

Министерство образования Республики Беларусь
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

УДК 004.932

№ госрегистрации 20160652

Инв №

Проректор по научной работе
к.т.н., доцент

_____ Д.О. Глухов

«28» декабря 2018г.

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Разработка методов и алгоритмов обработки сложных динамических изображений
(заключительный)

ГПНИ «Информатика, космос и безопасность» 1.7.05

Начальник ОСНИ

_____ Т.В. Гончарова

«28» декабря 2018г.

Научный руководитель
заведующий кафедрой
вычислительных систем и сетей,
к.т.н., доцент

_____ Р.П. Богуш

«28» декабря 2018г.

Новополоцк 2018

СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель,
зав. каф. ВСиС, к.т.н., доцент

Богуш Р.П.
(введение, п.1.1, гл. 2,
п.3.3, гл.4, заключение)

Ответственный исполнитель
магистрант каф. ВСиС

Захарова И.Ю.
(гл.3)

Аспирант каф. ВСиС

Ярошевич П.В.
(п.1.1-1.5)

Нормоконтролер

Ищенко Л.В.

Реферат

Отчет 74 с., 35 рис., 7 табл., 72 источника

ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЬ, АНАЛИЗ ИЗОБРАЖЕНИЙ,
СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ, УПРАВЛЕНИЕ ПАРКОВКОЙ,
ОБНАРУЖЕНИЕ ЛЕСНЫХ ПОЖАРОВ, МИКРОСТРУКТУРА МЕТАЛЛОВ.

Объект исследования или разработки: изображения и последовательности изображений.

Цель исследований - синтез методов и алгоритмов обработки сложных динамических изображений с улучшенными качественными характеристиками для повышения эффективности интеллектуальных систем анализа видеоинформации.

При проведении исследований использовались следующие методы и методологии: обработки изображений; распознавания образов; теории вероятности и математической статистики; программирования.

В результате исследований разработаны новые алгоритмы обработки сложных динамических изображений: алгоритм классификации изображений парковочных мест на свободные и занятые с использованием гистограмм ориентированных градиентов и метода опорных векторов, позволяющий повысить вероятность правильной классификации; алгоритм обнаружения дыма лесного пожара на динамических изображениях на основе блочного пространственно-временного анализа и ковариационных признаков, который детектирует области с дымом малого размера на сложных кадрах видеопоследовательностей; алгоритм обнаружения объектов на видеопоследовательностях с разрешением 4К, который направлен на повышение точности обнаружения объектов небольших размеров на изображениях большого разрешения.

Результаты НИР используются в контрольно-испытательной лаборатории учреждения образования «Полоцкий государственный университет», а также при организации учебного процесса по курсам «Цифровая обработка сигналов и изображений», «Цифровая обработка изображений» и при проведении научно-исследовательской работы студентов первой ступени высшего образования специальностей 1-400201 «Вычислительные машины, системы и сети», 1-400101 «Программное обеспечение информационных технологий» и второй ступени высшего образования специальности 1-408004 «Математическое моделирование, численные методы и комплекс программ».

Полученные алгоритмы могут использоваться при разработке современных систем обнаружения произвольных объектов на видеопоследовательностях, в том числе получаемых с камер большого разрешения, для обнаружения и классификации объектов в видеосистемах управления автомобильными парковками, системах мониторинга лесных пожаров.

Содержание

Введение	6
1 Структура и алгоритмы обработки изображений интеллектуальных систем управления парковками	7
1.1 Структура интеллектуальной системы управления парковками	8
1.2 Сегментация парковочных мест на изображениях автомобильных парковок	10
1.3 Вычисление вектора признаков парковочного места	16
1.4 Классификация признаков	17
1.5 Программная реализация и результаты исследований	18
2 Обнаружение дыма лесного пожара на динамических изображениях	27
2.1 Общие принципы построения алгоритмов детектирования дыма лесного пожара на видео	28
2.2 Алгоритм раннего обнаружения лесных пожаров на основе пространственно-временного анализа и метода опорных векторов	28
2.3 Программная реализация и результаты исследования алгоритма	34
3. Обнаружение объектов на видеопоследовательностях с большим разрешением на основе сверточных нейронных сетей	37
3.1 Сверточные нейронные сети для обнаружения объектов на основе их статических признаков	37
3.1.1 Модель AlexNet	37
3.1.2 Модель R-CNN	37
3.1.3 Модель Fast R-CNN	37
3.1.4 Модель Faster R-CNN	38
3.1.5 Модель GoogLeNet и модификации	38
3.1.6 Модель ResNet и модификации	39
3.1.7 Модель YOLO и модификации	40
3.2 Сверточные нейронные сети для обнаружения объектов с учетом их динамических признаков	41
3.2.1 Модель Seq-NMS	41
3.2.2 Модель T-CNN и модификации	42
3.2.3 Модель DFFCNN	43
3.3 Блочный алгоритм обнаружения объектов на видеопоследовательностях с большим разрешением на основе СНС YOLO	44
3.3.1 Описание алгоритма	44
3.3.2 Результаты исследований	47
3.3.3 Модификация блочного алгоритма	49

4 Анализ изображений микроструктуры металлов	53
4.1 Этапы обработки	53
4.1.1 Предварительная обработка	53
4.1.2 Подсчет зерен	55
4.1.3 Определение длин хорд	56
4.1.4 Подсчет пересечения границ	57
4.1.5 Определение содержания углерода	57
4.2 Программная реализация алгоритмов обработки	57
Заключение	61
Список использованных источников	63
ПРИЛОЖЕНИЕ А Перечень публикаций по заданию «Информатика, космос и безопасность» 1.7.05 за 2016-2018гг.	68
ПРИЛОЖЕНИЕ Б Список конференций, на которых докладывались результаты исследований	71
ПРИЛОЖЕНИЕ В Акты внедрения результатов НИР	72

Введение

Обработка и анализ изображений на основе искусственного интеллекта с последующим принятием решения без участия человека все шире используется при решении сложных прикладных задач [1-3]. Известно, что последовательности видеоизображений характеризуются тем, что содержат значительный объем информации, но при этом на обработку и извлечение необходимой информации требуются огромные вычислительные затраты. Однако, значительное повышение производительности вычислительной техники за последнее десятилетие предопределило развитие и использование ресурсоемких методов обработки информации, возможности которых ранее были ограничены аппаратными средствами, что привело к расширению спектра решаемых практических задач. Обнаружение объектов на видеопоследовательностях и дальнейшее их распознавание являются актуальными задачами для автоматизированных систем управления и принятия решений, использующих видеонаблюдение и техническое зрение.

За последние несколько десятков лет число людей, которые живут в городах, постоянно растет, особенно это характерно для развитых регионов. Вместе с ростом городского населения, увеличивается количество используемых транспортных средств. Такие тенденции приводят к значительному усложнению парковки автомобилей. Следовательно, увеличивается количество времени на поиск свободного парковочного места, при этом повышаются стрессовые состояния водителей и загрязнение воздуха. Поэтому интеллектуальные системы управления парковками, направленные на оптимизацию пути к свободному парковочному месту являются актуальными и позволяют минимизировать указанные выше проблемы. Такие системы целесообразно интегрировать в комплексные системы управления городским имуществом «Smart City» («умный город»). Поэтому необходимо разработка структуры видеосистемы управления парковкой с учетом возможности ее интеграции в «Smart City» и алгоритмов обработки видеоинформации.

Актуальной прикладной задачей, решаемых в последнее время с помощью систем машинного зрения, является обнаружение лесных пожаров. Установлено что, каждый год, приблизительно 0,1% мировой лесной поверхности разрушается пожарами [4]. Чаще всего для обнаружения лесных пожаров используются смотровые башни, установленные для наблюдения за лесом. При этом, камеры видеонаблюдения размещаются на смотровых башнях, для контроля окружающего леса. Несмотря на то, что многие научно-исследовательские работы направлены на разработку систем автоматического обнаружения пожара на основе методов машинного зрения, данная проблема не решена в полной мере.

Исходными данными для выполнения НИР были цветные изображения, и последовательности цветных изображений.

Цель исследований - синтез методов и алгоритмов обработки сложных динамических изображений с улучшенными качественными характеристиками для повышения эффективности интеллектуальных систем анализа видеoinформации.

В результате исследований разработаны новые алгоритмы обработки сложных динамических изображений: алгоритм классификации изображений парковочных мест на свободные и занятые с использованием гистограмм ориентированных градиентов и метода опорных векторов; алгоритм обнаружения дыма лесного пожара на динамических изображениях на основе блочного пространственно-временного анализа и ковариационных признаков; алгоритм обнаружения объектов на видеопоследовательностях с разрешением 4К, который направлен на повышения точности обнаружения объектов небольших размеров на изображениях большого разрешения.

Метрологическое обеспечение НИР: компьютеры с характеристиками CPUi7 4.3 ГГц, ОЗУ 32 Гб, GPU Nvidia GeForce GTX 1070 и CPU Intel(R) Core(TM)2 Duo 2.6ГГц, ОЗУ4.0ГБ; ноутбук Dell Inspiron N5110N CPU Intel Core i5-2410M ОЗУ - 3.0ГБ; языки программирования Java и C++; библиотека машинного обучения метода опорных векторов jlibsvm; библиотека компьютерного зрения OpenCV 3.1; библиотеки изображений PASCAL VOC и COCO.

По результатам выполнения НИР подготовлены промежуточные отчеты за 2016г. и за 2017г.:

Разработка методов и алгоритмов обработки сложных динамических изображений: отчет о НИР (промежуточ.)/УО «ПГУ»; рук. Богуш Р.П.; исполн.: Ярошевич П.В. – Новополоцк, 2016. – 38с. – Библиогр.: с.35-37. - №ГР 20160652 от 04.04.2016. – Инв. № б/н.

Разработка методов и алгоритмов обработки сложных динамических изображений: отчет о НИР (промежуточ.)/УО «ПГУ»; рук. Богуш Р.П.; исполн.: Захарова И.Ю. – Новополоцк, 2017. – 31с. – Библиогр.: с.29-30. - №ГР 20160652 от 04.04.2016. – Инв. № б/н.

1 Структура и алгоритмы обработки изображений интеллектуальных систем управления парковками

1.1 Структура интеллектуальной системы управления парковками

Существует ряд моделей интеллектуальных видеосистем управления парковкой [5-8]. В [6] рассматривается система управления и мониторинга автостоянки с помощью видеокамеры для простого внутреннего гаража. При этом, на стоянке имеется постоянный источник освещения. С учетом этой особенности свободные места для парковки обнаруживаются с использованием метода выделения краев, при этом учитывается только внутренний край. В [7] система парковки использует коричневое округленное информационное изображение, которое изображено на каждом парковочном месте, и на основе анализа видимости этого пятна выдает информацию о свободных местах. На дисплее отображается количество имеющихся доступных парковочных мест. Данная система была протестирована и результаты представлены только для 8 парковочных мест. В [8] представлена видеосистема для обнаружения вакантных парковочных мест на основе анализа цветовых гистограмм и разности гауссовских признаков, классификатора на основе метода опорных векторов. Данная система может использоваться для навигации водителей к свободному парковочному месту автономной стоянки транспортных средств.

Структура интеллектуальной видеосистемы управления парковкой, представлена на рисунке 1 и включает основные блоки: видеокамера, модуль обработки видеоинформации, облачная платформа, локальный модуль хранения информации и устройство отображения видеоинформации (экран).

Модуль обработки видеоинформации такой системы включает две подсистемы: подсистема локализации парковочных мест и подсистема классификации парковочных мест.

Подсистема локализации парковочных мест предназначена для определения расположения парковочного места на видеокадре (сегментации) и записи его координат. При штатном режиме работы видеосистемы управления парковкой функционирование данной подсистемы не требуется.

Преобразование перспективы изображения полученного выполняется для обеспечения возможности описания парковочного места прямоугольником и облегчения обнаружения разделяющих полос парковки. Решение данной задачи необходимо при установке нового программно-аппаратного комплекса управления парковкой, переносе точки обзора (видеокамеры), модернизации оборудования системы, замене программного обеспечения и т. д. На последнем шаге работы подсистемы выполняется определение и запись координат парковочных мест для видеокадра в устройство хранения информации, которые будут использоваться подсистемой классификации парковочных мест.

