами по 90 изображений. Первый пакет идет на обучение, второй сознательно выбрасывается, третий – на тестирование, четвертый выбрасывается. Такой подход позволяет обеспечить то, что тесты и обучающая выборка будут из разных по времени кадров видеопотока. В обучающей и тестовой выборках в итоге оказалось по 4500 изображений. Из них 2601 и 2604 соответственно – изображения с лесом без очагов пожаров. Отметим, что при подаче изображений на ИНС их размер менялся на 640x640 пикселей.

Для экспериментов использован YOLO-подход. Полученные результаты представлены в табл.16. Отметим, что mAP вычисляется без привязки к критически значимому порогу уверенности, что принято в исследованной научной литературе: в экспериментах используется значение порога уверенности в 0.1%.

Таблица 16 – Результаты экспериментов по применению YOLO для задачи поиска прямоугольных областей с очагами пожаров

Версия ИНС	Точность, %	Полнота, %	F1-мера, %	Порог уверенности, %	mAP@.5, %	mAP@.5:.95, %	Скорость обработки, кадров/с
4 csp mish	78.200	70.400	74.10	51.0	74.500	39.200	55
5s	79.440	71.123	75.05	51.5	76.691	40.705	49
<b>5m с SiLU-активацией</b>	<b>79.308</b>	<b>74.345</b>	<b>76.75</b>	<b>44.4</b>	<b>77.899</b>	<b>41.274</b>	<b>45</b>
5m с Mish-активацией	80.159	72.126	75.93	49.9	77.143	41.081	43
5l	79.526	72.253	75.72	49.7	76.846	41.834	35
6t	73.700	72.300	73.00	50.0	73.7	38.320	122
7x	73.440	69.250	71.28	23.6	72.790	35.430	39

YOLO 5m с SiLU-активацией обеспечивает высокие показатели при решении поставленной задачи. Анализ результатов нейросетевой обработки тестовой выборки выявил, что явные ошибки выделения огня фактически отсутствуют, однако, найденный огонь зачастую раздроблен на части, во многих случаях такое разделение не соответствует разметке в датасете, что приводит к падению числовых показателей. Очевидно, что для такого класса как «огонь» необходимо в будущем подготовить более гибкие инструменты оценки результатов ИНС, учитывающие возможность такого дробления целевых объектов на части.

Для оценки полученных результатов и перспектив использования YOLO-подхода обратимся к работе [60], посвященной задаче обнаружения огня и дыма на изображениях, из которой возьмем соответствующую статистическую информацию с информацией о лучших решениях – см. табл.17. Обработывались изображения размером 512x512 пикселей, использовался графический ускоритель Nvidia GeForce RTX 3060 Laptop.

Таблица 17 – Статистика по использованию разнородных детекторов объектов

Версия ИНС	Бэббон	Усредненная точность, %	mAP@.5, %	mAP@.75, %	Усредненная полнота, %	Скорость обработки, кадров/с
<i>Детекторы с двумя стадиями</i>						
Regionlets	ResNet-101	70.87	81.49	74.05	44.64	–
Mask R-CNN	ResNet-101	72.31	82.14	75.21	45.29	–
LH R-CNN	ResNet-101	74.62	84.12	77.13	46.29	–
Cascade R-CNN	ResNet-101	75.54	85.96	78.86	47.16	–
<b>D-RFCN + SNIP</b>	<b>DPN-98</b>	<b>77.22</b>	<b>87.31</b>	<b>80.82</b>	<b>48.01</b>	–
<i>Одностадийные детекторы</i>						
YOLOv5	CSP-Darknet53	66.02	77.12	69.25	41.52	156
RefineDet512(multi scale)	ResNet-101	70.25	80.96	73.29	43.97	67
FireNet	–	77.15	87.23	80.34	48.48	23
<b>DCNN</b>	–	<b>77.64</b>	<b>87.64</b>	<b>80.40</b>	<b>48.60</b>	<b>30</b>
<b>DeepSmoke</b>	<b>EfficientNet</b>	<b>78.16</b>	<b>87.31</b>	<b>80.80</b>	<b>48.93</b>	<b>33</b>
<i>Детекторы с двумя стадиями (мультиориентированные)</i>						
F <sup>3</sup> Net	ResNet-152	74.26	–	–	–	–
CSL	ResNet-152	74.73	–	–	–	–
MRDet	ResNet-101	74.99	–	–	–	–
SCRDet++	ResNet-101	75.12	–	–	–	–
<b>FR-EST</b>	<b>ResNet-101-DCN</b>	<b>77.15</b>	–	–	–	–
<i>Одностадийные детекторы (мультиориентированные)</i>						
RDD	ResNet-101	76.19	86.02	79.39	48.11	79
S <sup>2</sup> A-Net	ResNet-101	77.69	87.08	79.92	48.26	46
GWD	ResNet-152	78.15	87.69	80.46	48.82	29
<b>MVMNet</b>	<b>CSP-Darknet-53</b>	<b>78.92</b>	<b>88.05</b>	<b>81.15</b>	<b>49.08</b>	<b>122</b>

Полученный результат по YOLO 5 совпал с приведенным выше. Если, например, ориентироваться на показатель mAP@.5, становится очевидным, что его можно увеличить на величину порядка 10%, то есть снизить число ложных срабатываний. Например, ИНС MVMNet, опирающаяся на архитектуру YOLO 5, для получения столь высоких показателей использует мультиориентированные прямоугольники, решения Soft-SPP, VAM и Mixed-NMS.

#### 2.4. Выделение границ пожаров с точностью до пикселя

В данном разделе приводятся результаты практического решения задачи семантической сегментации изображений с БПЛА для обнаружения источников огня. Для обучения и тестирования использовались те же данные, что и для экспериментов с YOLO, но с пиксельной точностью. Получены существенно лучшие результаты по сравнению

теми, что были достигнуты на задаче классификации и выделения прямоугольных областей с целевыми очагами огня (см. табл.18).

Таблица 18 – Результаты экспериментов по выделению границ очагов пожаров с точностью до пикселя

Версия ИНС	Нормализованная F1-мера, %	Размер кадра	Скорость обработки, кадров/с
U-Net с добавлением по одному слою свертки и кодеру, и декодеру	97.48	512x512	29.0947
U-Net с VGG16 в качестве бэкбона	99.35	512x512	12.1702
<b>U-Net с ResNet18 в качестве бэкбона</b>	<b>99.36</b>	<b>512x512</b>	<b>37.0069</b>
EfficientSeg	97.47	512x512	54.9941
TransUnet с 6 vit-блоками и 6 vit-заголовками, размер mhsa-блока 512, размер трансформера 128	97.40	512x512	18.7780
SwinUnet	91.81	448x448	26.8159
EffuNet	97.54	512x512	56.9498
Linknet с VGG16 в качестве бэкбона	99.30	512x512	11.6184
<b>Linknet с ResNet18 в качестве бэкбона</b>	<b>99.31</b>	<b>512x512</b>	<b>38.0806</b>

Замена бэкбона VGG16 на ResNet18 утраивает скорость работы нейросетей, однако на границах выделенных очагов пожара появляется небольшая «бахрома», что некритично и не мешает успешному решению поставленной задачи. Выявлена основная причина снижения показателей ИНС – мелкие фрагменты фона рядом с огнем некорректно классифицируются, в том числе это ветки и хвоя.

Проведенные эксперименты показали эффективность применения ИНС с архитектурами Xception v1 и LW-FIRE для решения задачи классификации потока кадров от БПЛА, обеспечивается F1-мера 85.7%. В задаче выделения очагов возгораний с помощью прямоугольников сравнивались актуальные версии YOLO (версии от 4 до 7), лучшие результаты показала YOLO 5m, обеспечившая F1-меру 76.75% в сочетании со скоростью обработки 45 кадров/с, то есть в режиме реального времени. В будущих исследованиях необходимо подготовить более гибкие инструменты оценки результатов ИНС, работающих с прямоугольными рамками, учитывающие возможность дробления целевых объектов на части. Кроме того, стало очевидно, что для поиска целевых регионов с огнем с помощью YOLO-сетей необходимо использовать определенные модификации. Подобные модификации позволяют увеличить mAP@.5 на 10%, то есть снижают число ложных срабатываний. В задаче сегментации лучшие результаты показали ИНС Linknet и U-Net с ResNet18-бэкбоном.

## 2.5. Алгоритм обнаружения линии горизонта

Авторами настоящего препринта предложен алгоритм, который улучшает исходные снимки, предназначенные для обнаружения возгораний, за счет устранения части информации, находящейся выше горизонта [61]. Таким образом, отсекается ненужная часть изображения (небо), которая может содержать образы (например, облака), которые классификатор может отнести к дыму, что даст ложно положительный результат классификации.

Решается задача обнаружения линии горизонта на снимках местности, полученных от камеры беспилотного летательного аппарата. Цель исследования – повысить качество обнаружения природных пожаров путем определения линии горизонта и преобразования на этой основе исходного изображения. Предлагается метод, адаптированный к сниженной видимости в условиях природного пожара. Способ основан на вычислении локального контраста, эквивалентного показателю информативности пикселей изображения. Размер сканирующего окна устанавливается индивидуально для каждого изображения, путем применения модели множественной линейной регрессии. В качестве факторов модели

выбраны признаки, характеризующие особенности условий съемки. Разработан алгоритм слияния точек линии горизонта в единое множество.

Задача определения дыма по фотографиям является актуальной в свете необходимости раннего обнаружения пожаров на территориях лесных регионов России. Отечественные и зарубежные исследования показали высокую трудность решения этой задачи, в особенности из-за плохой различимости дыма на фоне облаков, что приводит к ошибкам [62-65]. В связи с этим в проведенных исследованиях поставлена задача выделения на снимках линии горизонта, что позволяет выполнить преобразование исходной информации, путем удаления потенциально облачной части снимка, мешающей нахождению дыма. Алгоритм строится на принципах поиска границ, определения информативных признаков изображения и построении модели множественной линейной регрессии.

Уменьшить количество ложных сигналов о задымлении можно путем применения метода, предназначенного для обнаружения горизонта при нечеткой границе с небом в сложных погодных условиях, например, туман и низкая облачность [66]. Кандидатами точек линии горизонта назначаются пиксели, находящиеся на пике перепада яркости вертикального градиента от светлого к темному. Метод усложнен многократной проверкой одних и тех же кандидатов на соответствие установленным правилам.

В условиях дождя предложено определять линию горизонта, применяя нейронную сеть DenseNet [67]. Предлагаемая модель устраняет помехи от дождевых капель. Информацию о краевых точках получают, вычитая сглаженное изображение из исходного. Линию горизонта воссоздают, используя алгоритм динамического программирования. Вычисляются все возможные линии в кадре. Предполагается, что наиболее длинная линия на изображении вероятностей является линией горизонта.

Примером использования принципов машинного обучения для решения рассматриваемой задачи является работа [68]. Используется семантическая сеть, обученная на наборе объектов (небо, деревья, здания, горы и т.д.), формирующих линию горизонта. Объекты, принадлежащие к разным классам, по-разному влияют на форму горизонта на открытых полуприродных ландшафтах.

Разумный компромисс между вычислительными затратами и производительностью предложен в публикации [69]. Применяются принципы неглубокого обучения для поиска линии горизонта в горной местности. Алгоритм включает использование детектора границ Canny. К каждому граничному пикселю подбирается фильтр из обученного банка фильтров и применяется к фрагменту изображения, центром которого он является. Метод разделения водной поверхности и неба на изображениях представлен в работе [70]. Применяя фильтр Гаусса, выполняют преобразование исходного изображения в градиентное. Регионами, потенциально принадлежащими линии горизонта, назначаются области, содержащие максимальные перепады значений интегрального изображения. Оптимальное положение линии горизонта определяется итеративным приближением кривой.

Из анализа источников следует, что разработанные методы обнаружения линии горизонта адаптированы под отдельные ситуации. В настоящем исследовании была поставлена и успешно решена задача разработки метода вычисления линии горизонта, устойчивого к влиянию погодных и ландшафтных условий. Идея предлагаемого метода – отсечение области облаков, потенциально мешающей определению задымлений.

Линию горизонта на изображении можно охарактеризовать как область, отделяющую гладкую текстуру неба от темной и менее однородной поверхности земли или воды. Границы этой области могут быть четкими и узкими или более размытыми и широкими, но неизменно содержат вертикальный градиентный переход от светлого к темному тону. Вычислим локальный контраст в качестве количественной оценки информативности изображения в скользящей окне. Количественно оценить яркостную контрастность можно, применяя формулу среднего квадратического отклонения (СКО) [71]:

$$\partial[X] = \sqrt{\left| \frac{1}{K^2} \sum_{i=1}^{K^2} X_i^2 - \left( \frac{1}{K^2} \sum_{i=1}^{K^2} X_i \right)^2 \right|}, \quad (1)$$

где  $K$  – размер стороны квадратного скользящего окна,  $X$  – значение яркости  $i$ -го пикселя в скользящем окне  $K \times K$ .

Чем больше значение  $\partial[X]$ , тем выше информативность фрагмента изображения. Таким образом, помещая центр скользящего окна в каждый пиксель снимка, производим оценку его информативности в пределах окна [72], в том числе точках линии горизонта. Результаты обработки некоторых ландшафтных снимков представлены на рис. 5.

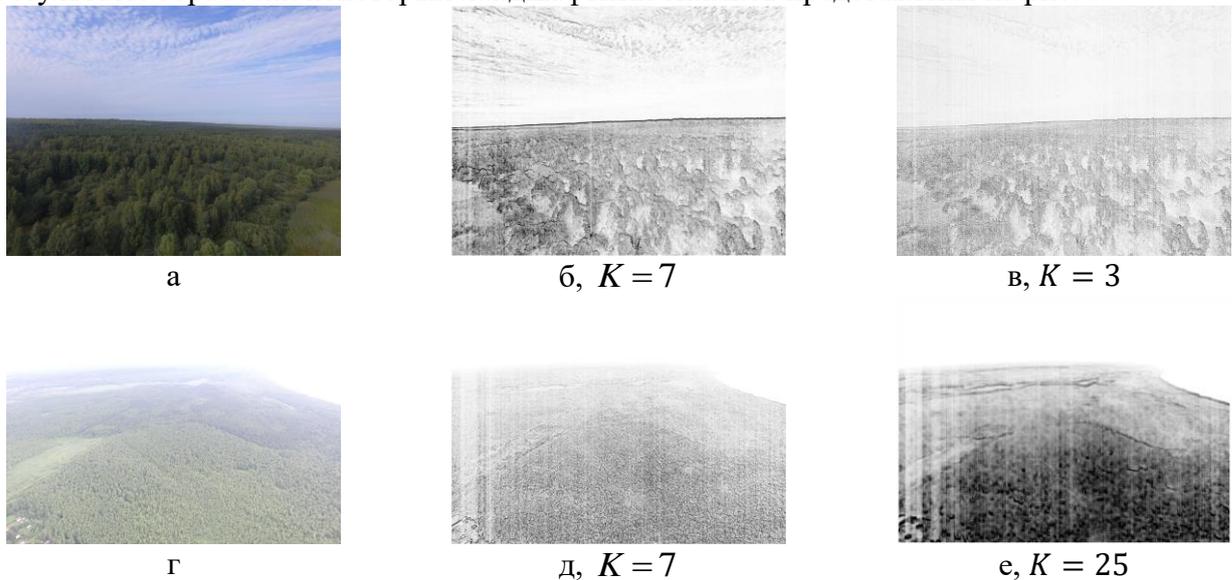


Рисунок 5 – Исходные изображения и результаты их обработки

СКО яркостей соседних пикселей определяет локальную контрастность [73]. Более четкая линия горизонта будет связана с высоким СКО и большей резкостью. Фрагмент изображения, содержащий значительное количество пикселей с высокой разницей в оттенках серого будет содержать высокое значение информативности.

Зададим некоторое пороговое значение  $t$  такое, что  $\partial[X]=0$ , когда  $\partial[X] < t$ . Отсевание таким образом малоинформативных точек повышает качество и скорость решения задач анализа изображений.

На рис. 6 показан процесс отсеивания малоинформативных точек.

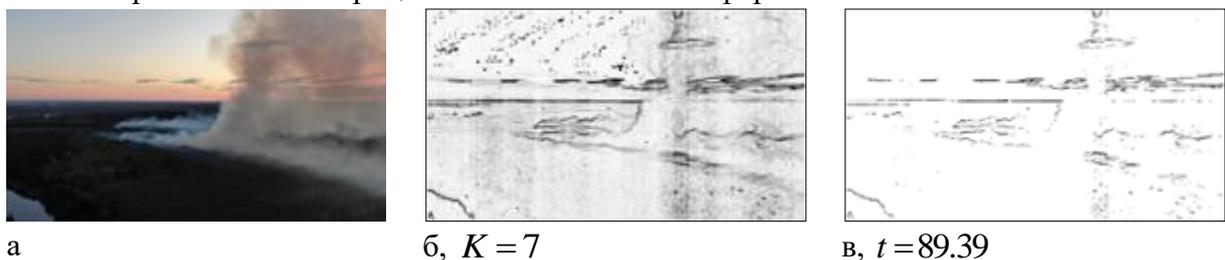


Рисунок 6 – Результат отсеивания малоинформативных значений СКО

Установлено, что размер стороны квадратного скользящего окна  $K$  ключевым образом влияет на результат решения ряда задач компьютерного зрения [74]. Окно малого размера работает как детектор градиента яркости на резких границах и недостаточно для определения размытой линии горизонта. По мере увеличения размера окна, тонкие границы становятся неразличимы, но при этом начинают проявляться широкие. Следовательно, выбор оптимального размера  $K$  скользящего окна и порога информативности  $t$  существенно влияет на качество решения задачи настоящей работы.

Для уточнения алгоритма построения линии горизонта введем понятия «левой», «центральной» и «правой» ее частей. Для этого введем две разбивающие линию точки:  $(y_{first}, x_{first})$  и  $(y_{end}, x_{end})$ , значения которых определены в приведенном ниже алгоритме.

*Шаг 1.* Вычисление факторов, влияющих на размер стороны скользящего окна  $K$ , и порога информативности  $t$ , предназначенного для отсеивания неинформативных точек.

Вычисляются факторы, характеризующие основные особенности отображения линии горизонта на фотоснимке, и строятся модели (уравнения) множественной линейной регрессии [75]:

$$t = a_0 + a_1 f_1 + \dots + a_{17} f_{17}, \quad (2)$$

$$N = d_0 + d_1 f_2 + \dots + d_{14} f_{15}, \quad (3)$$

где  $a_0, a_1 \dots a_{16}$  и  $d_0, d_1 \dots d_{14}$  – параметры регрессионной модели,  $f_1, \dots, f_{17}$  – факторы (объясняющие переменные, признаки).

Получаемый из уравнения (3) параметр  $N$  связан с размером  $K$  соотношением:  $K = 2N + 1$ . К факторам, характеризующим изображение, относятся признаки, отражающие: четкость или плавность цветояростного перехода в области линии горизонта, наличие или отсутствие объектов с высокой яркостью.

Введем показатель яркостной однородности полутонового изображения, определяемый формулой  $(Y_{max} - Y_{min}) / (M[Y] - \partial[Y])$ , где  $M[Y]$  – математическое ожидание яркости изображения,  $\partial[Y]$  – СКО,  $Y_{max}$ ,  $Y_{min}$  – максимальная и минимальная яркость соответственно. Чем больше разнородных деталей с четкими границами, тем меньше сглаженность изображения. Наоборот – высокая сглаженность соответствует однородной текстуре фотографии.

Введем показатель яркостной монотонности полутонового изображения, определяемый формулой  $M[Y] / \partial[Y]$ . Монотонность яркости показывает плавность переходов между участками изображения. Эмпирически замечена корреляция между оптимальным размером сканирующего окна и отношением средней яркости к среднему контрасту. При увеличении значения монотонности яркости размер сканирующего окна  $K$  целесообразно увеличивать, а порог отсеивания  $t$  – снижать. В таблице 19 показаны факторы, влияющие на размер скользящего окна и порога отсеивания.

Таблица 19 – Факторы, влияющие на размер скользящего окна и порога отсеивания (часть 1)

№	Название фактора	Формула	Коэффициент в регрессионной модели	
			$a_i$	$d_i$
1	Параметр $N$	$f_1 = N$	$a_1 = -12.06$	–
2	Математическое ожидание яркости красного канала	$f_2 = M[R] = \frac{1}{S} \sum_{i=1}^S r_i$ , где $S$ – количество пикселей изображения, $r_i$ – яркость $i$ -го пикселя в красном цветовом канале	$a_2 = 1.96$	$d_1 = -0.19$
3	СКО яркости в градациях красного канала	$f_3 = \partial[R] = \sqrt{M[R^2] - M^2[R]}$	$a_3 = 1.98$	$d_2 = 0.03$
4	Математическое ожидание яркости зеленого канала	$f_4 = M[G] = \frac{1}{S} \sum_{i=1}^S g_i$ , где $g_i$ – яркость $i$ -го пикселя в зеленом цветовом канале	$a_4 = 3.86$	$d_3 = -0.16$
5	СКО яркости в градациях зеленого канала	$f_5 = \partial[G] = \sqrt{M[G^2] - M^2[G]}$	$a_5 = 1.04$	$d_4 = 0.01$
6	Математическое ожидание яркости синего канала	$f_6 = M[B] = \frac{1}{S} \sum_{i=1}^S b_i$ , где $b_i$ – яркость $i$ -го пикселя в синем цветовом канале	$a_6 = 1.37$	$d_5 = 0.03$

7	СКО яркости в градациях синего канала	$f_7 = \partial[B] = \sqrt{M[B^2] - M^2[B]}$	$a_7 = 0.02$	$d_6 = 0.03$
8	Яркостная однородность голубого канала	$f_8 = \frac{b_{\max} - b_{\min}}{M[B] - \partial[B]}$ , где $b_{\max}$ – максимальная яркость в синем цветовом канале, $b_{\min}$ – минимальная	$a_8 = -0.06$	$d_7 = 0.01$
9	Яркостная монотонность голубого канала	$f_9 = \frac{M[B]}{\partial[B]}$	$a_9 = -1.89$	$d_8 = 1.19$
10	Математическое ожидание яркости в градациях серого	$f_{10} = M[Q] = \frac{1}{S} \sum_{i=1}^S q_i$ , где $q_i$ – яркость $i$ -го пикселя в градациях серого	$a_{10} = -4.78$	$d_9 = 0.42$
11	СКО яркости в градациях серого	$f_{11} = \partial[Q] = \sqrt{M[Q^2] - M^2[Q]}$	$a_{11} = -4.24$	$d_{10} = -0.14$
12	Максимальная яркость в градациях серого	$f_{12} = q_{\max} = \max q_i, i = 1, 2, \dots, S$	$a_{12} = -1.03$	$d_{11} = 0.04$
13	Разность между максимальной и минимальной яркостями в градациях серого	$f_{13} = \Delta = q_{\max} - q_{\min}$	$a_{13} = 0.13$	$d_{12} = -0.02$
14	Яркостная однородность в градациях серого	$f_{14} = \frac{\Delta}{M[Q] - \partial[Q]}$	$a_{14} = 1.31$	$d_{13} = 0.06$
15	Яркостная монотонность в градациях серого	$f_{15} = \frac{M[Q]}{\partial[Q]}$	$a_{15} = 69.22$	$d_{14} = 4.88$

По шестидесяти наблюдениям вычислена оценка факторов и свободные члены:  $a_0 = 164.92$ ,  $d_0 = -8.87$ . Полученные коэффициенты регрессионной модели представлены в таблице 19 выше. Факторы  $f_{16}$  и  $f_{17}$  будут рассмотрены на Шаге 5.

*Шаг 2. Предобработка изображения.*

Выполняется преобразование цветного изображения в изображение в оттенках серого. Яркости пикселей нормализуются по формуле  $X = (q - q_{\min}) / \Delta$ , где  $q$  – исходная яркость пикселя в градациях серого цвета.