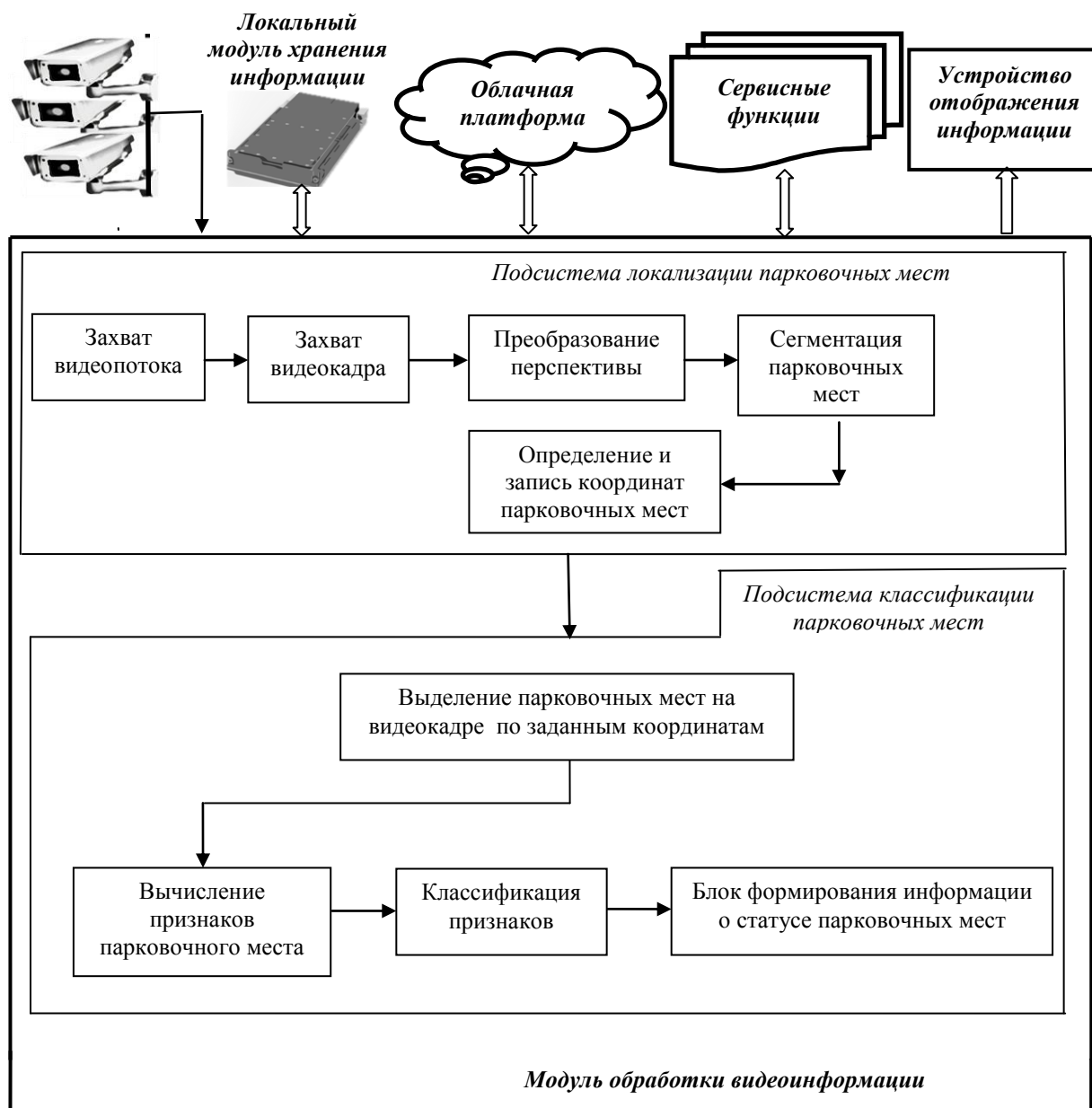


Рисунок 1 – Структура интеллектуальной видеосистемы управления парковкой

К задачам подсистемы классификации парковочных мест относятся: выделение парковочных мест на видеокadre по заданным координатам – на основе известных координат для дальнейшей обработки выделяются области на изображении (регионы интереса), которые соответствуют парковочным местам; вычисление признаков парковочного места – для каждого региона интереса формируется вектор признаков с целью описания его состояния; классификация признаков – выполняется разделение парковочных мест на свободные и занятые; блок формирования информации о статусе парковочных мест совмещает данные классификации с входным видеорядом для визуального отображения занятости парковочных мест. В настоящее время активно ведутся исследования в данной области и предлагается ряд методов для определения занятости парковочных мест по видеоизображениям: методы, основанные на обнаружении автомобиля на

парковочном месте [9,10]; методы, основанные на сравнении обрабатываемого парковочного места с эталонным свободным местом [11,12] и комбинированные методы [13,14], построенные на основе первых двух подходов. Методы первой группы неустойчивы к ошибкам классификации, возникающим из-за взаимного перекрытия автомобилей на видеоизображениях. Методы второй группы предполагают использование модели эталонного свободного парковочного места, поэтому вероятность ложной классификации увеличивается при появлении естественных шумовых факторов, людей, при локальном изменении освещенности и т.д. Комбинированные методы, как правило, предполагают использование преимуществ обоих подходов, но требуют, при этом, значительных вычислительных затрат.

Облачная платформа предусмотрена для масштабирования системы и (или) обеспечения возможности интеграции данной системы в комплексную систему Smart City.

Функциональность данной системы может быть расширена за счет дополнительного модуля «Сервисные функции», который может включать подсчет количества автомобилей на стоянке, определение типа транспортного средства, предоставление справочной информации в виде маршрута к парковочному месту, удаленное наблюдение владельца за автомобилем с помощью смартфона и др. Такой подход поможет в решении еще одной важной задачи, такой как, сокращение негативного влияния на окружающую среду, в том числе, и за счет уменьшения количества вредных выбросов при поиске свободного места на большой парковке в мегаполисах.

1.2 Сегментация парковочных мест на изображениях автомобильных парковок

Точное определение расположения парковочного места на изображении необходимо при выполнении ряда задач, например, при установке нового программно-аппаратного комплекса управления парковкой, при переносе точки обзора, при модернизации оборудования или программного обеспечения и т.д.

В работе [14] для анализа занятости парковки используются изображения, полученные на основе аэросъемки. При этом, предлагаемый алгоритм учитывает особенность размещения автомобилей над поверхностью Земли и построен на основе анализа карт интенсивности и высот для сегментации автомобиля на парковочном месте. После сегментации парковочных мест и автомобилей на них, восстанавливается изображение парковки со свободными парковочными местами. На основе изображений автомобилей генерируются 3D структуры, которые затем используются для моделирования активности на парковке. Для эффективной работы метода требуется четкая видимость разделяющих полос. Таким образом, при достаточно большой занятости парковки алгоритм некорректно сегментирует регионы интереса и автомобили. В статье [15] рассматривается структурирование

парковочных мест на автостоянке на основе комбинированного метода, включающего два шага. При этом первый шаг выполняет оператор, который выделяет границы каждого места парковки белым цветом и присваивает центральному пикселю желтый цвет. На втором шаге на основе программной реализации алгоритма выполняется автоматический поиск преобразованных пикселей желтого цвета и определение координат границ обнаруженного региона интереса. В [16] авторы предлагают алгоритм определения расположения парковочных мест, состоящий из следующих шагов: определение расположения линий, применение процедур фильтрации, предварительное объединение линий в парковочные места и ряды, операции интерполяции, экстраполяции и предсказания для обнаружения пропущенных мест парковки. Метод достаточно эффективен, однако необходимо использовать изображения, полученные на основе аэросъемки с четкой видимостью разделяющих полос.

Разработан эффективный алгоритм для автоматической сегментации парковочных мест на изображениях автомобильных парковок, получаемых с использованием камер видеонаблюдения, который включает преобразование перспективы исходного изображения, адаптивную бинаризацию методом Отсу, применение операций математической морфологии, горизонтальное дифференцирование изображения, построение вертикальных линий, фильтрацию и определение координат парковочных мест [17]. Алгоритм состоит из последовательности следующих шагов:

Преобразование перспективы изображения парковки. На исходном изображении определяются четыре исходные точки (x_i, y_i) $i = \overline{1,4}$, координаты которых на преобразованном изображении являются вершинами прямоугольной области (x'_i, y'_i) $i = \overline{1,4}$. Данная область характеризуется минимальной площадью, которая охватывает исходные точки. Преобразование перспективы выполняется как:

$$(x', y', 1) = H \times (x, y, 1)^T$$

где: H - матрица гомографии.

Матрица гомографии формируется на основе вектора $G_H = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})^T$ в результате решения системы линейных алгебраических уравнений:

$$V \cdot G_H = Y$$

где $V = \begin{bmatrix} (x_i, y_i, 1) & 0 & -\overline{(x_i, y_i)^T \cdot (x'_i, y'_i)} \\ 0 & (x'_i, y'_i, 1) & 0 \end{bmatrix}$ - матрица, содержащая

координаты точек исходного и преобразованного изображения; $Y = (x'_1, x'_2, x'_3, x'_4, y'_1, y'_2, y'_3, y'_4)^T$ - вектор столбец преобразованных координат.

Бинаризация изображения методом Отсу. Данный алгоритм, используя анализ гистограммы изображения, делит его на два класса [18].

Фильтрация на основе операций математической морфологии. Операция замыкания - для удаления разрывов на разделяющих парковочные места полосах, размывания - фильтрация мелких и несвязанных элементов.

Замыкание выполняется на основе правила:

$$A \bullet B = (A + B) - B$$

где $A + B$ - операция дилатации; A - обрабатываемое изображение; B - структурирующий элемент типа «квадрат» размером 7×7 ; $A - B$ - операция эрозии.

Размыкание выполняется как:

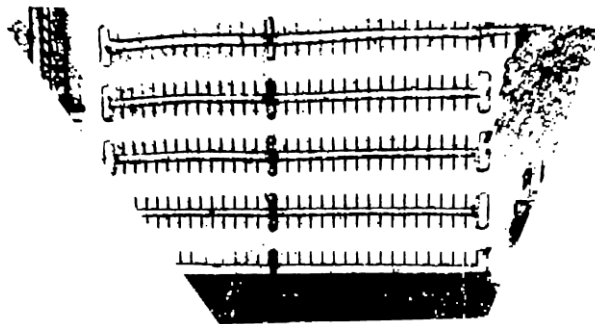
$$A \circ B = (A - B) + B$$

где B - структурирующий элемент типа «квадрат» размером 5×5 .

Результат применения первых трех шагов показан на рисунке 2.



а)



б)

Рисунок 2 - Пример применения первых трех шагов алгоритма: а) исходное изображение; б) результат обработки после третьего шага

Формирование горизонтальных линий. Изображению $I_{x,y}$, $x = \overline{0, w-1}$, $y = \overline{0, h-1}$, w - ширина изображения, h - высота изображения, ставится в соответствие конечное множество векторов $\{V_i(x, y, l)\}_{i=0, nov-1}$, где каждый вектор $V_i(x, y, l)$, из nov общего числа векторов, задает неразрывную горизонтальную последовательность пикселей, с началом в координате (x, y) и длиной l , а именно $(x, y) = (x, y + 1) = \dots = (x, y + l - 1) = Color.Black$. Множество векторов горизонтальных линий $\{V_i\}_{i=0, nov-1}$, определяется как:

```

nov = 0, l = 0
for y = 0 : h - 1
    for x = 0 : w - 1

```

```

if  $I_{x,y} \equiv Color.Black$ 
 $V(x, y, l) = V(0, 0, 0)$ 
while  $I_{x,y} \equiv Color.Black$ 
 $V : x = V : x + 1$ 
 $V : l = V : l + 1$ 
 $V_{nov}(x, y, l) = V(x, y, l)$ 
 $nov = nov + 1, l = 0$ 

```

Результат выполнения этого шага к выделенному фрагменту изображения (рисунок 3а) показан на рисунке 3б.