*Шаг 3. Вычисление и нормализация информативных областей изображения [76].*

Каждому пикселю присваивается значение СКО яркости  $\partial[X]$  в области, центром которой он является. Далее выполняется нормализация:  $\bar{\partial}_{nm} = c(\partial_{nm}[X] - \partial_{m\min}[X]) / (\partial_{m\max}[X] - \partial_{m\min}[X])$ , где  $c = 255$  – максимальное значение яркости цифровых изображений в формате RGB,  $\partial_{m\min}[X]$  – минимальное значение информативности в  $m$ -ом столбце пикселей изображения,  $\partial_{m\max}[X]$  – максимальное значение информативности в  $m$ -ом столбце пикселей изображения,  $n$  – номер строки. В результате обработки получим яркостное изображение, характеризующее информативность фрагментов исходного кадра. Нормализация по столбцам пикселей, а не по всей площади изображения, способствует более четкому проявлению участков линии горизонта, перекрываемых дымом.

*Шаг 4. Фильтрация информативных пикселей изображения по яркости областей сверху и снизу.*

Вычисляются и сравниваются средние значения информативности, расположенных выше и ниже пикселя в  $m$ -ом столбце и  $n$ -ой строке изображения:

$$U_{nm} = \frac{1}{K} \sum_{i=n-1}^{n-K} \bar{\partial}_{im}, \quad V_{nm} = \frac{1}{K} \sum_{i=n+1}^{n+K} \bar{\partial}_{im}, \quad A_{nm} = \frac{1}{K} \sum_{i=n-1}^{n+K} \bar{\partial}_{im}.$$

Фильтрация выполняется по формуле

$$F_{nm} = \begin{cases} \bar{d}_{nm}, & \text{если } (V_{nm} > A_{nm}) \wedge (A_{nm} > U_{nm}), \\ 0, & \text{иначе} \end{cases}$$

где  $F_{nm}$  – отобранные значения яркостей. Получим изображения, представленные на рис. 7 е, ж, з.

*Шаг 5.* Вычисление значений факторов  $f_{16}$ ,  $f_{17}$ , коэффициентов регрессионной модели  $a_{16}$ ,  $a_{17}$  и порога  $t$  (уравнение (2)). Порог вычисляется из уравнения (2).

В табл. 19 показаны два оставшихся фактора, влияющих на размер скользящего окна и порога отсеивания.

Таблица 19 – Факторы, влияющие на размер скользящего окна и порога отсеивания (часть 2)

№	Название фактора	Формула	Коэффициент в регрессионной модели $a_i$
16	Средняя яркость информативных пикселей	$f_{16} = \frac{1}{W} \sum_{i=1}^W F_i$ , где $W$ – количество пикселей с ненулевой информативностью, т.е. таких что: $F_i \neq 0$ .	$a_{16} = 0.37$
17	Коэффициент информативности изображения	$f_{17} = \frac{S}{W}$	$a_{17} = 1.71$

*Шаг 6.* Отсеивание малоинформативных точек.

Фильтрация выполняется по формуле

$$I_{nm} = \begin{cases} F_{nm}, & \text{если } F_{nm} \geq t \\ 0, & \text{иначе} \end{cases},$$

где  $t$  – порог, вычисленный из уравнения множественной регрессии (2). Результат фильтрации показан оттенками серого цвета на рис. 7 и, к, л.

*Шаг 7.* Объединение точек в линии.

Обозначим линию как множество  $L_{yx}$ , для которого  $(y, x)$  – координаты первого элемента.

Шаг 7.1. Присваиваем множеству  $L_{yx}$  пиксель с ненулевой информативностью  $I_{yx} \neq 0$ .

Шаг 7.2. Поиск пикселя с максимальной информативностью  $I_{ij}$  в столбце с номером  $j = x + 1$ , такой что  $i \in [y - (\alpha - N); y + (\alpha - N)]$ . Далее в экспериментах принято  $\alpha = 10$ . При работе в горной местности принимается  $\alpha = 80$ .

Шаг 7.3. Если  $I_{ij} \neq 0$ , то присвоить элемент  $I_{ij}$  множеству  $L_{yx}$ ; сделать замену:  $x = j$ ,  $y = i$ ; перейти к шагу 7.2. В противном случае – выполняем поиск очередного пикселя с ненулевой информативностью, такого что  $x > j$ , и переходим к шагу 7.1.

Если  $j = J$  где  $J$  – размер ширины изображения в пикселях, то перейти к шагу 8.

Примеры получаемых линий обозначены зеленым цветом на рис. 7 и, к, л.

*Шаг 8.* Объединение линий.

Обозначим группу линий как множество  $Z_{yx}$  для которого  $(y, x)$  – координаты первого пикселя в первой линии,  $(y_{last}, x_{last})$  – координаты последнего пикселя в последней линии. Каждая линия  $L_{yx}$ , представляемая множеством своих точек, может принадлежать

только одному  $Z_{yx}$ , т.е. для любых точек  $(y_1, x_1)$  и  $(y_2, x_2)$  выполняется условие  $Z_{y_1x_1} \cap Z_{y_2x_2} = \emptyset$ .

Шаг 8.1. Включаем в  $Z_{yx}$  точки первой линии  $L_{yx}^k$ ,  $k=1$ , т.е.  $L_{yx}^k \subset Z_{yx}$ .

Шаг 8.2. Поиск линии  $L_{yx}^{k+1}$  с максимальной информативностью первой точки  $I_{ij}$ ,  $j = x_{last} + 1$ , такой что  $i \in [y - \varepsilon; y + \varepsilon]$ , где  $\varepsilon = (\alpha - N)(j - x_{last})$ .

Шаг 8.3. Если  $I_{ij} \neq 0$ , то  $L_{yx}^{k+1} \subset Z_{yx}$ , выполнить замену  $k = k + 1$  и перейти к шагу 8.2. В противном случае перейти к шагу 8.1.

Если  $j = J$  то перейти к шагу 9.

*Шаг 9.* Выбор линии  $Z_{yx}$  с наибольшей длиной. Выбранная линия назначается «центральной частью» линии горизонта.

*Шаг 10.* Восстановление пропусков между участками «центральной части» линии горизонта.

Введем обозначения:  $(y_f, x_f)$  – координаты начала пропуска,  $(y_l, x_l)$  – координаты окончания пропуска,  $\gamma$  – угол наклона линии горизонта на участке пропуска.

Шаг 10.1. Поиск точки с максимальной информативностью  $I_{ij}$  в столбце с номером  $j = x_f + 1$ , такой что  $i \in [y_f - \beta; y_f + \beta]$ , где  $\beta = \alpha - N$ .

Шаг 10.2. Если  $I_{ij} \neq 0$ , выполнить замену  $x_f = j$ . В противном случае: вычислить координату  $i$  по формуле:  $i = y_f + (j - x_f) \sin \gamma$ , выполнить замену  $x_f = j$ .

Если  $j = J$  то перейти к шагу 11. В противном случае – перейти к шагу 10.1.

*Шаг 11.* Вычисление «правой части» линии горизонта.

Введем обозначения:  $(y_{start}, x_{start})$  – координаты первой точки «центральной части» линии горизонта,  $(y_{end}, x_{end})$  – координаты последней,  $(y_{last}, x_{last})$  – координаты последней точки «правой части» линии горизонта,  $\sin \varphi = (y_{end} - y_{start}) / (x_{end} - x_{start})$ .

Шаг 11.1. Примем  $(y_{last}, x_{last}) = (y_{end}, x_{end})$ .

Шаг 11.2. Найдем точку с максимальной информативностью  $I_{ij}$  в столбце с номером  $j = x_{last} + 1$ , такую что  $i \in [y_{last} - \beta; y_{last} + \beta]$ .

Шаг 11.3. Назначаем крайней точкой пиксель с координатами, полученными по формуле:  $(y_{last}, x_{last}) = \begin{cases} (i, j), \text{ если } I_{ij} \neq 0 \\ (y_{start} + (j - x_{start}) \sin \varphi, j), \text{ иначе} \end{cases}$ .

Если  $j = J$  то перейти к шагу 12, в противном случае – выполнить переход к шагу 11.2.

*Шаг 12.* Вычисление «левой части» линии горизонта.

Введем обозначение:  $(y_{first}, x_{first})$  – координаты первой точки «левой части» линии горизонта.

Шаг 12.1. Примем  $(y_{first}, x_{first}) = (y_{start}, x_{start})$ .

Шаг 12.2. Найдем точку с максимальной информативностью  $I_{ij}$  в столбце с номером  $j = x_{first} - 1$ , такую что  $i \in [y_{first} - \beta; y_{first} + \beta]$ . Назначаем первой точкой пиксель с координатами, полученными по формуле:  $(y_{first}, x_{first}) = \begin{cases} (i, j), \text{ если } I_{ij} \neq 0 \\ (y_{start} + (j - x_{start}) \sin \varphi, j), \text{ иначе} \end{cases}$ .

Если  $j=0$  то перейти к шагу 13. В противном случае – выполнить переход к шагу 12.2.

На рис. 7 м, н, о показан результат восстановления линии горизонта.

*Шаг 13.* Проверка линии горизонта на истинность.

Вычисляются и сравниваются средние значения яркости в градациях серого над и под линией горизонта. Введем обозначения:  $l$  – длина линии горизонта,  $q(y_i, x_i)$  – яркость в градации серого  $i$ -ой точки линии горизонта. Если выполняется условие

$$\frac{1}{l} \sum_{i=1}^l q_{(y_i-N, x_i)} > \frac{1}{l} \sum_{i=1}^l q_{(y_i+N, x_i)}, \quad (4)$$

то линия горизонта считается вычисленной правильно, выполняется переход к шагу 14. В противном случае (см. рис. 7 н, р) – увеличивается размер скользящего окна и выполняется переход к шагу 3.

Неверная линия горизонта может быть получена в случаях тумана (см. рис. 7 б, р) или задымления на дальнем плане фотографии. В таких условиях линия горизонта размыта и слабо выражена. Соответственно, надо увеличить размер сканирующего окна, что поспособствует повышению СКО в областях с нечеткими границами. На последующих итерациях  $N=N+2$ . Если на первой итерации  $N < 9$ , то на второй итерации устанавливается  $N=9$ . Порог, в этом случае, определяется по формуле  $t = \rho - 2(T-1)$ , где  $\rho=15$  – эмпирическая константа,  $T$  – номер итерации. Всего выполняется не больше четырех итераций. Если за четыре итерации не найдена линия горизонта, удовлетворяющая условию (4), то выполняется переход к шагу 17.

*Шаг 14.* Сглаживание «левой части» линии горизонта.

Введем обозначение:  $P_k(y_k, x_k)$  –  $k$ -я точка линии горизонта,  $P_{l/2}(y_{l/2}, x_{l/2})$  – центральная точка линии горизонта.

Шаг 14.1. Примем  $k=l/2$ .

Шаг 14.2. Если не выполняется условие

$$|y_{k-1} - y_k| \leq \beta, \quad (5)$$

то осуществляется поиск ближайшей точки с ненулевой информативностью  $I_{ij} \neq 0$ , при

$$j = x_{k-1}, \text{ такой что } r \in \left[ y_k - \frac{\beta}{2}; y_k + \frac{\beta}{2} \right].$$

Шаг 14.3. Если найдена точка, удовлетворяющая условиям, заданным на шаге 14.2, то примем  $P_k(y_k, x_k) = P_{k-1}(r, x_{k-1})$ .

В противном случае вычислим  $y$ -координату точки  $P_{k-1}$  по формуле  $r = y_k - (x_k - x_{k-1}) \sin \theta$ , где  $\sin \theta = (y_{k+K} - y_k) / (x_{k+K} - x_k)$ . Если полученное значение  $r$  удовлетворяет условию (5), то примем  $P_k(y_k, x_k) = P_{k-1}(r, x_{k-1})$ . Иначе – примем  $P_k(y_k, x_k) = P_{k-1}(y_k, x_{k-1})$ .

Если  $k-1=1$ , то перейти к шагу 15, иначе – выполняется переход к шагу 14.2.

*Шаг 15.* Сглаживание «правой части» линии горизонта.

Шаг 15.1. Примем  $k=l/2$ .

Шаг 15.2. Если не выполняется условие

$$|y_{k+1} - y_k| \leq \beta, \quad (6)$$

то осуществляется поиск ближайшей точки с ненулевой информативностью  $I_{ij} \neq 0$ , при

$$j = x_{k+1}, \text{ такой что } r \in \left[ y_k - \beta/2; y_k + \beta/2 \right].$$

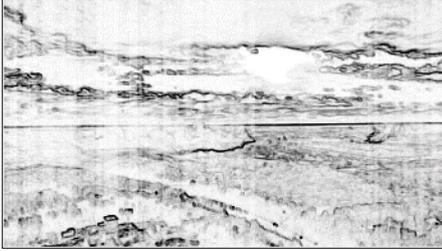
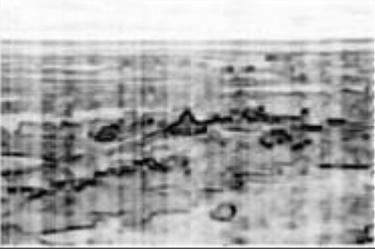
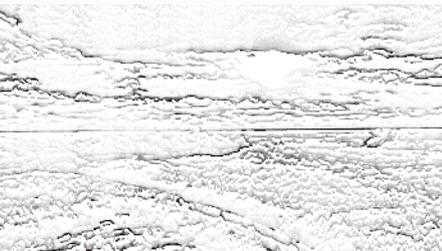
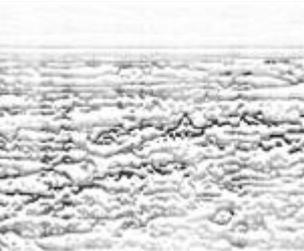
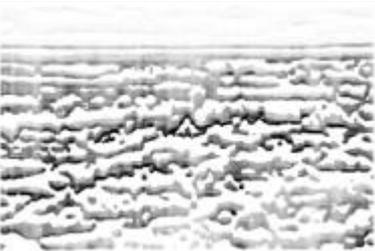
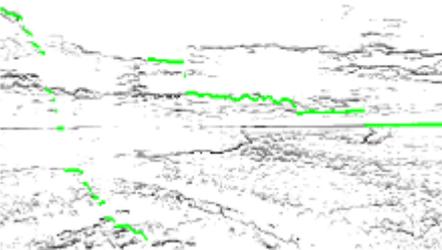
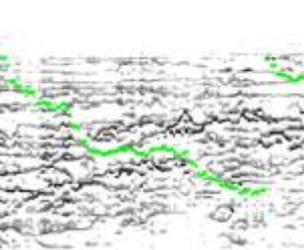
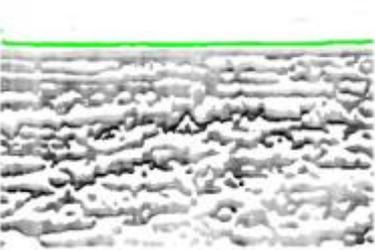
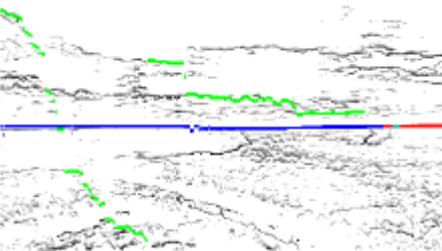
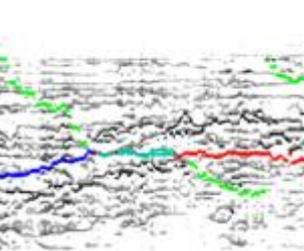
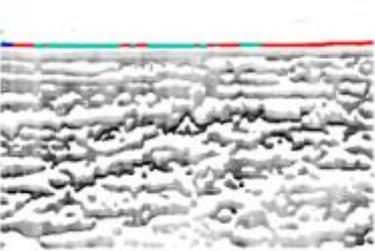
Шаг 15.3. Если найдена точка, удовлетворяющая условиям, заданным на шаге 15.2, то примем  $P_k(y_k, x_k) = P_{k+1}(r, x_{k+1})$ .

В противном случае вычислим  $y$ -координату точки  $P_{k+1}$  по формуле  $r = y_k + (x_{k+1} - x_k) \sin \psi$ , где  $\sin \psi = (y_k - y_{k-K}) / (x_k - x_{k-K})$ . Если полученное значение  $r$  удовлетворяет условию (6), то примем  $P_k(y_k, x_k) = P_{k+1}(r, x_{k+1})$ , иначе – примем  $P_k(y_k, x_k) = P_{k+1}(y_k, x_{k+1})$ .

Если  $k+1=l$ , то перейти к шагу 16. В противном случае – выполняется переход к шагу 15.2.

Шаг 16. Отображение линии горизонта на снимке (см. рис. 7 п, с).

Шаг 17. Конец.

№ Шага	 <p style="text-align: center;">а</p>	 <p style="text-align: center;">б</p>	
3	 <p style="text-align: center;">в, <math>N = 4</math></p>	 <p style="text-align: center;">г, <math>N = 5</math></p>	 <p style="text-align: center;">д, <math>N = 9</math></p>
4	 <p style="text-align: center;">е</p>	 <p style="text-align: center;">ж</p>	 <p style="text-align: center;">з</p>
7	 <p style="text-align: center;">и, <math>t = 55.08</math></p>	 <p style="text-align: center;">к, <math>t = 58.54</math></p>	 <p style="text-align: center;">л, <math>t = 13</math></p>
9-12			

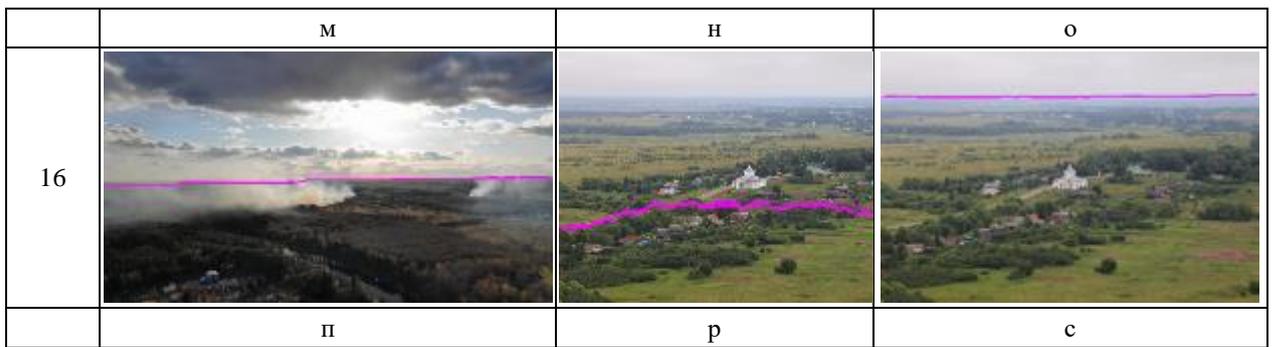


Рисунок 7 – Этапы вычисления линии горизонта на фотографиях местности

На рисунке показаны основные этапы вычисления линии горизонта в усложненных условиях. В крайнем левом столбце отображен процесс обнаружения при частичном перекрытии дымом и густой темной облачности. В среднем и крайнем столбцах представлена ситуация с размытым дальним планом снимка, требующая второй итерации вычисления линии горизонта. Из рисунка видно, что метод учитывает природные факторы линии горизонта: более светлая и гладкая текстура выше ее, более темная и сложная текстура под ней; протяженность слева направо. Исходя из указанных признаков, при поиске пикселей-кандидатов, не выполняется анализ каждой краевой точки, а совершается один проход слева-направо. Так на рис. 7 м, н, о в каждом столбце зеленым цветом отмечено не более одного кандидата. Соответственно, предложенный алгоритм обнаружения линии горизонта значительно ускоряет процесс решения поставленной задачи.

### 3. Задача обнаружения людей и транспортных средств рядом с очагом возгорания

Рассмотрим общую технологию для обнаружения и позиционирования объектов интереса. Отслеживание является одним из наиболее важных компонентов обнаружения объектов в реальных приложениях. Для реализации этой функции предлагается воспользоваться технологией DeepSORT [77].

С помощью детектора объектов (нейронной сети) определяются позиция, размер и класс отдельных объектов. Далее применяется венгерский алгоритм, чтобы связать определенные объекты с уникальными идентификаторами, которые уже были ранее найдены на предыдущих найденных объектов (на более ранних кадрах) и отслеживаются с помощью фильтров Калмана. DeepSORT позволяет улучшить точность определения и уменьшить количество переключений между объектами, когда, допустим, один человек на кадре загораживает ненадолго другого, и теперь человек, которого загородили, считается новым объектом. Для этого используется новый элемент, так называемый «внешний вид» объектов, которые появляются на кадре (appearance). В качестве источника данных об объекте может быть использована любая нейронная сеть, обученная классификации, достаточно откинуть слой классификации и использовать эти признаки. Авторы DeepSORT создали новую метрику для подсчета расстояния, которая включает в себя как предсказания с помощью фильтров Калмана, так и «косинусовое расстояние» (cosine distance). Отдельные объекты сравниваются не только с учетом позиции, но и с использованием знаний о размерах и пропорции сторон.

Использовалось следующее аппаратное обеспечение: процессор Intel Core i5 4670, графический ускоритель Nvidia GTX 1080 Ti, 24 Гб DDR-3. Базовые библиотеки: pytorch, torchvision, tensorflow, ultralytics, torchreid и deep\_sort\_realtime.

В качестве основы для экспериментов был взят пакет deep\_sort\_realtime на языке Python [78]. Эта реализация имеет следующие особенности:

- для поиска объектов используются нейронные сети из пакета Torchvision [79], в том числе fasterrcnn\_resnet50\_fpn\_v2, fasterrcnn\_resnet50\_fpn, fasterrcnn\_mobilenet\_v3\_large\_fpn, fasterrcnn\_mobilenet\_v3\_large\_320\_fpn,

fcos\_resnet50\_fpn, ssd300\_vgg16, ssdlite320\_mobilenet\_v3\_large, retinanet\_resnet50\_fpn, retinanet\_resnet50\_fpn\_v2;

- для извлечения информативных признаков из объектов предлагается использовать mobilenet\_v2 [80] (1280 признаков), torchreid [81] (512 признаков), CLIP [82] (1024 признака);

- при использовании torchreid можно также выбрать базовую нейронную сеть, обученную на датасете ImageNet [83], по умолчанию используется достаточно эффективная osnet\_ain\_x1\_0 [84], однако можно выбрать и другие, например, shufflenet, mobilenetv2 и mlfn;

- можно использовать альтернативный вариант извлечения признаков на Tensorflow и mobilenet\_v2;

- изображения, подаваемые на модуль извлечения признаков, масштабируются до 224x224 в случае mobilenet\_v2 и до 256x128 в случае torchreid.

Проведенные эксперименты, показали, что

- наиболее адекватные результаты при поиске объектов показывают нейронные сети fasterrcnn\_resnet50\_fpn\_v2, fasterrcnn\_mobilenet\_v3\_large\_fpn, при этом вторая чаще пропускает объекты, но в 3 раза быстрее; для более простых случаев с хорошими исходными данными вполне можно использовать и более быструю fasterrcnn\_mobilenet\_v3\_large\_fpn;

- наиболее адекватные результаты при извлечении признаков показывают mobilenet\_v2 и torchreid, однако, вариант на torchreid существенно медленнее. CLIP показывает примерно такие же результаты, как и torchreid;

- osnet\_ain\_x1\_0 показывает лучшие результаты при использовании torchreid, чем другие варианты;

- реализация mobilenet\_v2 на Tensorflow в разы медленнее, чем реализация в Torchvision.