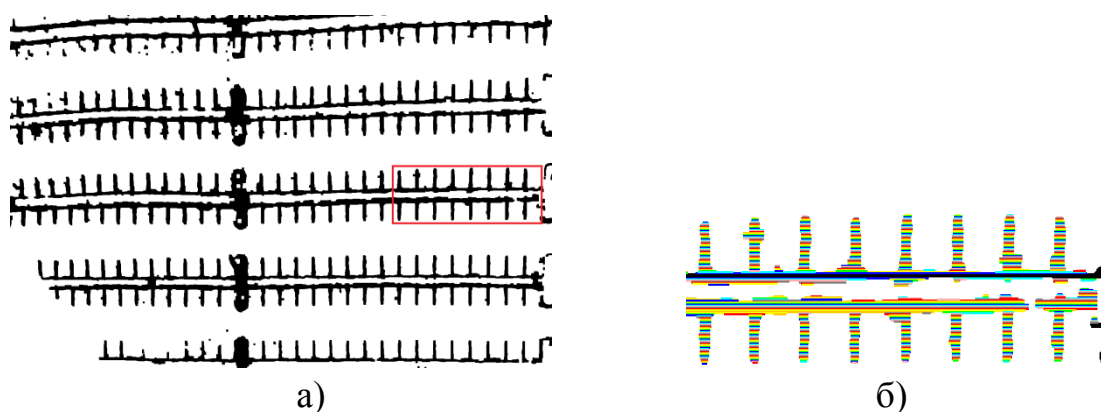


Рисунок 3 – Пример горизонтального дифференцирования: а) исходное изображение с выделенным фрагментом; б) результат обработки фрагмента

Построение вертикальных линий, пример построения вертикальных разделяющих полос по множеству горизонтальных линий $\{L_j(\bar{V})\}_{j=0, nol-1}$ на основе элементов множества векторов $\{V_i\}$, где nol - количество детектированных линий, следующим образом:

$j = 0, nov_j = 0$ (nov_j - количество векторов в линии)

$T = V_j$

while $nov \neq 0$

$nov_j = 0$

$L : V_{nov_j} = T$

$nov_j = nov_j + 1$

for $i = 0 : nov - 1$

$T = V_i$

$\min = \min(L : V_0 : l, L : V_{nov_j-1} : l, T : l)$

if $L : V_{nov_j-1} : y - T : y \equiv 1$ and $\min > |L : V_0 : x - T : x|$

and $\min > |L : V_0 : x + L : V_0 : l - T : x + T : l|$

$L : V_{nov_j} = T$

```

        nov_j = nov_j + 1
        remove T from {V_i}
        i = i - 1
    if nov_j ≡ 1
        remove T from {V_i}
    else
        L_j = L
        j = j + 1

```

Результат работы данного шага алгоритма для изображения, приведенного на рисунке 3 б, показан на рисунке 4. В вертикальной линии L_j , как указано в условии, образующие её вектора должны находиться один под другими и не превышать соседние по длине.

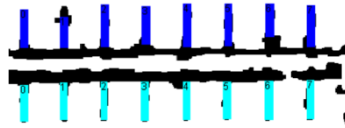


Рисунок 4 - Результат построения вертикальных линий

Описание линий начальными координатами (x, y) , шириной w и высотой h : $S_j(x, y, w, h)$. Для этого в область, образуемую последовательностью векторов, вписывается прямоугольник, с максимально возможной шириной и высотой. Результатом данного шага является детектирование вертикальных полос, разделяющих парковочные места.

Определение координат парковочных мест. Линии сортируются по начальным координатам: слева на право, сверху вниз. Затем они объединяются в парковочные места, образуя множество парковочных мест $\{S_k(L_l, L_r)\}_{k=0, nos-1}$, где L_l, L_r - левая и правая разделяющие полосы соответственно, nos - количество парковочных мест. Формирование множества парковочных мест $\{S_k\}$ происходит следующим образом:

```

i = 0, nos = 0
S = S(L_0, L_1)
i = i + 1
for i = i : nol - 1
    if max(S : L_i : w, S : L_r : w, L_i : w) > |(S : L_r : x - S : L_i : x) - (L_i : x - S : L_r : x)|
        S_nos = S
        nos = nos + 1
        S = S(S_nos-1 : L_r, L_i)
    else if i < nol - 1
        S = S(L_i, L_{i+1})
        i = i + 1

```

В результате чего может быть определено расположение мест парковки, (рисунок 5).



Рисунок 5 - Пример определения координат парковочных мест

Описание парковочного места S_k . Каждое сегментированное парковочное место описывается начальными координатами (x, y) , шириной w , высотой h :

$\{S_k(x, y, w, h)\}_{k=0, nos-1}$, при этом:

$$x = L_l : x + L_l : w;$$

$$y = \min(L_l : y, L_r : y);$$

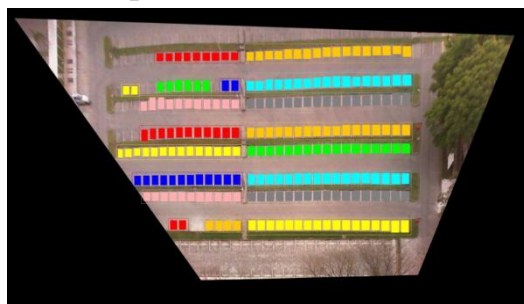
$$w = L_r : x - x;$$

$$h = \max(L_l : y + L_l : h, L_r : y + L_r : h) - y.$$

На рисунке 6 приведены примеры работы алгоритма.



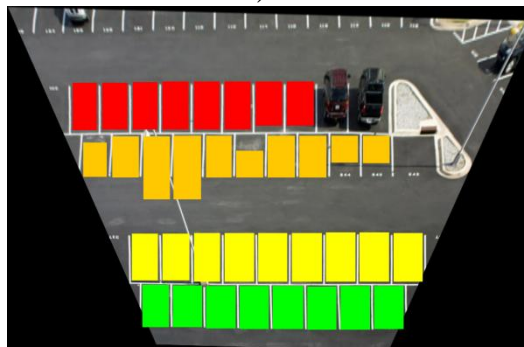
a)



b)



c)



d)

Рисунок 6 - Пример сегментации мест парковки: а) и с) исходные изображения парковок, b) и d) результат сегментации парковочных мест

Анализ результатов исследований свидетельствует о том, что предложенный алгоритм позволяет достаточно эффективно сегментировать парковочные места на изображениях автопарковок. Ошибки в работе алгоритма возможны в случае существенного отличия размеров парковочных мест или их перекрытия сторонними объектами.

1.3 Вычисление вектора признаков парковочного места

Для построения вектора признаков изображения парковочного места перспективным является использование гистограмм ориентированных градиентов, т.к. в данном случае обеспечивается инвариантность относительно поворота изображения и изменения масштаба, устойчивость к шуму и изменению освещения [19]. Предлагаемый алгоритм построения вектора признаков парковочного места на основе гистограммы ориентированных градиентов требует выполнения следующих шагов:

- 1) Вычисление градиентов для исходного изображения I с размерами $w \times h$:

$$G_{x,y}^x = I_{x,y} \cdot M^x \quad G_{x,y}^y = I_{x,y} \cdot M^y$$

где $0 \leq x < w$, $0 \leq y < h$, $M(M^x, M^y)$ дискретный дифференцирующий оператор (M^x и M^y для осей x и y соответственно).

- 2) Вычисление модуля вектора градиента $G_{x,y}$ и ориентации $\theta_{x,y}$ для изображения I :

$$G_{x,y} = \sqrt{G_{x,y}^x{}^2 + G_{x,y}^y{}^2} \quad \theta_{x,y} = \arctan\left(\frac{G_{x,y}^y}{G_{x,y}^x}\right)$$

и формирование матрицы градиентов G и ориентаций θ изображения I .

- 3) Разбиение G на ячейки $C_{m,n}$ размером $C_w \times C_h$ и накопление мощностей градиентов в соответствии с ориентациями θ , для каждой из них:

$$C_{m,n}^l = \sum_{\substack{i=m \cdot cw \\ j=n \cdot ch}}^{(m+1) \cdot cw - 1 \\ (n+1) \cdot ch - 1} \begin{cases} G_{i,j}, l \cdot \frac{2\pi}{b} \leq \theta_{i,j} < (l+1) \cdot \frac{2\pi}{b} \\ 0, \cdot \frac{2\pi}{b} > \theta_{i,j} \quad \theta_{i,j} \geq (l+1) \cdot \frac{2\pi}{b} \end{cases}$$

где b - число ориентаций в ячейке, которое определяется поставленной задачей, $0 \leq l < b$, $0 \leq m < cw$, $0 \leq n < ch$, $cw = \frac{w}{C_w}$, $ch = \frac{h}{C_h}$.

- 4) Объединение ячеек в блоки $B_{f,g}$ размером $B_w \times B_h$:

$$B_{f,g} = \begin{pmatrix} C_{f,g} & \cdots & C_{f+B_w-1,g} \\ \vdots & \ddots & \vdots \\ C_{f,g+B_h-1} & \cdots & C_{f+B_w-1,g+B_h-1} \end{pmatrix}$$

где $0 \leq m < cw$, $0 \leq g < bh$, $bw = cw - B_w + 1$, $bh = ch - B_h + 1$.

- 5) Формирование вектора признаков:

5.1) Вычисление второй нормы ориентаций ячеек блока $B_{f,g}$:

$$\|B_{f,g}\|_2 = \sqrt{\sum_{i=f}^{f+B_w-1} \sum_{j=g}^{g+B_h-1} \sum_{l=0}^{b-1} C_{i,j}^l}$$

5.2) Нормализация ориентаций ячеек блока:

$$C_{i,j}^{l,f,g} = \frac{C_{i,j}^l}{\|B_{f,g}\|_2}$$

5.3) Составление вектора признаков:

$$d_{b \cdot B_w \cdot B_h \cdot (bw \cdot g + f) + b \cdot (B_w \cdot (j-g) + (i-f)) + l} = C_{i,j}^{l,f,g}$$

Размер вектора d зависит от размера изображения $w \times h$, размера ячеек $C_w \times C_h$ и размера блока $B_w \times B_h$, а также от количества ориентаций в ячейке b . Таким образом, количество элементов в признаковом векторе может быть определено как:

$$D_s = b \cdot B_w \cdot B_h \cdot \left(\frac{w}{C_w} - 1\right) \cdot \left(\frac{h}{C_h} - 1\right)$$

1.4 Классификация признаков

Для классификации дескрипторов полученных на основе гистограмм ориентированных градиентов, парковочного места, применяется метод опорных векторов, который позволяет разделять парковочные мест на два класса - свободные и занятые. Правило классификации парковочных мест может быть записано в следующем виде [20]:

$$a(d) = \text{sign} \left(\sum_{i=0}^n \lambda_i y_i K(d_i, d) - w_o \right)$$

где $a(d)$ - решающее правило (принимающее одно из значений меток класса: +1 и -1 для свободных и занятых мест соответственно), $\lambda(\lambda_0, \lambda_1, \dots, \lambda_n)$ - вектор двойственных переменных, $y(y_0, y_1, \dots, y_n)$ - вектор меток классов опорных векторов, $\{d_i\}_{i=1, \dots, n}$ - множество опорных векторов признаков, w_0 - пороговое значение, d - обрабатываемый вектор признаков.

Эффективность работы классификатора в значительной степени определяется применяемой функцией ядра. Для классификации парковочных мест могут быть использованы следующие функции:

- линейная:

$$K_{lin}(d_a, d_b) = d_a \cdot d_b$$

- радиальная:

$$K_{rbf}(d_a, d_b) = \exp(-\gamma \cdot \|d_a - d_b\|^2)$$

- полиномиальная:

$$K_{poly}(d_a, d_b) = (\gamma \cdot K(d_a, d_b) + c)^\delta$$

- пересечения гистограмм:

$$K_{hist}(d_a, d_b) = \sum_i \min(|d_{a_i}|, |d_{b_i}|)$$

где d_a и d_b - признаковые вектора; γ , δ , c - параметры соответствующих ядер.

1.5 Программная реализация и результаты исследований

Для проведения экспериментов и оценки результатов исследований рассмотренный подход реализован на языке Java с использованием библиотеки компьютерного зрения OpenCV 3.0.0. При этом реализованы следующие классы.