Для расширения возможностей в рамках проекта был внесен ряд изменений в исходный код библиотеки deep\_sort\_realttime, в том числе

- добавлена возможность использования Yolo 8 для поиска объектов, которая в разы быстрее fasterrcnn\_resnet50\_fpn\_v2;

- добавлена возможность извлечения информативных признаков с помощью mobilenet\_v3\_small (1024 признака) и mobilenet\_v3\_large (1280 признаков) на базе Torchvision;

- входные изображения обрабатываются с помощью letterbox-предобработки, возвращающей изображения с высотой и шириной, кратными 32, при этом пропорции не искажаются, незадействованные области заполняются серым цветом;

- добавлено автоматическое масштабирование надписей, накладываемых на результат обработки (количество кадров в секунду, надписи, идентификатор трека, класс объекта);

- изображение для онлайн-просмотра результатов обработки автоматически масштабируется под заданный размер окна;

- размер изображений, подаваемых на модуль извлечения признаков, зависит от того, с каким размером изображений работает нейронная сеть, занимающаяся детекцией объектов, например, в случае, когда в кадре много объектов, через параметр imgsz задаем ориентировочные габариты входа нейронной сети (по умолчанию это 640x640), если поменять на более крупный вход, то это приводит к уменьшению размера изображений, из которых извлекаются признаки, это логично, так как предполагается, что в кадре много мелких объектов и нет смысла их растягивать;

- для ограничения числа отслеживаемых информационных векторов с данными об объектах значение параметра nn\_budget при инициализации объекта DeepSort изменено с None на 128, при существенном числе объектов в кадре можно поставить большее значение;

в случае None нагрузка на процессор при трекинге существенно возрастает, что приводит к катастрофическому падению скорости трекинга;

- обработка видеопоследовательностей стала буферизованной, по умолчанию обрабатывается по 6 кадров за раз, что весьма существенно влияет на эффективность использования графического ускорителя вычислений;

- число одновременно обрабатываемых изображений в модулях извлечения признаков на базе mobilenet и CLIP увеличено с 16 до 32;

- добавлена проверка на размеры изображений, получаемых от детектора (при тестировании однажды было получено изображение с нулевой шириной);

- при отрисовке результатов трекинга не отрисовываются изображения, вышедшие за границу кадра;

- переработан код подсчета скорости детекции и трекинга.

Результат обработки отдельных кадров (информация о найденных объектах) выдается в виде набора векторов с информацией об областях с объектами, каждый из которых имеет следующий формат:

*[идентификатор класса] [координата x левого верхнего угла] [координата y левого верхнего угла] [координата x правого верхнего угла] [координата y правого верхнего угла]*. Идентификатор класса равен «0» при локализации пешеходов с помощью нейронных сетей типа Yolo; «1» при использовании нейронных сетей из пакета Torchvision, что соответствует формату COCO-датасета.

#### 4. Проведение экспериментальных исследований

В данном разделе представлены результаты экспериментальных исследований и дана их оценка.

##### 4.1. Проведение экспериментальных исследований с отсечением горизонта

Путем исключения части изображения, содержащей облака, предлагается уменьшить количество ложных сигналов о пожаре при патрулировании территорий беспилотным летательным аппаратом. Дополнительно, информация о положении линии горизонта способствует повышению точности распознавания и определения масштабов пожара. Рассмотрим результаты работы предложенного алгоритма в условиях высокого задымления, высокой облачности, полуприродного ландшафта и горной местности (рис. 8).

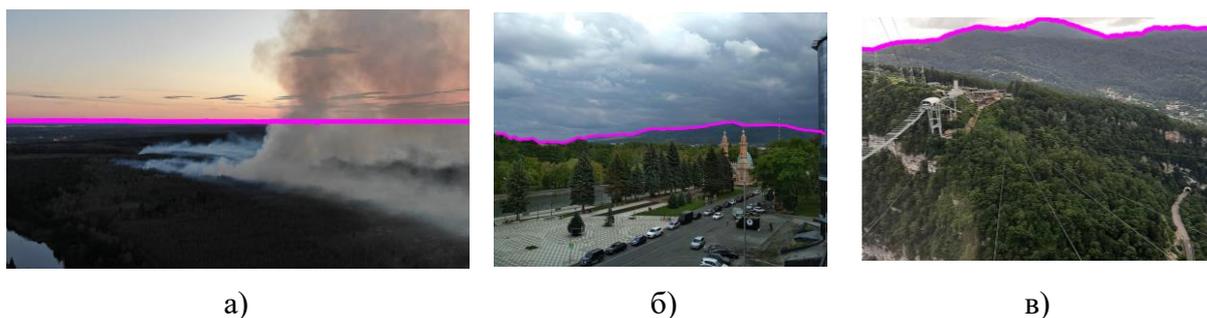


Рисунок 8 – Результаты обнаружения линии горизонта в усложненных условиях, а) задымление б) полуприродный ландшафт в) горная местность

Вычисление точек линии горизонта основано на определении локальной контрастности изображения в скользящем окне. Размер скользящего окна и порог информативности пикселей определяется автоматически индивидуально для каждого изображения. Проверка гипотезы об эффективности сегментации горизонта в качестве предварительной обработки для обнаружения лесных пожаров показала, что средняя точность без использования информации о линии горизонта составила 84.02%. После определения линии горизонта качество распознавания возросло до 95.03%. Таким образом,

применение алгоритма расчета линии горизонта повысило точность распознавания пожаров на 11.01%.

#### 4.2. Проведение экспериментальных исследований с нейронными сетями

Рассмотрим выделение очагов с пожарами на основе ИНС YOLO. Примененные нейронные сети YOLO 5s, 5m и 5l состоят из 270, 369 и 467 слоев и, соответственно, имеют более чем 7, 21 и 46 миллионов регулируемых весовых коэффициентов. В качестве основы для обучения использовалась библиотека Torch версии 1.10.2 с CUDA 11.3. Исходные изображения от UAV (датасет FLAME, файл 9) размером 3840x2160 были раздроблены на девять непересекающихся частей, каждая такая часть размером 1280x720 пикселей. Информация о локациях с пожарами извлечена из png-масок, подготовленных разработчиками исходного датасета для сегментации (FLAME, файл 10), затем с помощью алгоритма маркировки связанных компонент были извлечены данные об точках с огнем: сведения о их размерах и координатах сохраняются в текстовые файлы. Далее фрагменты снимков с информацией о точной локации места с пожаром подаются на ANN. Чтобы из одного датасета сформировать и тестовые изображения, и обучающие, выполняется разбиение обработанных данных пакетами по 90 изображений. Первый пакет идет на обучение, второй сознательно выбрасывается, третий – на тестирование, четвертый выбрасывается. Так сделано специально, чтобы обеспечить то, что тесты и обучающая выборка будут из разных по времени кадров видеопотока. В обучающей и тестовой выборках в итоге оказалось по 4500 изображений. Из них 2601 и 2604 соответственно – изображения с лесом без очагов пожаров. Отметим, что при подаче изображений на нейронную сеть идет изменение их размера до 640x640. Скорость обучения достаточно высокая, например, десять эпох обучения батчами по 16 изображений и SGD-оптимизатором заняли 105 минут. Увеличение размера батча несколько ускоряет обучение, однако требует использования большего объема памяти графического ускорителя.

Для базовой оценки способности YOLO 5s находить области с огнем обратимся к табл. 20. Эксперименты проведены для определения подходящего оптимизатора обучения и ориентировочного размера батча.

Таблица 20 – Базовая оценка способности YOLO 5s находить области с огнем

Launch options	Precision, %	Recall, %	F1-score, %	Confidence threshold, %	mAP@.5, %	mAP@.5:.95, %
--optimizer SGD --img 640 --batch 16 --epochs 10	74.368	68.508	71.32	41.5	72.497	34.680
--optimizer SGD --img 640 --batch 16 --epochs 50	76.185	71.717	73.88	43.3	75.305	39.524
--optimizer SGD --img 640 --batch 32 --epochs 10	74.815	67.736	71.10	38.1	72.890	35.730
<b>--optimizer SGD --img 640 --batch 32 --epochs 50</b>	<b>77.926</b>	<b>71.273</b>	<b>74.45</b>	<b>47.5</b>	<b>76.119</b>	<b>39.990</b>
--optimizer SGD --img 640 --batch 64 --epochs 10	72.233	66.834	68.68	39.5	70.530	33.706
--optimizer SGD --img 640 --batch 64 --epochs 50	78.151	71.020	74.42	42.4	75.618	39.825
--optimizer AdamW --img 640 --batch 16 --epochs 10	68.200	57.397	62.33	41.5	61.602	26.913

--optimizer AdamW --img 640 --batch 16 --epochs 50	73.928	66.507	70.02	41.7	71.129	34.867
--optimizer AdamW --img 640 --batch 32 --epochs 10	70.304	57.027	62.97	43.2	61.696	27.660
--optimizer AdamW --img 640 --batch 32 --epochs 50	76.320	66.172	70.88	47.4	71.366	35.433
--optimizer AdamW --img 640 --batch 64 --epochs 10	71.717	58.424	64.39	38.1	63.478	27.248
--optimizer AdamW --img 640 --batch 64 --epochs 50	75.755	67.714	71.51	43.4	72.738	36.208

Конфигурация с SGD-оптимизатором, используемым в YOLO 5s по умолчанию, и с пакетами по 32 изображения показала полноту в 71.273%, найден 1351 из 1896 очагов пожаров, 383 ложных срабатывания (в основном на дым без огня). Для отсека фоновых объектов в YOLO обязательно используется порог по уверенности. В данном случае он равен 47.5%.

Чтобы оценить возможность дальнейшего улучшения результатов, число эпох обучения было увеличено с 50 до 300. Результаты экспериментов с нейронными сетями YOLO 5s, 5m и 5l представлены в табл.21.

Таблица 21 – Результаты экспериментов по применению YOLO для задачи поиска очагов возгораний

ANN	Precision, %	Recall, %	F1-score, %	Confidence threshold, %	mAP@.5, %	mAP@.5:.95, %	Processing speed, frames/s
5s	79.440	71.123	75.05	51.5	76.691	40.705	49
5m	79.308	74.345	76.75	44.4	77.899	41.274	45
5l	79.526	72.253	75.72	49.7	76.846	41.834	35

### 4.3. Примеры обнаружения транспортных средств и людей в видеопотоке

В качестве датасета для тестирования использовалась выборка на основе бенчмарков TLP [85], DIVOTrack [86], MvMhAT [87] и с сайта Pixabay [88].

В зависимости от характера видео менялась модель для трекинга (параметр model), модуль извлечения признаков (параметр embedder), отслеживаемые классы (параметр cls), порог уверенности сети (threshold), масштабирующий коэффициент, определяющий размер входа нейронной сети и размер изображений при извлечении признаков (параметр imgsiz). Наиболее сложный случаи – файлы Park\_View1.mp4 и Square\_View1.mp4, где для успешного трекинга пришлось использовать yolov8x, самую сложную из Yolo 8, и масштабирующий коэффициент imgsiz был взят равным 960, из-за большого числа мелких объектов. В остальных случаях он был взят равным 640. В табл. 22 приведены итоговые скорости обработки файлов.

Таблица 22 – Результаты экспериментов

Датасет	Файл	Модуль детекции объектов	Модуль извлечения признаков	Скорость обработки, к/с
MvMhAT	video_traffic_1.mp4	yolov8l	mobilenet_v3_small	12.19
	video_traffic_2.mp4	yolov8l	mobilenet_v3_small	18.73
	mvmhat_1_1.mp4	yolov8m	torchreid	17.39
	mvmhat_1_4.mp4	yolov8m	torchreid	16.36
	video_traffic_1.mp4	fasterrcnn_resnet50_fpn_v2	mobilenet_v3_small	5.13
	video_traffic_2.mp4	fasterrcnn_resnet50_fpn_v2	mobilenet_v3_small	5.44
	mvmhat_1_1.mp4	fasterrcnn_resnet50_fpn_v2	torchreid	5.10

	mvmhat_1_4.mp4	fasterrcnn_resnet50_fpn_v2	torchreid	5.11
DIVOTrack	Park_View1.mp4	yolov8x	torchreid	12.48
	Park_View2.mp4	yolov8m	torchreid	15.18
	Park_View3.mp4	yolov8m	torchreid	17.40
	Square_View1.mp4	yolov8x	torchreid	10.28
	Square_View2.mp4	yolov8m	torchreid	13.31
	Square_View3.mp4	yolov8m	torchreid	15.40
TLP	TinyTLP_Drone2.mp4	yolov8m	mobilenet_v3_small	31.34
Pixabay	motorcycle_-_57844 (1080p).mp4	yolov8m	mobilenet_v3_small	35.91

Разработанная технология позволяет обнаруживать и «вести» объект интереса (человек, техника). Когда объект обнаружен, рамка его обрисовывает и далее на последовательности кадров сопровождает с указанием выданного объекту ID (идентификатора). Вследствие отсутствия у исполнителя видеоклипов с пожарами, людьми и техникой одновременно, для экспериментов были взяты в основном, видео, содержащие технику и людей на фоне некоторой растительности.

Рассмотрим некоторые примеры результатов обработки видеоклипов (см. рис. 9-11).

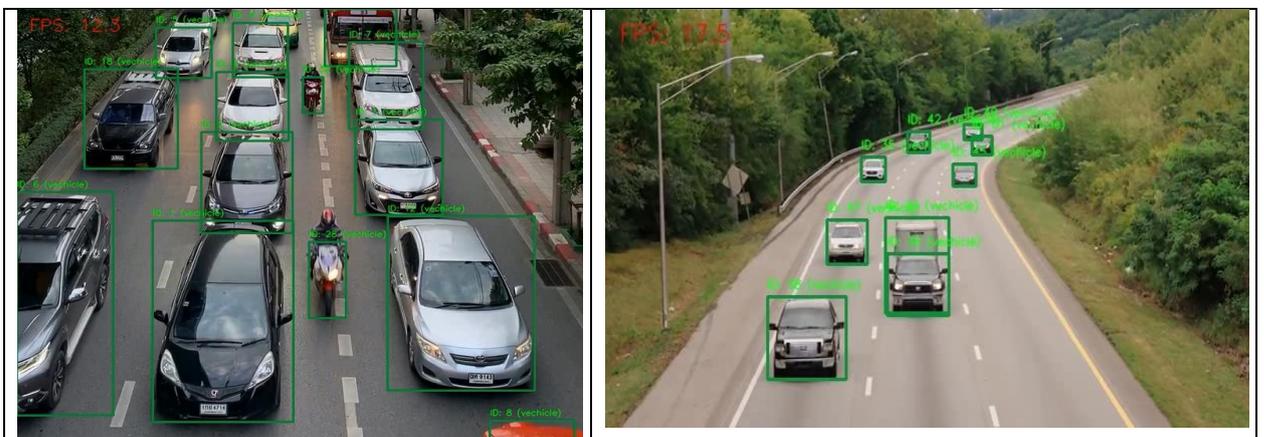


Рисунок 9 – Обнаружение и отслеживание движения техники

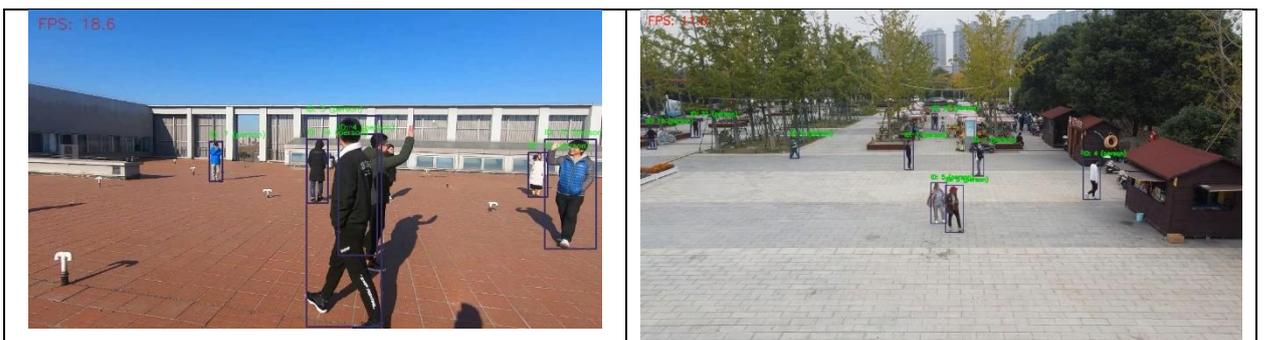


Рисунок 10 – Обнаружение и отслеживание движения людей



Рисунок 11 – Обнаружение и отслеживание движения людей и транспортных средств

Эксперименты показали, что нет принципиального различия между результатами извлечения признаков с использованием mobilenet\_v2, mobilenet\_v3\_small и mobilenet\_v3\_large, однако, mobilenet\_v3\_small работает на 20% быстрее. Алгоритм DeepSort реализуется непосредственно на процессоре, поэтому скорость трекинга существенно зависит от его характеристик. Проведенные доработки библиотеки позволили существенно поднять скорость детектирования и трекинга целевых объектов. Применение Yolo 8 позволило существенно поднять скорость обработки изображений по сравнению с исходной версией. Представленная технология позволит на последующем этапе выполнения проекта решить задачи взаимодействия БПЛА с объектами, находящимися в зоне пожаров и задымлений.

#### 4.4. Экспериментальные исследования по распознаванию транспортных средств и их номеров

Поддерживаются следующие виды транспортных средств: «*велосипед*», «*легковой автомобиль*», «*мотоцикл*», «*автобус*» и «*грузовик*». При использовании Yolo эти классы транспортных средств имеют идентификаторы «1», «2», «3», «5» и «7», при использовании нейронных сетей из пакета Torchvision – «2», «3», «4», «6», «8», что соответствует формату COCO-датасета.

Выполнены исследования по обнаружению и дешифровке номеров на легковых автомобилях, автобусах и грузовиках. На основе библиотеки с открытыми кодами tflite\_avto\_num\_recognition [89] удалось получить результаты, представленные ниже в табл. 23.

Программный код написан на языке Python с использованием двух моделей Tensorflow Lite для высокопроизводительной обработки целочисленных данных как на маломощных процессорах, так и на встроенной графике мобильных процессоров с архитектурой ARM и др.

В качестве данных для тренировки нейронной сети, используемой для поиска номеров, использован набор данных о парковке в китайском городе CCPD [90], а также 600 фотографий из находящегося в открытом доступе датасета StopNumbers. Для тренировки была выбрана модель ssd\_resnet50\_v1\_fpn. Для сохранения модели была использована программа export\_tflite\_graph\_tf2.

В качестве модели для распознавания номера был взят доработанный вариант модели ANPR.

Пошаговый алгоритм:

- 1) загружаем дообученную tflite-модель для нахождения расположения номера машины, выделяем область с номером (см. рис.12);
- 2) делаем преобразование Canny;
- 3) получаем преобразование Хафа из результатов Canny;
- 4) используя функцию hough\_line\_peaks, получаем необходимый угол разворота;
- 5) увеличиваем контрастность;
- 6) загружаем обученную tflite модель CNN-LSTM-CTC для распознавания символов;
- 7) распознаем символы, используя информацию о пробелах в качестве разделителей.





Рисунок 12 – Фрагмент с номером до и после обработки

Таблица 23 – Результаты экспериментов с обнаружением и распознаванием автомобильных номеров

№	Изображение	Реальный номер	Распознанный номер
0		X209OH150	X209OH150
1		B451PX799	B451PX799
2		B451PX799	B451PX799

3		B451PX799	B451PX797
4		B451PX799	451XPX799
5		B827PY763	B827PY763
6		B827PY763	B827PY763
7		H437KE69	H437KE69

8		K221PH73	K221PH73
9		B827PY763	B827PY763
10		H437KE69	B437KE69
11		K221PH73	K221PH73
12		K221PH73	K221PH73

В целом полученное программное обеспечение позволило выполнить поставленные задачи. Распознавание не требует больших вычислительных ресурсов и может эффективно выполняться на обычных процессорах.

## **5. Рекомендации по применению методов обнаружения пожаров и задымлений**

Задача обнаружения пожаров в лесах Ярославской области представляется крайне актуальной, учитывая негативные последствия на социо-эколого-экономические сферы. Важность проблемы особо подчеркивалась в выступлениях губернатора Ярославской области, по поручению которого на вышках сотовой связи были дополнительно установлены видеокамеры и видеозаписывающие устройства для усиления мониторинга за пожароопасной обстановкой в лесах. Решение проблемы раннего выявления возгораний в режиме реального времени требует разработки новых эффективных методов автоматического анализа фото- и видеопотоков, поступающих со стационарных и мобильных камер.

Научно-обоснованные методические рекомендации по применению разработанных алгоритмов и методов предназначены, в первую очередь, для учреждений, ответственных за пожаробезопасность Ярославской области, включая ФГБУ «Национальный парк «Плещеево озеро», ГКУ ЯО «Переславское лесничество», Департамент лесного хозяйства Ярославской области и другие заинтересованные организации региона. В силу универсальности рекомендации могут быть использованы разработчиками систем обработки данных технического зрения.

На основании выполненных научных исследований даются следующие рекомендации:

1. В качестве методологического обеспечения систем раннего обнаружения возгораний предлагаются:

- 1) методы и алгоритмы мониторинга и обнаружения возгораний, нарушений экологического состояния наблюдаемой территории стационарными и мобильными средствами технического зрения;
- 2) программное обеспечение высокопроизводительной обработки данных дистанционного зондирования в автоматическом режиме, обеспечивающее позиционирование мест и размеров возгораний;
- 3) методы обнаружения людей и транспортных средств, оказавшихся рядом с очагом возгорания.

2. Для улучшения качества алгоритмов поиска объектов системами технического зрения предлагаются

- 1) методы устранения смазов и фильтрации изображения, позволяющие нивелировать искажения, вызванные скачками подвижных камер из-за ветровых нагрузок;
- 2) алгоритм устранения горизонта, за счет которого ускоряется анализ снимков и нивелируется влияние атмосферных эффектов.

3. Для ускорения процессов обработки потоковой информации предлагается использовать

- 1) многоядерные и многопроцессорные вычислители, что позволяет достичь режимов реального времени или близких к ним;
- 2) графические ускорители, встраиваемые непосредственно в вычислители наземных станций наблюдения.

4. Для установки на наземных станциях: экспериментальное программное обеспечение раннего обнаружения задымлений и возгораний на лесных территориях на основе средств высокопроизводительной обработки информации со стационарных и подвижных видеокамер.

## Заключение

В настоящем Препринте представлены методологические основы для решения задач обнаружения очагов возгорания на основе средств технического зрения в масштабе времени, близком к реальному. Проведен обзор состояния предметной области, представлены методы улучшения изображений, алгоритмы обнаружения огня и дыма на изображениях и видеопотоках, полученных, в том числе, с БПЛА. Большое внимание уделено используемым тестовым наборам данных, типам современных нейронных сетей, обрабатываемых изображениям, обобщению результатов мировых исследований в данной области.

Значительная часть исследований по данной тематике основана на применении сверточных нейронных сетей с использованием модели глубокого обучения. Используются реализации на основе эффективных и хорошо зарекомендовавших себя архитектурах СНС. Выполнено сравнительное тестирование существующих архитектур и предлагаются модификации, способствующие повышению качества их работы. Применение подходов на основе современных нейросетевых моделей глубокого обучения позволяет строить эффективные системы семантической сегментации изображений, демонстрируются высокие показатели точности и полноты в задачах классификации. В ряде исследований, при решении рассматриваемых задач, нейросетевые классификаторы обеспечивают показатели качества в диапазоне 80-99%. Проведенные эксперименты показали эффективность применения нейронных сетей Xception v1 и LW-FIRE для решения задачи классификации потока кадров от БПЛА. При этом обеспечивается F1-мера вплоть до 85.7%. В задаче выделения очагов возгораний хорошие результаты демонстрирует нейронная сеть YOLO 5m, показывающая F1-меру 76.75% в сочетании со скоростью обработки 45 кадров в секунду.