Element предназначен для хранения информации о горизонтальных векторах $v_i(x, y, l)$. В качестве свойств класса используются переменные x , y , и l , которые описывают последовательность пикселей длиной l с началом в координате (x, y) . Конструктор *Element* производит инициализацию полей класса;

ElementsLine: применяется для описания вертикальной полосы $L_j(\bar{V})$ набором векторов $\{v_i\}$. Класс наследует `ArrayList<Element>` и представляет собой список элементов *Element*. *ElementsLine* содержит координаты (sx, sy) начала и координаты (ex, ey) окончания линии и параметры размеров (w, h) линии; метод *calculate* осуществляет определение размеров линии по имеющимся элементам текущего списка и переводит класс в вид $L_j(x, y, w, h)$, где (x, y) координатами верхнего левого угла;

Lines представляет собой список элементов *ElementsLine*, наследуя `ArrayList<ElementsLine>` и содержит метод *defineElementsLine*, который определяет параметры линий по координатам ее начала и размерам для всех содержащихся в списке элементов;

ParkingSpace предназначен для хранения информации о парковочном месте, которое описывается левой и правой разделяющими линиями $S_k(L_l, L_r)$ и параметрами начальной точки и размером парковочного места $S_k(x, y, w, h)$. Полями класса являются левая и правая разделяющие парковочные линии $(left, right)$, координаты места

парковки (sx, sy, ex, ey) , размеры парковочного места (w, h) . Класс содержит конструктор *ParkingSpace*, осуществляющий инициализацию его свойств;

ParkingSpaceGroup представляет собой список элементов *ParkingSpace*, наследуя *ArrayList<ParkingSpace>*, и хранит информацию о местах парковки в парковочном ряду. Полями класса являются координаты парковочного ряда и размеры парковочного ряда. Конструктор *ParkingSpaceGroup* осуществляет инициализацию свойств класса. Метод *add* является переопределенной функцией интерфейса *List*, предназначен для определения изменений свойств класса при добавлении нового региона интереса;

PakingLot является вершиной иерархии описания парковки, наследует *ArrayList<ParkingSpaceGroup>*. Данный класс содержит методы *read* и *write*. Метод *read* считывает данные, описывающие регионы интереса и их статусы на изображении парковки и возвращает элемент класса *PakingLot*. Метод *write* предназначен для записи элемента *PakingLot* в файл;

PakingSpaceDetector используется для локализации регионов интереса парковочных мест, согласно алгоритму, описанному в разделе 1. Для выполнения шагов 1-3, 4-6 и 7 алгоритма реализованы функции *processImage*, *detectVecticalLines* и *detectParkingSpaces* соответственно. Метод *detect* последовательно вызывает указанные функции. Для объединения парковочных мест в ряды предназначен метод *definePakingLot*.

Для осуществления чтения и записи данных парковки в/из файлов формата *.xml реализованы классы *XMLParkingLotReader* и *XMLParkingLotWriterDescriptor* является интерфейсом. Классы которые реализуют этот интерфейс должны предоставить свою реализацию метода *compute*. Метод *compute* возвращает, вычисленный для изображения, дескриптор, который представляет собой массив вещественных чисел;

HOGProperties предназначен для хранения информации о параметрах дескриптора. Класс содержит следующие свойства: *imageSize* - размер изображения, *cellSize* - размер ячейки; *binSize* - количество ориентаций в ячейке; *blockSize* - размер блока; *overlappingStep* - шаг перекрывания для блоков;

HOG наследует класс *HOGProperties* и реализует интерфейс *Descriptor*. Класс содержит конструктор по умолчанию, конструктор копирования, конструктор создания дескриптора по передаваемым параметрам размеров. Методы *computePreparableData* и *computeGandB* предварительно вычисляет матрицы номеров ячеек и весов, используемых при билинейной интерполяции, и значения градиента и номеров контейнеров ячеек, между которыми разделяется мощность градиента. Метод *getBlockDescriptor* вычисляет гистограмму ориентаций ячеек внутри блока. Нормализации гистограммы блока осуществляется методом *normalization*;

Метод опорных векторов реализован с помощью класса *SVM*, с использование библиотеки *jlsvm*. Эта библиотека позволяет использовать при обучении и

классификации собственные функции ядра. К свойствам класса относятся: логическая переменная *isTrained* - флаг того был ли обучен данный элемент класса SVM, false по умолчанию; *problem* - элемент класса непостоянной проблемы бинарной классификации; *parameters* - элемент класса содержащего в себе параметры метода опорных векторов; *model* - элемент класса бинарной модели. Метод *addTrainVector* осуществляет добавление признакового вектора с меткой обозначающей принадлежность дескриптора к одному из двух классов к проблеме *problem*. Метод *train* производит обучение бинарной модели *model*, с параметрами классификатора *parameters*, на проблеме *problem*. После этого значение флага *isTrained* меняется на true, что не позволит больше добавлять признаков в проблему и производить обучение. Методами *save* и *load* осуществляется запись и чтение опорных векторов решения проблемы в файл и из файла соответственно. Метод *test* в качестве параметров принимает дескриптор, представляющий собой массив вещественных чисел, производит его классификацию и возвращает метку принадлежности дескриптора к одному из двух классов.

Для дальнейших экспериментов, направленных на определении оптимальных значений параметров гистограмм ориентированных градиентов, в качестве данных использовались изображения парковки Папского католического университета Параны (Pontifical Catholic University of Parana) из базы данных Pk-Lot [21]. Pk-Lot содержит изображения парковки, полученные при различных погодных условиях, теневых эффектах и освещенности. Парковочные места находятся на различном удалении от камеры, которая установлена достаточно высоко над парковкой. Это позволяет перейти к виду сверху на изображение парковки путем преобразования перспективы, при этом отсутствуют перекрытия изображений парковочных мест транспортными средствами, расположенными рядом.

Для тестирования алгоритма использованы 2153 изображения парковки с общим количеством изображений парковочных мест 292808, которые были получены при различных погодных условиях (таблица 1).

Таблица 1 - Основные характеристики используемых изображений парковки

Погодные условия	Период съемки (дней), количество изображений	Общее количество изображений мест парковки	
		свободных	занятых
Туман	6/664	61406	28898
Дождь	4/416	18670	37906
Солнечно	12/1073	78801	67127

На рисунке 7 представлен пример изображений свободных и занятых парковочных мест, полученных в различных условиях съемки, используемых при

обучении классификатора, отличных от изображений, на основе которых проводились исследования характеристик алгоритма.

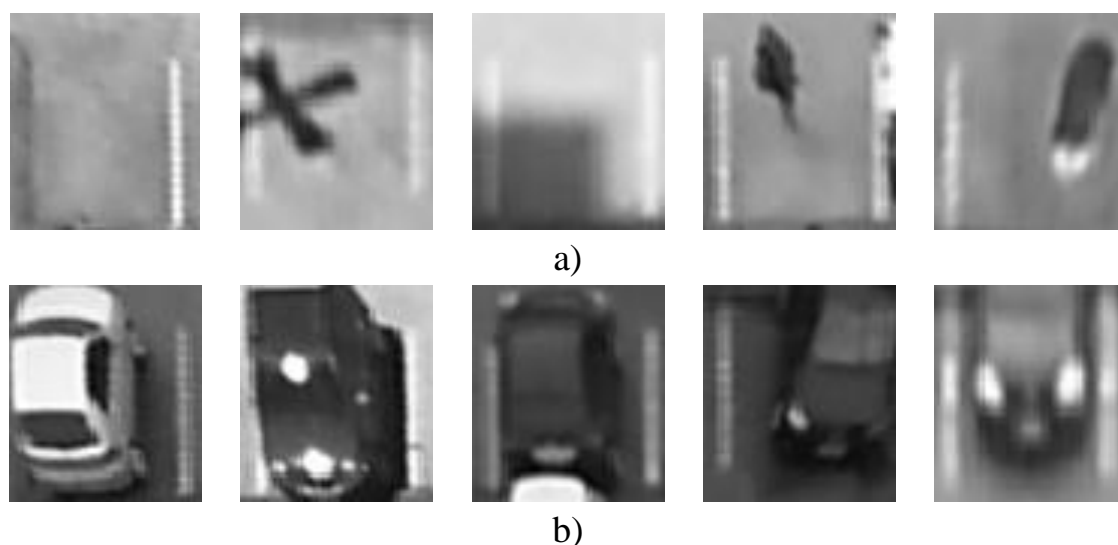


Рисунок 7 - Пример изображений, используемых при обучении классификатора: а) свободные парковочные места, б) занятые места парковки

На рисунке 8 представлена визуализация ориентаций ячеек свободного и занятого парковочных мест, вычисленных на основе гистограмм ориентированных градиентов.

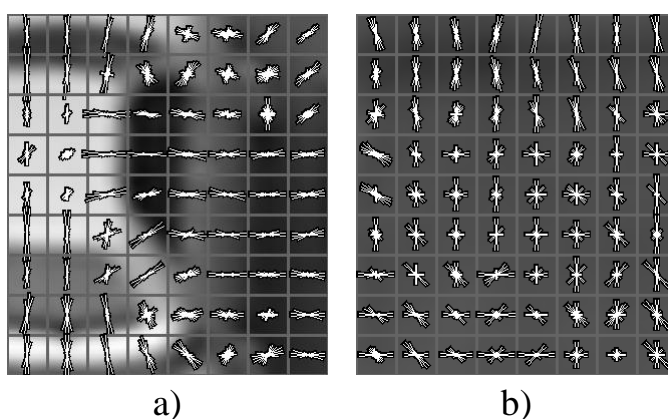


Рисунок 8 - Визуализация ориентаций ячеек для изображений парковочных мест: а) занятое парковочное место, б) свободное парковочное место

Полученный вектор признаков предлагаемого алгоритма имеет ряд параметров, которые определяют эффективность классификации: размеры парковочного места, размеры ячейки, гистограммы ориентаций ячейки, блока, шага перекрытия блоков и схемы нормализации. Поэтому, для предложенного алгоритма необходимо выполнить исследования с целью определения наиболее значимых параметров, обеспечивающих максимальную эффективность классификации.

Эффективность дескриптора оценивается вероятностью правильной классификации изображений парковочных мест $RR(w, h, C_w, C_h, b)$, которая вычисляется по формуле:

$$RR(w, h, C_w, C_h, b) = \frac{TP + TN}{TP + FP + TN + FN}$$

где w , h - ширина и высота изображения парковочного места; C_w , C_h - ширина и высота ячейки, b - количество ориентаций в ячейке; TP - количество правильно классифицированных свободных парковочных мест; FP - количество свободных мест парковки, классифицированных как занятые; TN - количество правильно классифицированных занятых парковочных мест, FN - количество занятых мест парковки, классифицированных как свободные.

Вероятность ложной классификации изображений свободных парковочных мест определяется как:

$$FPR = \frac{FP}{TP + FP}$$

Вероятность ложной классификации изображений занятых парковочных мест определяется как:

$$FNR = \frac{FN}{TN + FN}$$

На рисунке 9 представлены результаты эксперимента по определению наиболее эффективного размера изображения парковочного места дескриптора для наиболее используемых размеров ячейки при решении других практических задач 8×8 и количестве ориентаций в ячейке равном 9 для признакового вектора, формируемого на основе гистограмм ориентированных градиентов. Из рисунка 8 следует, что наиболее эффективными могут быть размеры изображений парковочных мест: 64×56 , 64×64 , 64×72 , 64×80 , 48×104 , для которых необходимо определение значений последующих параметров алгоритма.

На рисунках 10 и 11 показаны результаты экспериментов по определению наиболее эффективных размеров ячеек, обеспечивающих максимальную вероятность правильного обнаружения для изображений с размерами, полученными на предыдущем этапе исследований. Анализ рисунков 10 и 11 показывает, что максимальная вероятность правильной классификации при различных размерах ячеек обеспечивается для соответствующих размеров изображений парковочного места, при следующих параметрах дескриптора (w, h, C_w, C_h) парковочного места: $(64, 56, 8, 8)$, $(64, 64, 8, 8)$, $(64, 72, 8, 6)$, $(64, 72, 8, 8)$, $(64, 72, 8, 9)$, $(64, 72, 8, 12)$, $(64, 80, 8, 5)$, $(64, 80, 8, 8)$, $(64, 80, 8, 16)$, $(48, 104, 8, 8)$.