Научная значимость проведенных исследований заключается в создании соответствующей мировому уровню интеллектуальной технологии раннего обнаружения и предупреждения пожаров, основанной на комплексном использовании методов интеллектуального анализа потоков данных от систем технического зрения.

Практическая значимость проекта заключается в создании методологических основ и программного обеспечения системы обнаружения очагов возгораний лесных территорий с оповещением об опасности лиц, принимающих решения. Экономический эффект достигается за счет минимизации экономических и экологических потерь, оперативного обнаружения и поддержки принятия решений по ликвидации нештатных ситуаций, связанных с пожароопасной ситуацией.

Исследования были выполнены на основе фото- и видеоматериалов, предоставленных СГБУ ЯО «Лесная охрана» и Национальным парком «Плещеево озеро». В исследованиях применяются последние достижения в области искусственного интеллекта и машинного обучения, в том числе используются как оригинальные, так и доработанные нейронные сети. Решена задача обработки видеопотока в режиме реального времени с использованием наземной станции, оснащенной графическим ускорителем вычислений. Полученные результаты были доложены на Фестивале академической науки (с. Вятское Ярославской области, 19 мая 2023 г.). Материалы были представлены на Форумах Армия-2022 и Армия-2023, на Конгрессе молодых ученых 2023 года и получили освещение в СМИ Ярославской области.

Исследование выполнено за счет гранта Российского научного фонда № 22-11-20001, <https://rscf.ru/project/22-11-20001/> и гранта в форме субсидии из областного бюджета организациям Ярославской области.

Коллектив авторов благодарит за поддержку Проекта РФФ № 22-11-20001 Российский Научный Фонд, Правительство Ярославской области, Министерство инвестиций и промышленности Ярославской области, профильные организации: ФГБУ «Национальный парк «Плещеево озеро», Департамент лесного хозяйства Ярославской области, ГКУ ЯО «Переславское лесничество».

## Литература

1. V.Parthipan, D.Dhanasekaran. Preventing and Monitoring of Framework for Forest Fire Detection and Data Analysis Using Internet of Things (IoT). *International Journal of Engineering and Advanced Technology (IJEAT)* ISSN: 2249-8958 (Online), Volume-8 Issue-3S, February 2019, p.691-695.
2. Sidhant Goyal, MD Shagill, Arwinder Kaur, Harpreet Vohra, Ashima Singh. A YOLO based Technique for Early Forest Fire Detection. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020. p.1357-1362.
3. Zhen Wang(B), Huidan Zhang, Muxin Hou, Xiaoting Shu, Jianguo Wu, and Xiaoqian Zhang School of Electrical and Energy, Engineering Nantong Institute of Technology, Nantong, Jangsu, China A Study on Forest Flame Recognition of UAV Based on YOLO-V3 Improved Algorithm. – Springer Nature Singapore Pte Ltd. 2021 K. Li et al. (Eds.): LSMS 2021/ICSEE 2021, CCIS 1468, pp. 497–503.
4. Cheng Y., Chen K., Bai H. and etc. “An Efficient Fire Detection Algorithm Based on Multi-Scale Convolutional Neural Network”. *Fire and Materials*, 2021, 12 p. DOI: 10.1002/fam.3045.
5. Pradeep Kumar G, Rahul R, Ravindharan N. Early Forest Fire Detection Using Machine Learning Algorithms. *International Journal of New Technology and Research (IJNTR)* ISSN: 2454-4116, Volume-7, Issue-4, April 2021, pp. 1-5.
6. Henry Cruz(B), Tatiana Gualotuña1, María Pinillos, Diego Marcillo, Santiago Jácome, and Efraín R. Fonseca C. Machine Learning and Color Treatment for the Forest Fire and Smoke Detection Systems and Algorithms. *CIT 2020, AISC 1326*, pp. 109–120, 2021. [https://doi.org/10.1007/978-3-030-68080-0\\_8](https://doi.org/10.1007/978-3-030-68080-0_8).
7. Фраленко В.П. Нейросетевые методы обнаружения возгораний в лесных массивах. – *Искусственный интеллект и принятие решений*, №1, 2023, с.67-77. DOI: 10.14357/20718594230107 (RSCI).
8. Geetha S., Abhishek C.S., Akshayanat C.S. Machine Vision Based Fire Detection Techniques: A Survey. *Fire Technology*, 2021, Vol.57, pp.591-623. DOI: 10.1007/s10694-020-01064-z
9. Ciprián-Sánchez J.F., Ochoa-Ruiz G., Rossi L. and etc. “Assessing the Impact of the Loss Function, Architecture and Image Type for Deep Learning-Based Wildfire Segmentation”. *Appl. Sci.*, 2021, Vol.11(15), 7046. DOI: 10.3390/app11157046
10. Shubhangi Chaturvedi, Pritee Khanna, Aparajita Ojha “A Survey on Vision-Based Outdoor Smoke Detection Techniques for Environmental Safety”. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2022, Vol.185, pp.158-187. DOI: 10.1016/j.isprsjprs.2022.01.013.
11. Favorskaya M.N. Early Smoke Detection in Outdoor Space: State-of-the-Art, Challenges and Methods. In: Virvou, M., Tsihrintzis, G.A., Jain, L.C. (eds). *Advances in Selected Artificial Intelligence Areas. Learning and Analytics in Intelligent Systems*, 2022, Vol.24. Springer, Cham. DOI: 10.1007/978-3-030-93052-3\_8
12. Huo Y., Zhang Q., Jia Y. and etc. “A Deep Separable Convolutional Neural Network for Multiscale Image-Based Smoke Detection”. *Fire Technology*, 2022. DOI: 10.1007/s10694-021-01199-7.
13. Miao J., Zhao G., GaoY. and etc. “Fire Detection Algorithm Based on Improved YOLOv5”. 2021 International Conference on Control, Automation and Information Sciences (ICCAIS), 2021, pp.776-781. DOI: 10.1109/ICCAIS52680.2021.9624619
14. Li Y., Zhang W., Liu Y. and etc. “A Visualized Fire Detection Method Based on Convolutional Neural Network Beyond Anchor”. *Applied Intelligence*, 2022. DOI: 10.1007/s10489-022-03243-7.

15. Wang S., Zhao J., Ta N. and etc. "A Real-Time Deep Learning Forest Fire Monitoring Algorithm Based on an Improved Pruned + Kd Model". *J Real-TimeImage Proc* 18, 2021, pp.2319-2329. DOI: 10.1007/s11554-021-01124-9.
16. Wang S. and etc. "Forest Fire Detection Based on Lightweight Yolo" 2021 33rd Chinese Control and Decision Conference (CCDC), 2021, pp.1560-1565, DOI: 10.1109/CCDC52312.2021.9601362.
17. Yiheng Cai, Yajun Guo, Yuanyuan Li and etc. "Fire Detection Method Based on Improved Deep Convolution Neural Network". In *Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition (ICCP '19)*, 2019. Association for Computing Machinery, New York, NY, USA, pp.466-470. DOI: 10.1145/3373509.3373570.
18. Athanasios N., Themistocleous M., Kalabokidis K. and etc. "Big Data Analysis in UAV Surveillance for Wildfire Prevention and Management". Themistocleous, M., Rupinoda Cunha, P. (eds) *Information Systems. EMCIS 2018. Lecture Notes in Business Information Processing*, 2019, Vol.341. Springer, Cham. DOI: 10.1007/978-3-030-11395-7\_5.
19. Ghali R., Akhloufi M.A., Mseddi W.S. Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation. *Sensors*, 2022, Vol.22(5), 1977. DOI: 10.3390/s22051977.
20. Wu B., Xu C., Dai X. and etc. *Visual Transformers: Token-based Image Representation and Processing for Computer Vision*, 2020. DOI: 10.48550/arXiv.2006.03677. URL: <https://arxiv.org/abs/2006.03677>.
21. Andrija Vidović, Tomislav Mihetec, Bo Wang, Igor Štimac. *Challenges and Solutions for Urban UAV Operations // International Scientific Conference "Science and Traffic Development"*, 2018, 10 p.
22. Abramov N.S., Talalayev A.A., Fralenko V.P., Shishkin O.G., Khachumov V.M. Neural network technology to search for targets in remote sensing images of the Earth // *CEUR Workshop Proceedings*. Vol. 2391. *Proceedings of the V International conference Information Technology and Nanotechnology. Session Image Processing and Earth Remote Sensing*. Samara, Russia, May 21-24, 2019, pp.180-186.
23. H. D. K. Motlagh, F. Lotfi, H. D. Taghirad and S. B. Gerami, "Position Estimation for Drones based on Visual SLAM and IMU in GPS-denied Environment", 2019 7th International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 2019, pp.120-124.
24. Y. Kim, H. Bang. Vision-Based Navigation for Unmanned Aircraft Using Ground Feature Points and Terrain Elevation Data // *Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering*, 2017 DOI: 10.1177/0954410017690548
25. P. Kumar, S. Sonkar, A. K. Ghosh and D. Philip, "Real-time vision-based tracking of a moving terrain target from Light Weight Fixed Wing UAV using gimbal control", 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), Prague, Czech Republic, 2020, pp.154-159.
26. J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, C. Bouman. Multi-Target Detection and Tracking from a Single Camera in Unmanned Aerial Vehicles (UAVs) // 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon Convention Center (October 9-14, 2016, Daejeon, Korea).
27. Фраленко В.П. Повышение качества видеопотока от системы технического зрения беспилотного летательного аппарата. – *Программные системы: теория и приложения*, №2, 2023, с.3-26. DOI: 10.25209/2079-3316-2023-14-2-3-26 (RSCI).
28. Датасет GoPro. URL: <https://paperswithcode.com/dataset/gopro> (дата обращения: 23.04.2024).
29. Awesome-Deblurring. URL: <https://github.com/subeeshvasu/Awesome-Deblurring> (дата обращения: 23.04.2024).

30. Test-time Local Converter (TLC). URL: <https://github.com/megvii-research/TLC> (дата обращения: 23.04.2024).
31. Multi-Axis MLP. URL: <https://github.com/google-research/maxim> (дата обращения: 23.04.2024).
32. EFNet. URL: <https://github.com/AHupuJR/EFNet> (дата обращения: 23.04.2024)
33. Learning degradation. URL: [https://github.com/dasongli1/Learning\\_degradation](https://github.com/dasongli1/Learning_degradation) (дата обращения: 23.04.2024).
34. Deep Generalized Unfolding (DGU). URL: <https://github.com/MC-E/Deep-Generalized-Unfolding-Networks-for-Image-Restoration> (дата обращения: 23.04.2024).
35. NAFNet width32. URL: <https://github.com/megvii-research/NAFNet> (дата обращения: 23.04.2024).
36. Stripformer. URL: <https://github.com/pp00704831/Stripformer> (дата обращения: 23.04.2024).
37. Uformer. URL: <https://github.com/ZhendongWang6/Uformer> (дата обращения: 23.04.2024).
38. Thakur A., Papakipos Z., Clauss C., Hollinger C., Andolina I. M., Boivin V., enarche-ahn, freol35241, Lowe B., Schoentgen M., Bouckenoghe R. abhiTron-ix/vidgear: VidGear v0.2.6. 2022. URL: <https://zenodo.org/records/6926196>
39. Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, Jia-Bin Huang “Hybrid Neural Fusion for Full-frame Video Stabilization”. 2021. arXivarXiv: 2102.06205
40. Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, Bing Zeng. MeshFlow: Minimum Latency Online Video Stabilization, Proceedings of the ECCV 2016. Lecture Notes in Computer Science, vol. 9910, 2016, pp. 800-815.
41. Grundmann M., Kwatra V., Essa I. Auto-directed video stabilization with robust L1 optimal camera paths // Proceedings of the CVPR 2011. 2011. pp. 225-232.
42. Grundmann M., Kwatra V., Essa I. Auto-directed video stabilization with robust L1 optimal camera paths // Proceedings of the CVPR 2011. 2011. pp. 225-232.
43. Video stabilization using homography transform. URL: [https://github.com/SergejVolkov/video\\_smoothing](https://github.com/SergejVolkov/video_smoothing) (дата обращения: 23.04.2024).
44. Yufei Xu, Jing Zhang, Stephen J. Maybank, Dacheng Tao. DUT: Learning Video Stabilization by Simply Watching Unstable Videos, 2022. arXivarXiv: 2011.14574.
45. Jinsoo Choi, In SoKweon. Deep Iterative Frame Interpolation for Full-frame Video Stabilization, 2019. arXivarXiv 1909.02641.
46. Wang M., Yang G.-Y., Lin J.-K., Zhang S.-H., Shamir A., Lu S.-P., Hu S.-M.. Deep online video stabilization with multi-grid warping transformation learning, IEEE Transactions on Image Processing, vol. 28, 2019, pp. 2283-2292.
47. OpenCV. URL: <https://learnopencv.com/video-stabilization-using-point-feature-matching-in-opencv/> (дата обращения: 23.04.2024).
48. Abramov N., Talalaev A., Fralenko V. Methods of Solution to the Task on Early Detection of Fire Outbreaks Based on Images and Video Streams from Controlled Territories // Proceedings of the 6th Computational Methods in Systems and Software 2022 (CoMeSySo 2022) (October 13-15, 2022), Vol.2. In: Lecture Notes in Networks and Systems (LNNS, Vol.597), 2023, pp.75-86. DOI: 10.1007/978-3-031-21438-7\_6.
49. Ultralytics YOLO v8.1.47, 2024. URL: <https://github.com/ultralytics/ultralytics> (дата обращения: 23.04.2024).
50. Lv W., Zhao Y., Xu S. and etc. “DETRs Beat YOLOs on Real-time Object Detection”, 2023, arXiv:2304.08069. DOI: 10.48550/arXiv.2304.08069.
51. Tsai, F.-J.; Peng, Y.-T.; Lin, Y.-Y.; Tsai, C.-C.; Lin, C.-W. Stripformer: Strip Transformer for Fast Image Deblurring. In Computer Vision – ECCV 2022; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Lecture Notes in Computer Science; Springer Nature Switzerland: Cham, 2022; Vol. 13679, pp. 146–162.

52. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P. The Flame Dataset: Aerial Imagery Pile Burn Detection Using Drones (UAVS). 2021. URL: <https://iee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs> (дата обращения: 23.04.2024).
53. Shuo Zhang, Shengbing CheZhen and LiuXu Zhang “A real-time and lightweight traffic sign detection method based on ghost-YOLO” // *Multimedia Tools and Applications* 82(9):1-25, 2023. DOI: 10.1007/s11042-023-14342-z.
54. RT-DETR-l object detection model. URL: <https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/rt-detr/rt detr-1.yaml> (дата обращения: 23.04.2024).
55. Image classification from scratch, 2020. URL: [https://keras.io/examples/vision/image\\_classification\\_from\\_scratch/](https://keras.io/examples/vision/image_classification_from_scratch/) (дата обращения: 23.04.2024).
56. Thomson W., Bhowmik N., Breckon T.P. Efficient and Compact Convolutional Neural Network Architectures for Non-temporal Real-time Fire Detection, 2020. URL: <https://arxiv.org/pdf/2010.08833.pdf>. DOI: 10.48550/arXiv.2010.08833.
57. Akagic A., Buza E. LW-FIRE: A Lightweight Wildfire Image Classification with a Deep Convolutional Neural Network. *Appl. Sci.*, 2022, Vol.12(5), 2646. DOI: 10.3390/app12052646.
58. Public API for tf.keras.applications namespace, 2024. URL: [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications](https://www.tensorflow.org/api_docs/python/tf/keras/applications) (дата обращения: 23.04.2024).
59. Фраленко В.П. Экспериментальное исследование возможностей нейронной сети типа «Darknet» на задаче обработки снимков дистанционного зондирования // *Авиакосмическое приборостроение*. 2017. №6. с.44-52.
60. Yaowen Hu, Jialei Zhan, Guoxiong Zhou and etc. “Fast Forest Fire Smoke Detection Using MVMNet”. *Knowledge-Based Systems*, 2022, Vol.241, 20 p. DOI: 10.1016/j.knosys.2022.108219.
61. Емельянова Ю.Г. Алгоритм нахождения линии горизонта на снимках, полученных с камеры беспилотного летательного аппарата. – *Авиакосмическое приборостроение*, №1, 2023, с.40-53. DOI: 10.25791/aviakosmos.1.2023.1318.
62. Богущ Р.П., Абламейко С.В. Алгоритм обнаружения дыма лесного пожара на видеоизображении // *Журнал Белорусского государственного университета. Математика. Информатика*. 2021. № 1. С. 91–101. DOI: 10.33581/2520-6508-2021-1-91-101.
63. Xu Y., Han N. Smoke Detection Algorithm Based on Wavelet Transformation and Energy Analysis // *Proc. of the 2012 National Conference on Information Technology and Computer Science (CITCS 2012)*. 2012. Atlantis Press. P. 410–413. ISBN 10.2991/citcs.2012.164. ISSN 1951-6851. DOI: 10.2991/citcs.2012.164.
64. Mukhiddinov M., Abdusalomov A.B., Cho J. A. Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5 // *Sensors*. 2022. Vol. 9384. Iss. 22. P 1–25. DOI: 10.3390/s22239384.
65. Wang C., Grau A., Guerra E., Shen Z., Hu J., Fan H. Semi-supervised wildfire smoke detection based on smoke-aware consistency // *Front. Plant Sci*. 2022. Vol. 980425. Iss. 13. P. 1–17. DOI: 10.3389/fpls.2022.980425.
66. Liu Y.J., Chiu C.C., Yang J.H. A Robust Vision-Based Skyline Detection Algorithm Under Different Weather Conditions // *IEEE Access*. 2017. Vol. 5. P. 22992–23009. DOI: 10.1109/ACCESS.2017.2728826.
67. Guo F., Mai Y., Tang J., Huang Y., Zhu L. Robust and Automatic Skyline Detection Algorithm Based on MSSDN // *Journal of Advanced Computational Intelligence and Intelligent Informatics*. 2020. Vol. 24. Iss. 6. P. 750–762. DOI: 10.20965/jaciii.2020.p0750.

68. Martinez-Sanchez L., Borio D., d'Andrimont R., Velde M. Skyline variations allow estimating distance to trees on landscape photos using semantic segmentation // *Ecological Informatics*. 2022. Vol. 70. P. 1–16. DOI: 10.1016/j.ecoinf.2022.101757.
69. Ahmad T., Emami E., Cadik M., Bebis G. Resource Efficient Mountainous Skyline Extraction using Shallow Learning // *Proc. of the International Joint Conference on Neural Networks (IJCNN)*. Hoffman Estates: Institute of Electrical and Electronics Engineers. 2021. P. 1–9. ISBN 978-1-6654-3900-8
70. Lin C., Chen W., Zhou H. Multi-Visual Feature Saliency Detection for Sea-Surface Targets through Improved Sea-Sky-Line Detection // *Journal of Marine Science and Engineering*. 2020. Vol 8. Iss. 10. P. 1–16. DOI: 10.3390/jmse8100799.
71. Zhu D., Wan L., Gao W. Fusion Method Evaluation and Classification Suitability Study of Wetland Satellite Imagery // *Earth Sciences Research Journal*. 2019. Vol. 23. No. 4. P. 339–346. DOI: 10.15446/esrj.v23n4.84350.
72. Вичевская Ю.А., Мурынов А.И. Структурный анализ изображений на основе использования функции информативности // *Альманах современной науки и образования*. Тамбов: Грамота. 2010. Т. 35. № 4. С. 53–55.
73. Тымчук А.И. О выборе уровней серого в задаче текстурной сегментации изображений на основе матриц яркостной зависимости // *Кибернетика и программирование*. 2018. № 3. С. 1–9. DOI: 10.25136/2306-4196.2018.3.26170.
74. Zhurbin I.V., Vazenova A.I., Shaura A.S., Zlobina A.G. Determining the Sliding Window Size and the Optimal Number of Clusters in the Problem of Texture Segmentation of Multispectral Aerial Photography Data // *Chemical Physics and Mesoscopy*. Udmurt Federal Research Center. 2020. Vol. 22. № 4. P. 434–447. DOI: 10.15350/17270529.2020.4.41.
75. Максимова Т.Г., Попова И.Н. Эконометрика: учебно-методическое пособие / Т.Г. Максимова, И.Н. Попова. СПб.: Университет ИТМО. 2018. – 70 с.
76. Емельянова Ю.Г. Алгоритмическое и программное обеспечение человеко-машинных интерфейсов с когнитивно-графическим отображением информации для систем космического назначения: дис. ... канд. техн. наук. Москва. 2019. 168 с.
77. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric, 2017. URL: <https://arxiv.org/abs/1703.07402>.
78. deep\_sort\_realtime. Github, 2024. URL: [https://github.com/levan92/deep\\_sort\\_realtime](https://github.com/levan92/deep_sort_realtime) (дата обращения: 23.04.2024).
79. Torchvision, 2024. URL: <https://pytorch.org/vision/stable/index.html> (дата обращения: 23.04.2024).
80. mobilenet\_v2, 2024. URL: [https://pytorch.org/vision/main/models/generated/torchvision.models.mobilenet\\_v2.html](https://pytorch.org/vision/main/models/generated/torchvision.models.mobilenet_v2.html) (дата обращения: 23.04.2024).
81. Zhou K., Yang Y., Cavallaro A. и др. Omni-scale feature learning for person re-identification, 2019. URL: <https://arxiv.org/abs/1905.00953> (дата обращения: 23.04.2024).
82. Radford A., Kim J.W., Hallacy C. и др. Learning transferable visual models from natural language supervision, 2021. URL: <https://arxiv.org/abs/2103.00020> (дата обращения: 23.04.2024).
83. Model Zoo, 2024. URL: [https://kaiyangzhou.github.io/deep-person-reid/MODEL\\_ZOO.html](https://kaiyangzhou.github.io/deep-person-reid/MODEL_ZOO.html) (дата обращения: 23.04.2024).
84. Zhou K., Yang Y., Cavallaro A. и др. Omni-scale feature learning for person re-identification, 2019. URL: <https://arxiv.org/abs/1905.00953> (дата обращения: 23.04.2024).
85. TLP, 2024. URL: <https://amoudgl.github.io/tlp/> (дата обращения: 23.04.2024).
86. DIVOTrack, 2024. URL: <https://github.com/shengyuhao/DIVOTrack> (дата обращения: 23.04.2024).

87. MvMHAT, 2024. URL: <https://github.com/realgump/MvMHAT> (дата обращения: 23.04.2024).
88. Pixabay, 2024. URL: <https://pixabay.com> (дата обращения: 23.04.2024).
89. tflite\_avto\_num\_recognition. URL: [https://github.com/sovse/tflite\\_avto\\_num\\_recognition](https://github.com/sovse/tflite_avto_num_recognition) (дата обращения: 23.04.2024).
90. CCPD Dataset. URL: <https://github.com/detectRecog/CCPD> (дата обращения: 23.04.2024).

## **ПРЕПРИНТ**

### **Методологическое обеспечение для решения задач обнаружения очагов возгорания на основе средств технического зрения в масштабе времени, близком к реальному**

#### **АВТОРЫ:**

<sup>1)</sup> к.т.н. Абрамов Н.С., <sup>1)</sup> к.т.н. Емельянова Ю.Г. <sup>1)</sup> к.т.н. Талалаев А.А.,  
<sup>1)</sup> к.т.н. Фраленко В.П., <sup>1,2,3,4)</sup> д.т.н. Хачумов В.М., <sup>1,2,3,4)</sup> к.ф.-м.н. Хачумов М.В.,  
<sup>1)</sup> Шишкин О.Г., <sup>1)</sup> к.т.н. Шустова М.В.  
<sup>1)</sup> ИПС им. А.К. Айламазяна РАН, <sup>2)</sup> ФИЦ ИУ РАН, <sup>3)</sup> РУДН, <sup>4)</sup> РТУ МИРЭА

Отпечатано в РПК «РекЛайм»  
152020, г. Переславль-Залесский, ул. Октябрьская, д. 20  
Тел. +7 (901) 051-86-02  
Тираж – 150 экз.