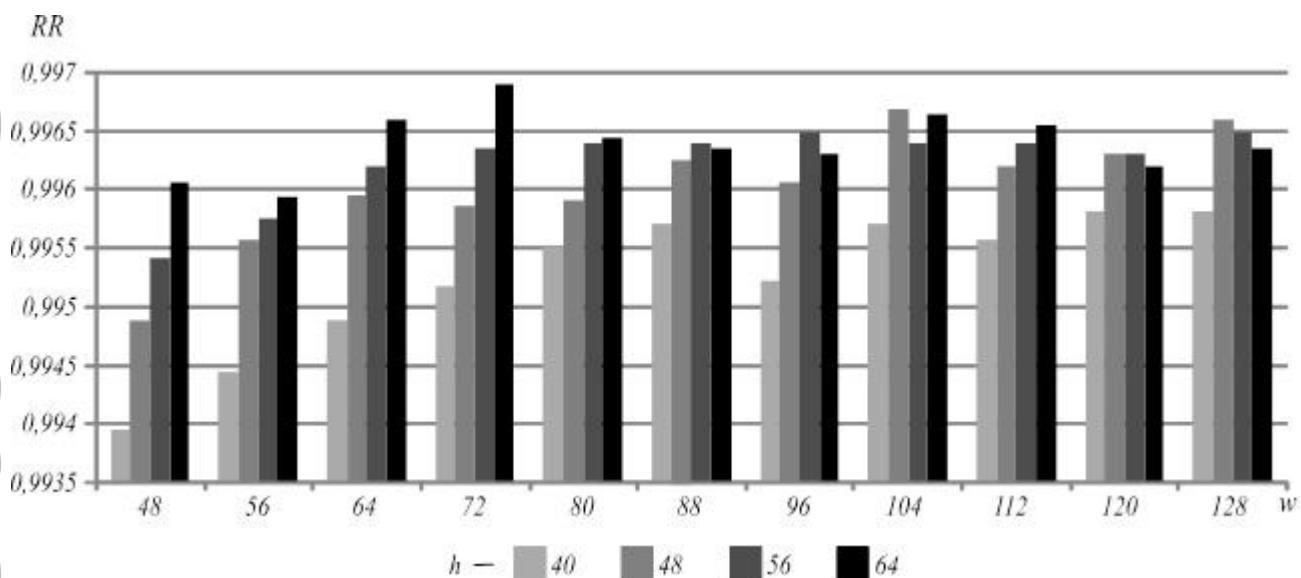


Рисунок 9 - Вероятность правильной классификации для различных размеров изображений парковочных мест

На следующем шаге проведены экспериментальные исследования по определению количества ориентаций в ячейке, обеспечивающих максимальную вероятность правильного обнаружения, в интервале от 4 до 18, т.е. в два раза меньше и больше от традиционного применяемого значения при решении других практических задач с использованием гистограмм ориентированных градиентов.

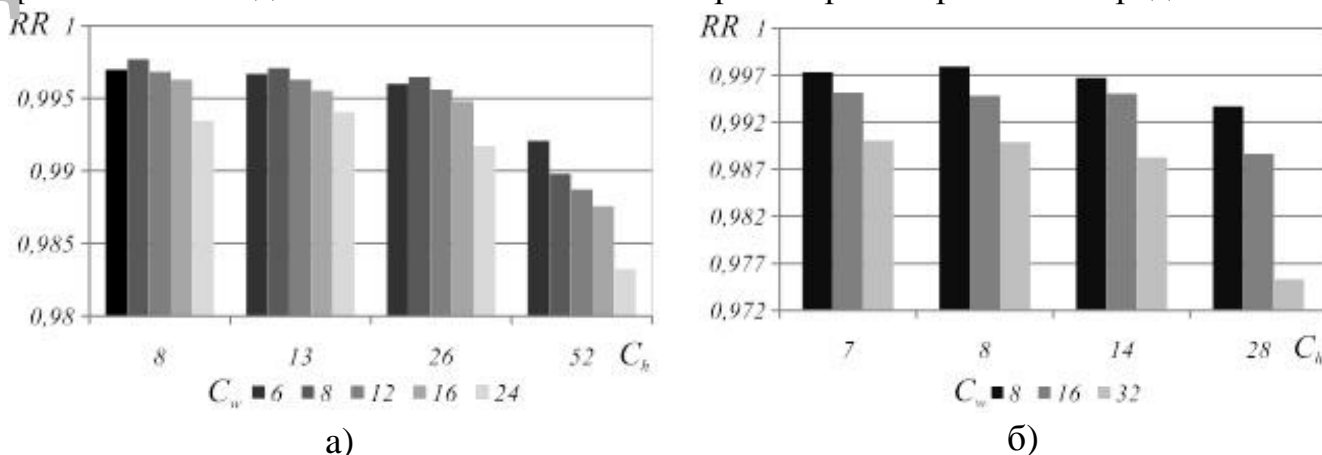


Рисунок 10 - Вероятность правильной классификации при различных размерах ячеек для изображения с $w \times h$: а) 48×104 , б) 64×56

На рисунке 12 представлены результаты исследований для количества ориентаций в ячейке в диапазоне от 9 до 18, т.к. для меньшего количества ориентаций вероятность правильного обнаружения значительно ниже. Результаты, представленные на рисунке 12, свидетельствуют о том, что вероятность правильной классификации тестовых изображений является максимальной для трех различных параметров дескриптора (w, h, C_w, C_h, b) : $(64, 72, 8, 6, 14)$, $(64, 72, 8, 6, 16)$, $(64, 72, 8, 8, 16)$. В этом случае необходимо найти признаковый вектор

МИНИМАЛЬНОЙ ДЛИНЫ.

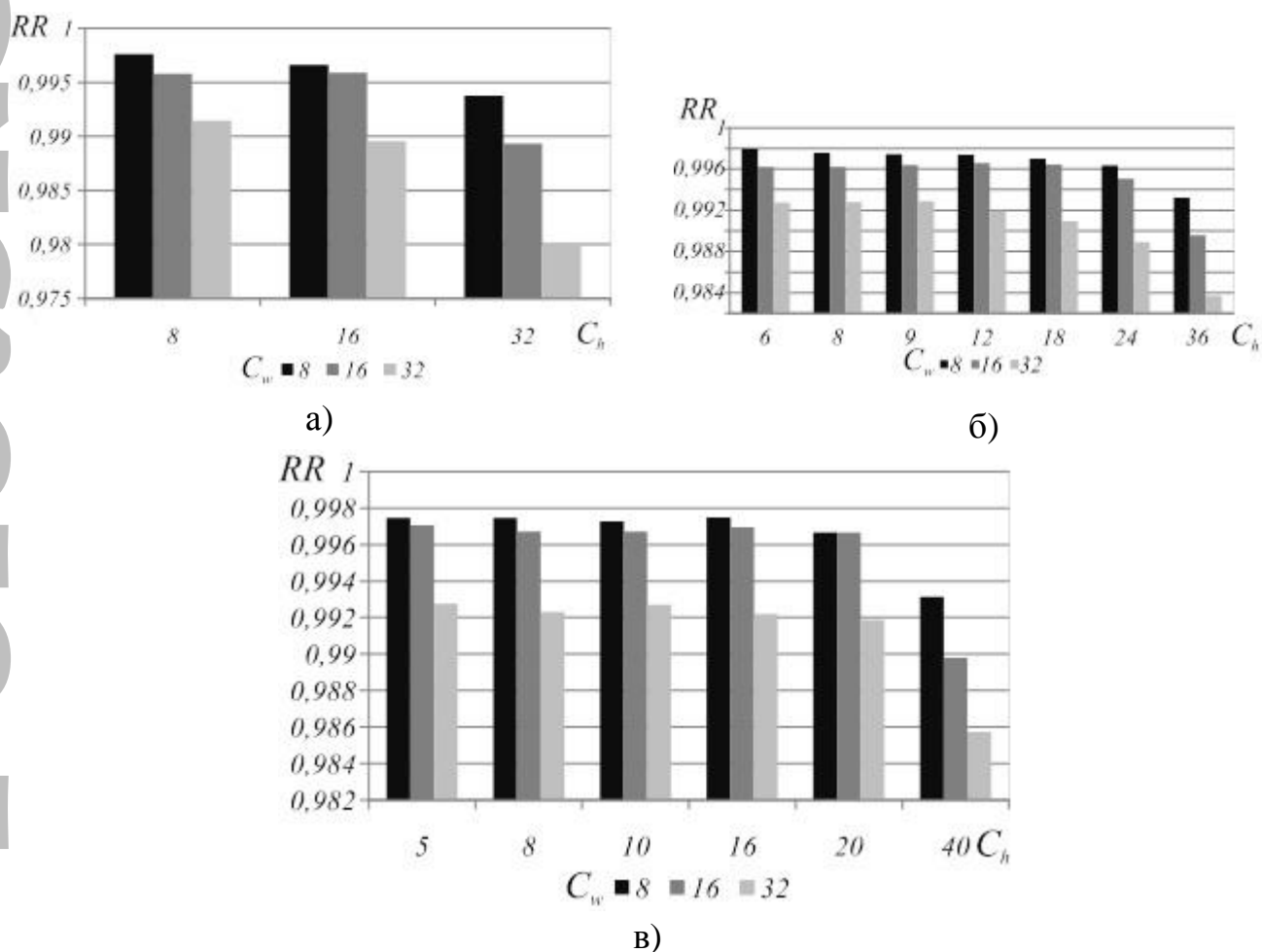


Рисунок 11 - Вероятность правильной классификации при различных размерах ячеек для изображения с $w \times h$: а) 64×64 , б) 64×80 , в) 64×72

Исходя из размеров полученных дескрипторов: $D_s(64,72,8,6,14) = 4928$, $D_s(64,72,8,6,16) = 5638$, $D_s(64,72,8,8,16) = 3584$, следует, что наиболее эффективными являются следующие параметры дескриптора: размер изображения парковочного места - 64×72 , размер ячейки - 8×8 , количество ориентаций в ячейке - 16. При использовании метода опорных векторов тип функции ядра и выбранные для него параметры в значительной мере влияют на результат классификации. Поэтому, проведены эксперименты по оценке эффективности классификации методом опорных векторов полученных признаков парковочных мест на основе гистограмм ориентированных градиентов при использовании различных функций ядра (таблица 2).

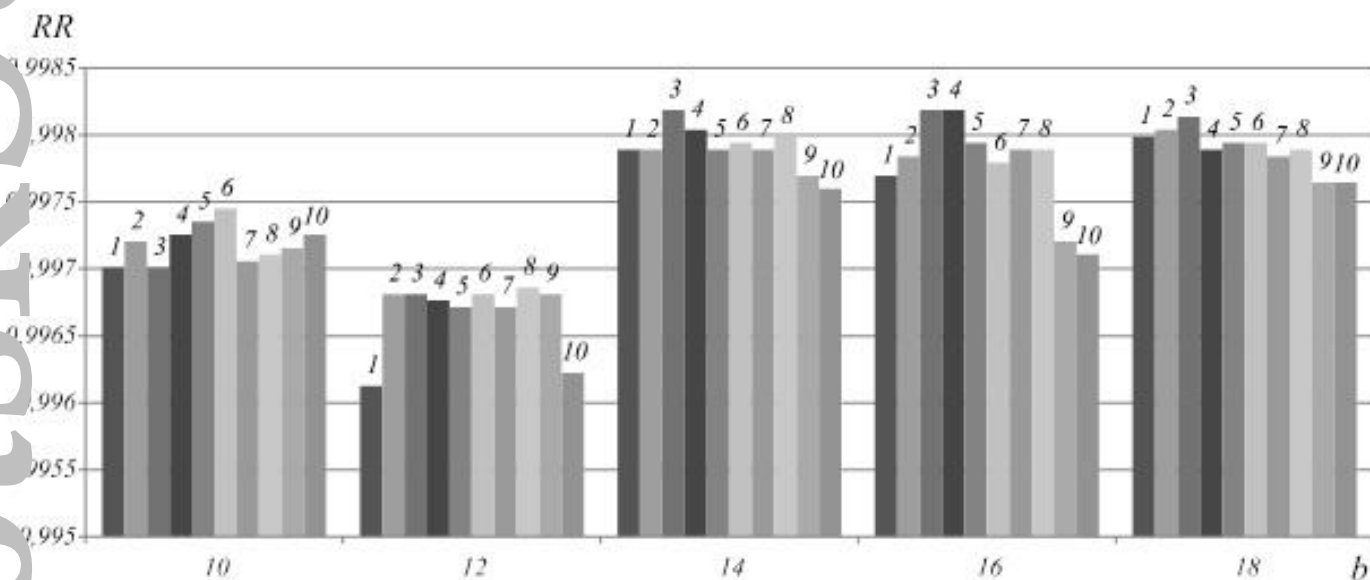
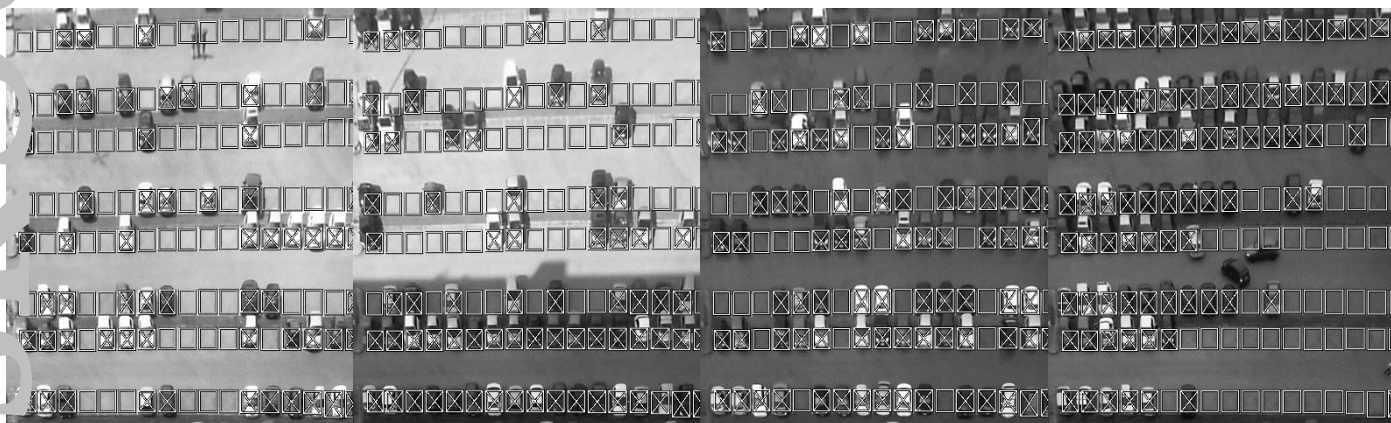


Рисунок 12 - Вероятность правильной классификации при различном количестве ориентаций в ячейках: 1 – (64, 56, 8, 8), 2 – (64, 64, 8, 8), 3 – (64, 72, 8, 6), 4 – (64, 72, 8, 8), 5 – (64, 72, 8, 9), 6 – (64, 72, 8, 12), 7 – (64, 80, 8, 5), 8 – (64, 80, 8, 8), 9 – (64, 80, 8, 16), 10 – (48, 104, 8, 8)

Таблица 2 - Сравнительная характеристика эффективности классификации при использовании различных функций ядра

Тип функции ядра	Ширина разделяющей полосы	Кол-во опорных векторов	RR	FPR	FNR
Линейная	4.366	161	0,99431	2,91e-3	9,16e-3
Пересечения гистограмм	2.387	262	0,99652	2,12e-3	5,19e-3
Радиальная, $\gamma = 0.1$	1.371	330	0,99706	1,41e-3	4,86e-3
Полиномиальная, $\delta = 7$, $\gamma = 0.1$, $c = 0$	0.752	479	0,99740	1,58e-3	3,86e-3

Результаты исследований, представленные в таблице 2 свидетельствует о том, что функция ядра на основе пересечения гистограмм является наиболее эффективной на основе комплексного анализа всех представленных качественных характеристик. На рисунке 13 показан пример классификации парковочных мест разработанным алгоритмом с определенными параметрами на свободные и занятые.



а) б) в) г)

Рисунок 13 - Результат классификации парковочных мест: а) в солнечный день, б) при наличии тени, в) после дождя, г) в пасмурную погоду

Сравнение характеристик представленного алгоритма с другими алгоритмами классификации парковочных мест, приведено в табл. 3. Анализ представленных данных показывает, что предложенный подход обеспечивает повышение вероятности правильной классификации изображений парковочных мест.

Таблица 3 - Сравнение характеристик алгоритмов классификации парковочных мест

Алгоритм	Дескриптор	Классификатор	Количество снимков парковки	Количество парковочных мест	Вероятность правильной классификации
C.-C. Huang, Y.-S. Dai, S.-J. Wang [22]	HOG $(w, h, C_w, C_h, b) = (64, 32, 16, 16, 8)$	Bayes	955	72	0,9945
C.-C. Huang, Y.-S. Tai, S.-J. Wang [13]	HOG $(w, h, C_w, C_h, b) = (64, 32, 16, 16, 10)$	Bayes	825	72	0,9939
C.-C. Huang, H.T. Vu, Y.-R. Chen [23]	HOG $(w, h, C_w, C_h, b) = (96, 48, -, -, -)$	SVM	1564	72	0,9955
R. Fusek at al. [24]	HOG $(w, h, C_w, C_h, b) = (96, 96, 8, 8, 4)$	SVM	–	57	0,691
R. Fusek at al. [24]	Haar	AdaBoots	–	57	0,95
M. Tschentscher at al. [25]	Histogram, HSV	k-NN	1010	36	0,9655
	DoG				0,9358
	Histogram, RGB	SVM			0,9712
	DoG				0,9413
L. Baroffio at al. [26]	Histogram, HSV	SVM	3791	28	0,96
			4152	37	0,93
			4474	100	0,87
Представленный алгоритм	HOG $(w, h, C_w, C_h, b) = (64, 72, 8, 8, 16)$	SVM	2153	136	0,997

2 Обнаружение дыма лесного пожара на динамических изображениях

Ключевой задачей современной системы обнаружения лесного пожара является раннее обнаружение. При этом детектирование признаков пожара в реальных условиях является нетривиальной задачей в виду разнообразия характеристик дыма в зависимости от широкого спектра факторов окружающей среды (погодные условия, освещение, наличие сходных объектов в кадре). Традиционно для обнаружения лесных пожаров используются наземные наблюдательные пункты, наземное и авиационное патрулирование [27]. В зоне наземной охраны лесов наряду с наземным патрулированием для обнаружения лесных пожаров используются передвижные телескопические мачты на базе автомобилей повышенной проходимости. Они оснащаются телеустановками и позволяют вести наблюдение за лесом в радиусе до 20 км. Такие наблюдательные пункты располагаются на возвышенностях в засушливые периоды пожароопасного сезона и оснащаются средствами связи и спутниковой навигации. У авиационного способа обнаружения лесных пожаров возможности гораздо выше. Особо следует отметить дистанционно пилотируемые летательные аппараты, оснащённые телекамерой, ИК-аппаратурой и системой спутниковой навигации GPS. Однако, дистанционно пилотируемые летательные аппараты имеют ряд существенных недостатков: слишком велика зависимость его полёта от ветра, сравнительно высокая аварийность и низкий ресурс, так же необходимость согласования полётов с органами управления воздушным движением. Но в силу ряда экономических и организационных причин в настоящее время данные способы мониторинга леса требуют слишком много затрат.

Для раннего обнаружения лесного пожара перспективным является использование алгоритмического обеспечения существующих систем видеомониторинга [27]. Данный подход имеет ряд преимуществ по сравнению с рассмотренными выше методами: оборудование систем видеонаблюдения возможно установить на уже существующие наблюдательные пункты; невысокая стоимость оборудования; при обнаружении практически не требуется вмешательство человека; возможность обнаружения возгорания в начальной стадии с точным определением местоположения очага пожара.

Высокая достоверность обнаружения пожара и низкая вероятность ложной тревоги, а также точность определения координат возгорания с использованием систем видеонаблюдения достигается как аппаратной частью (в большей степени характеристиками видеокамер), так и алгоритмическим обеспечением. В то же время, построение эффективных методов и алгоритмов для таких систем является достаточно сложной задачей, поскольку реальные видеопоследовательности характеризуются значительным количеством различных шумов и возмущений, имеющих схожие характеристики с заданными объектами: туман, солнечная засветка, световые блики различного характера, медленно движущиеся малоконтрастные объекты и т.д.

2.1 Общие принципы построения алгоритмов детектирования дыма лесного пожара на видео

Проведённый анализ существующих методов и алгоритмов обнаружения лесного пожара на видеопоследовательностях показывает, что они предполагают анализ таких основных свойств, как движение, цветовая и яркостная характеристики, хаотичность движения и др. для обнаружения дыма. В большинстве случаев, на первом этапе выполняется выделение областей-кандидатов, которые с достаточно высокой вероятностью могут относиться к областям дыма, а на втором этапе выполняется классификация данных областей. Для уменьшения ложного срабатывания видеодетектора лесного пожара используется постобработка. Основные отличия в существующих методах и алгоритмах решения подобной задачи заключаются в применении рассмотренных основных этапов и подходах к их реализации.

В таблице 4 представлен анализ современных алгоритмов (2002-2015гг.) детектирования основных признаков лесного пожара на динамических изображениях с использованием методики, предложенной в [28].

Таблица 4 – Современные алгоритмы обнаружения лесного пожара на видеопоследовательностях

ФИО авторов	Цветовая сегментация	Обнаружение движения	Анализ временных изменений границ	Анализ пространственных изменений яркости	Анализ движения	Блочная	Обучение	Постобработка
Z. Xiong [29]	-	+	+	-	+	-	-	-
Gomes-Rodriguez [30]	-	+	+	-	+	-	-	-
A. Genovese [31]	YUV	+	+	-	+	-	+	+
B. Ugur Toreyin [32]	YUV	+	+	-	+	+	+	-
D. Kim [33]	YUV	+	-	+	-	+	-	-
J. Vicente [34]	RGB	+	-	+	-	-	+	-
O. Gunay [35]	YUV	+	-	+	-	+	+	-
Y. Habiboglu [36]	YUV	+	-	+	-	+	+	+
B. Toreyin [37]	YUV	+	+	+	-	-	-	-
B. Toreyin [38]	YUV	+	+	-	+	-	-	-
P. Piccinini [39]	RGB	+	+	-	-	-	+	-
S. Calderara [40]	RGB	+	+	-	-	+	+	-
F. Yuan [41]	RGB	+	+	-	-	+	-	-
P. Morerio [42]	YCrCb	+	+	-	+	-	+	-

2.2 Алгоритм раннего обнаружения лесных пожаров на основе пространственно-временного анализа и метода опорных векторов

Разработан алгоритм обнаружения дыма лесного пожара на динамических изображениях, который требует выполнения следующих основных этапов:

детектирование медленно движущихся областей, цветовую сегментацию обнаруженных областей, вычисление ковариационных признаков сегментированных областей, классификацию полученных ковариационных признаков (рисунок 14).

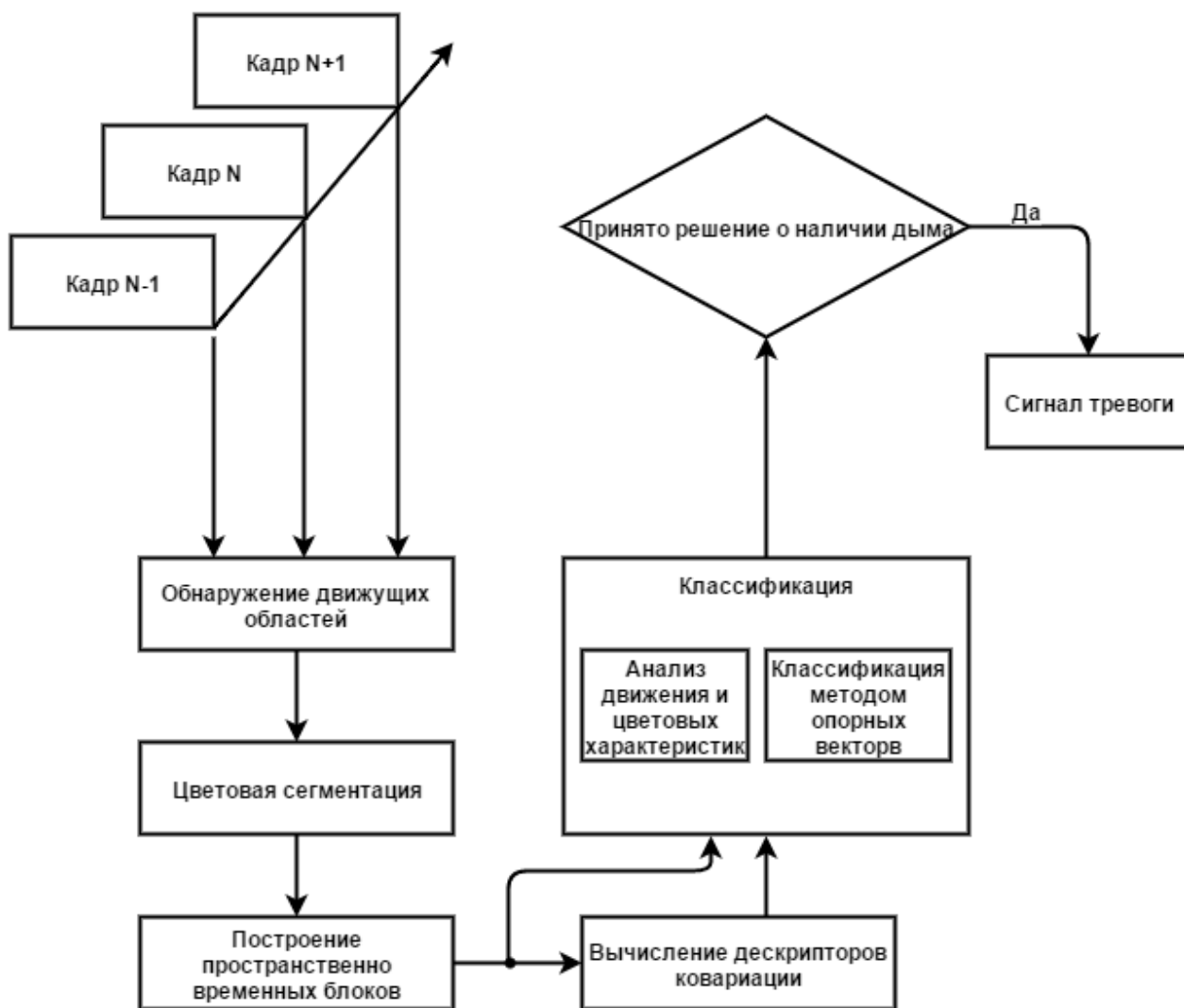


Рисунок 14 – Общая схема алгоритма обнаружения дыма лесных пожаров

Для обнаружения движущихся областей используется метод вычитания фона на основе смеси Гауссовых распределений. Данный алгоритм позволяет относить пиксель видеопоследовательности либо к фоновому кадру (BG), либо к объекту переднего плана (FG). Для того, чтобы адаптироваться к изменениям в кадре, обучающая выборка обновляется путем добавления новых объектов и удаления старых. Далее выбирается период времени T и в момент времени t формируются значения обучающей выборки $\{X_T = x^t, \dots, x^{t-T}\}$, где x^t - значение пикселя в момент времени t . Для каждого нового объекта обновляется обучающая выборка X_T и пересчитывается модель фона $p(x | X_T, BG)$.

Очевидно, что среди недавних объектов могут быть те, которые принадлежат к объектам переднего плана $p(x|X_T, BG + FG)$. Для их вычисления используется смесь Гауссовых распределений с M компонентами [43]:

$$p(x|X_T, BG + FG) = \sum_{m=1}^M \pi_m N(x; \mu_m; \sigma_m^2 I),$$

где μ_1, \dots, μ_M математические ожидания, а $\sigma_1, \dots, \sigma_M$ дисперсии, которые списывают Гауссовы компоненты, π_m - весовые коэффициенты

На рисунке 15 представлен результат работы метода вычитания фона на основе смеси Гауссовых распределений.

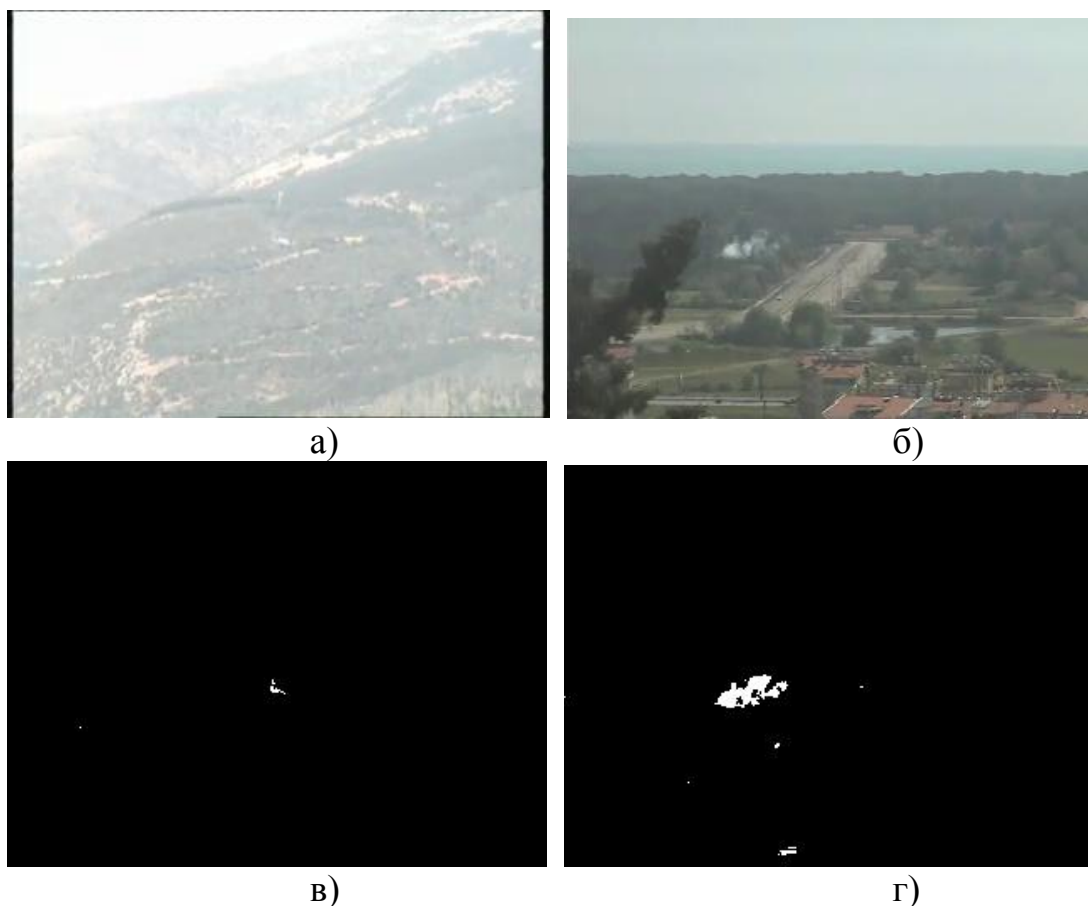


Рисунок 15 – Обнаружение движения на основе смеси Гауссовых распределений: а), б) кадры с лесным пожаром; в), г) бинарные маски движения

На этапе цветовой сегментации обнаруженных областей движения, в отличие от алгоритма, рассмотренного в [36], используется преобразование и сегментация в цветовом пространстве HSV. Переход от цветового пространства RGB к цветовому пространству HSV производится согласно условиям:

$$\text{if } (R == \max) \text{ then } H = \frac{G - B}{\max - \min},$$

$$\begin{aligned}
 & \text{if } (G = \max) \text{ then } H = 2 + (B - R) / (\max - \min), \\
 & \text{if } (B = \max) \text{ then } H = 4 + (R - G) / (\max - \min). \\
 & \quad H = H \times 60. \\
 & \text{if } (H < 0) \text{ then } H = H + 360 \\
 & V = \max, S = (\max - \min) / \max
 \end{aligned}$$

где \max , \min – максимальные и минимальные значения среди R , G и B .

Сегментация производится согласно условию:

$$\begin{aligned}
 & \text{if } ((100 < H) \ \&\& \ (H < 210)) \text{ and if } ((0.008 < S) \ \&\& \ (S < 0.5)) \\
 & \quad \text{and if } ((127 < V) \ \&\& \ (V < 255))
 \end{aligned}$$

В случае выполнения условия пиксель относится к области дыма. Пороговые значения для классификации в цветовом пространстве HSV подбирались эмпирическим путем. На рисунке 16 представлен результат сегментации в цветовом пространстве HSV.

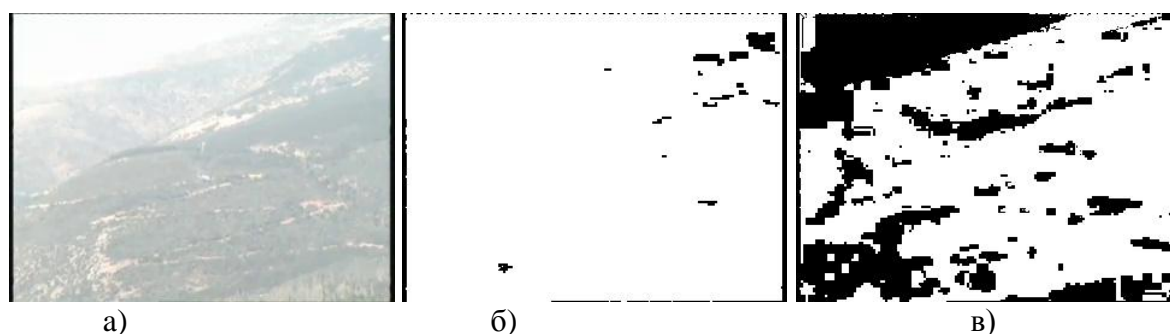


Рисунок 16 – Классификация дыма по цветовым признакам:
 а) в цветовом пространстве RGB; б) в цветовом пространстве YUV;
 в) в цветовом пространстве HSV

Для пространственно-временного анализа на видеоизображении используются распределенные во времени и нормализованные дескрипторы ковариации.

Для этого, первоначально происходит преобразование в цветовое пространство YUV и дескрипторы определяются в данном цветовом пространстве. Пусть $Y(i, j, n)$ интенсивность (i, j) – ого пикселя n -го кадра изображения пространственно-временного блока в видео, а $U_{i,j,n}$ и $V_{i,j,n}$ описывают значения цветовых характеристик соответствующего пикселя. В дополнение к пространственным параметрам применяются временные параметры I_t и I_{tt} , которые являются первыми и вторыми производными интенсивности пикселя относительно времени.

Свойства определённых параметров используются, для формирования ковариационной матрицы, представляющей пространственную информацию. Таким образом, ковариационные признаки формируются как:

$$\begin{aligned}
 Y_{i,j,n} &= Luminance(i, j, n), \\
 U_{i,j,n} &= ChrominanceU(i, j, n), \\
 V_{i,j,n} &= ChrominanceV(i, j, n), \\
 I_{i,j,n} &= Intensity(i, j, n), \\
 Ix_{i,j,n} &= \left| \frac{\partial Intensity(i, j, n)}{\partial i} \right|, \\
 Iy_{i,j,n} &= \left| \frac{\partial Intensity(i, j, n)}{\partial j} \right|, \\
 Ixx_{i,j,n} &= \left| \frac{\partial^2 Intensity(i, j, n)}{\partial i^2} \right|, \\
 Iyy_{i,j,n} &= \left| \frac{\partial^2 Intensity(i, j, n)}{\partial j^2} \right|, \\
 It_{i,j,n} &= \left| \frac{\partial Intensity(i, j, n)}{\partial n} \right|, \\
 Itt_{i,j,n} &= \left| \frac{\partial^2 Intensity(i, j, n)}{\partial n^2} \right|.
 \end{aligned}$$

Для определения значений ковариации в пространственно-временных блоках, видео разделяется на блоки размером $5 \times 5 \times F_{rate}$ (рисунок 17).

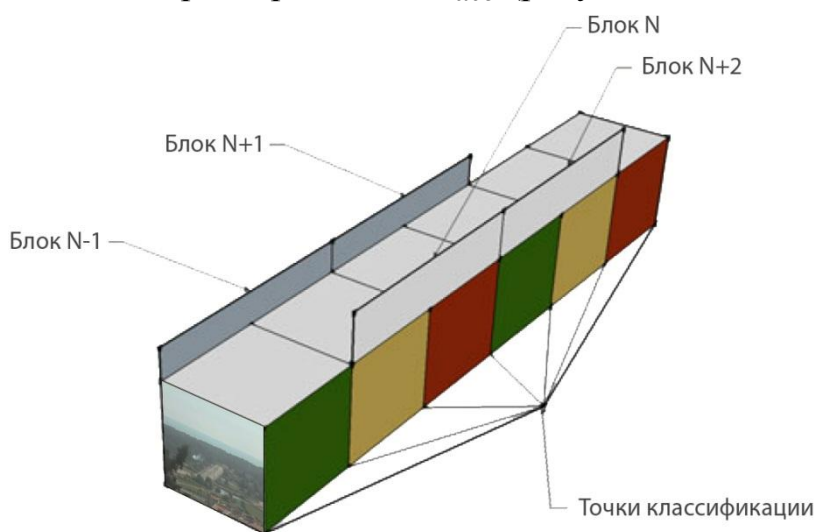


Рисунок 17 –Формирование пространственно-временных блоков

Вычисление параметров ковариации $\varphi(i, j, n)$ для каждого блока видеоизображения неэффективно, так как является ресурсоёмким процессом, поэтому используется наложение бинарной маски M , сформированной после обнаружения движущихся областей и цветовой сегментации:

$$\varphi(i, j, n) = \begin{cases} 1, & \text{if } M(i, j, n) = 1 \\ 0, & \text{otherwise} \end{cases}$$

Для ненулевых пикселей определяются следующие ковариационные признаки:

$$\Phi_{color}(i, j, n) = [Y(i, j, n); U(i, j, n); V(i, j, n)]^T,$$

$$\Phi_{ST}(i, j, n) = \begin{bmatrix} I(i, j, n) \\ I_x(i, j, n) \\ I_y(i, j, n) \\ I_{xx}(i, j, n) \\ I_{yy}(i, j, n) \\ I_t(i, j, n) \\ I_{tt}(i, j, n) \end{bmatrix}.$$

Значения ковариации извлекаются из ковариационной матрицы. Так как ковариационная матрица является симметричной, т.е. элементы симметричны относительно главной диагонали, то для дальнейших вычислений используются значения, расположенные на главной и ниже главной диагонали. Например, для матрицы $C(a, b)$, построенной из вектора свойств $\Phi_{color}(i, j, n)$ получим 6 значений ковариации (рисунок 18).

$$C(a, b) = \begin{bmatrix} C(1,1) & C(1,2) & C(1,3) \\ C(2,1) & C(2,2) & C(2,3) \\ C(3,1) & C(3,2) & C(3,3) \end{bmatrix}$$

Рисунок 18 – Выбор значений из ковариационной матрицы.

Аналогично, вектор движения $\Phi_{ST}(i, j, n)$ формирует 28 значений ковариации, следовательно, используется в дальнейшем 34 параметра ковариации. Из полученных векторов вычисляется ковариационная матрица. Затем путем нормализации ее нормализации получаем ковариационную матрицу $C(a, b)$, верхняя или нижняя треугольная часть которой образует вектор характеристик данного региона изображения.

Далее формируются исходные данные для обучения классификатора. При этом используются пространственно-временные блоки размером $5 \times 5 \times F_{rate}$, которые не перекрываются в пространственной области, но есть 50% перекрытие во временной

области, что позволяет выполнять классификацию с периодом $F_{rate}/2$. Блок передается в классификатор, если число пикселей в нем, относящихся к дыму, не менее 30% количества всех пикселей блока.

Классификация объектов осуществляется при помощи метода опорных векторов. В результате исследований эмпирическим путём определено, что лучше всего для решения поставленной задачи подходит функция ядра, основанная на радиальной базисной функции Гаусса.

2.3 Программная реализация и результаты исследования алгоритма

Для проведения экспериментов и оценки результатов исследований, разработанный алгоритм реализован на языке C++ с использованием среды разработки Microsoft Visual Studio 2013 и библиотеки компьютерного зрения OpenCV 3.0. Тестирование и анализ результатов работы алгоритма проводились с помощью ноутбука Dell Inspiron N5110N со следующими основными параметрами: центральный процессор - Intel Core i5-2410M с тактовой частотой 2,3ГГц, объем ОЗУ - 8,0ГБ, графический ускоритель – Nvidia GeForce GT 525M.

Эксперименты проведены на видеопоследовательностях, размещенных на сайтах <http://signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html>, <http://www.openvisor.org>, <http://cvpr.kmu.ac.kr>, http://wildfire.fesb.hr/index.php?option=com_content&view=article&id=62&Itemid=72.

Представленные на рисунке 19 кадры исследуемых видеопоследовательностей свидетельствуют, что использовались динамические изображения со сложным фоном (а,в), содержащие объекты, цвет и яркостные характеристики которых близки к характеристикам дыма (б,д). Кроме этого, исследовались видеопоследовательности (ж, и, к), на которых присутствовали дым с достаточно резким изменением направления и структуры, которое происходило из-за сильных порывов ветра. Результаты экспериментов показаны в таблице 5.

Таблица 5 – Результаты исследований алгоритма детектирования дыма лесного пожара

Видеоряд (см. рисунок 19)	Краткая характеристика видео	Дым присутствовал/ обнаружен с (номер кадра)	Дым найден/ присутствовал (число кадров)	Число кадров с ложным обнаружением / общее число
а)	Горит лес, туман, летают птицы	137/153	366/388	0/600
б)	Горит лес, движущиеся автомобили	172/277	1148/1265	0/1725
в)	Горит лес, туман	296/374	753/1104	0/1400
г)	Поселение, горят деревья	179/270	3116/3656	0/3835
д)	Поселение, горят деревья	946/1596	4933/5133	0/6079
е)	Открытая местность, задымление в траве, ветер	47/117	406/453	20/1125
ж)	Открытая местность, задымление в траве, движущийся человек	27/27	600/600	3/629

Продолжение табл. 5 - Результаты исследований алгоритма детектирования дыма лесного пожара

Видеоряд (рисунк 19)	Краткая характеристика видео	Дым присутствовал/ обнаружен с (номер кадра)	Дым найден/ присутствовал (число кадров)	Число кадров с ложным обнаружением / общее число кадров
а)	Открытая местность, задымление в траве	270/309	519/585	0/1725
б)	Задымление в лесу, дрожание камеры	1/140	10/150	75/150
в)	Задымление в лесу	1/141	299/500	0/500
д)	Задымление в лесу	1/129	322/500	0/500



Рисунок 19 – Примеры обнаружения дыма лесного пожара

Анализ результатов исследований показывает, что разработанный алгоритм обладает достаточно низким уровнем ложных тревог. Однако, при наличии схожих движущихся объектов, возможно ложное срабатывание. Алгоритм очень чувствителен к движению камеры, для корректной работы камера должна быть неподвижна. Наличие дыма с первого кадра видеопоследовательности увеличивает время обнаружения, так как методу вычитания фона нужно сформировать фоновый кадр. Идентификация лесного пожара происходит с минимальными временными задержками от 0 до 10 секунд. Согласно проведенному анализу, коэффициент ложноположительной классификации – от 0% до 1,7%.

3 Обнаружение объектов на видеопоследовательностях с большим разрешением на основе сверточных нейронных сетей

3.1 Сверточные нейронные сети для обнаружения объектов на основе их статических признаков

3.1.1 Модель AlexNet

Модель СНС AlexNet [44] включает восемь взвешенных слоев, из которых первые пять являются сверточными, а последние три являются полносвязными слоями. Данная модель была обучена на базе данных ImageNet ILSVRC-2010, содержащей 1.2 миллиона изображений, которые разбиты на 1000 классов. При этом, использовались искусственные расширения обучающей выборки, такие как сдвиг, вращение и удаление областей изображений. При тестировании сети на метриках top-1 и top-5 коэффициенты ошибок составили 67.4% и 40.9%, соответственно. Данная модель сети имеет ограничения при работе с многомасштабным анализом, подвержена переобучению из-за пропуска объектов на боковых выбросах [45], требует значительных вычислительных затрат. Для обучения использовались два графических процессора GTX 580 3GB, каждый из которых обрабатывал отдельно верхнюю и нижнюю части изображения, причем обмен результатами обработки на ограниченном наборе слоев [44].

3.1.2 Модель R-CNN

Модель СНС R-CNN предполагает выделение областей интереса на основе гипотез о местоположении объектов с использованием метода выборочного поиска. Далее размер регионов масштабируется к размеру входного слоя и обрабатывается СНС AlexNet. На последнем этапе выполняется бинарная классификация с использованием метода опорных векторов, которые были получены для каждого класса объектов. Повышение устойчивости модели к ошибочному делению объекта на фрагменты достигается за счет применения подавление не максимумов. При этом, совпадающие границы областей интереса на соседних фрагментах одного объекта удаляются из-за отсутствия существенного изменения величины градиента. Значение коэффициента ошибок при тестировании определяется величиной 15,3% в метрике ошибок top-5 [46]. Недостатком данной СНС является значительное время, затрачиваемое на обработку потока данных.

3.1.3 Модель Fast R-CNN

Для уменьшения временных затрат в 2015 году предложена модификация данного метода, Fast R-CNN [47], в которой используется усеченный метод опорных векторов. Особенностью Fast R-CNN является также представление регионов интереса в виде сверточной карты признаков. При этом используется 4-х размерный вектор $\{r, c, h, w\}$, где r, c - координаты верхнего левого угла, а h и w - высота и ширина региона соответственно. Далее карта признаков подается на слой

субдискретизации с размером окна 7×7 , выходом которого является максимальное значение для каждого положения окна. При обучении с использованием стохастического градиентного спуска подрегионы выбираются иерархически путем выборки N изображений с последующим определением R/N регионов интереса из каждого изображения. Для Fast R-CNN значения $R=128$, $N=2$. Обучение происходит более эффективно, применяя преимущества обмена признаками во время обучения. Данный метод быстрее осуществляет классификацию в выбранных областях интереса, но не учитывает существенные временные затраты на выделение этих областей перед обработкой в СНС.

3.1.4 Модель Faster R-CNN

Дальнейшее уменьшение временных затрат было предложено в модели Faster R-CNN [48]. Следует отметить, что в отличие от Fast R-CNN данная модель учитывает временные затраты на выделение объектов. В Faster R-CNN используется нейронная сеть для предположения о нахождении регионов интереса (Region Proposal Network, RPN). Вход сети RPN выделяет сверточные признаки, которые затем поступают на структуру, состоящую из двух полностью соединенных сверточных слоев. Первый слой является регрессионным и предсказывает прямоугольную область, ограничивающую объект. Второй слой является классификационным. Данная структура включена в сеть Faster R-CNN после последнего сверточного слоя. Для некоторых классов объектов качество работы СНС снижается, например, при ограничении метода для сегментации пешеходов [49]. Faster R-CNN не устойчива к мелким объектам и к зашумленным изображениям.

3.1.5 Модель GoogLeNet и модификации

Модель GoogLeNet [50] состоит более чем из 100 слоев, однако полностью подключенные слои не используются совсем. Также, по сравнению с AlexNet, количество обнаруживаемых параметров уменьшено в 12 раз, сверточные ядра большого размера заменены на последовательность ядер меньшего размера. В модели данной СНС выделяют блок Inception, который включает операции свертки с ядрами размером 5×5 , 3×3 , 1×1 и субдискретизации для окна размером 3×3 . Данный блок последовательно повторяется в архитектуре сети девять раз. Применение указанных выше ядер свертки позволяет извлекать признаки различных размеров в одном блоке. В данной модели по сравнению с R-CNN на первом этапе улучшен подход к выделению регионов интереса за счет включения в метод выборочного поиска алгоритма генерации множества ограничивающих рамок (multibox) [51]. В алгоритме применяется СНС AlexNet, которая генерирует заданное количество ограничивающих рамок для каждой области интереса, далее используется подавление не максимумов для исключения наименее совпадающих ограничивающих рамок с действительной областью интереса. При обучении СНС

GoogLeNet выборка искусственно расширялась путем многократного использования входных изображений, после чего результирующие значения многопеременной логической функции (softmax) усреднялись для идентичных изображений, что позволило улучшить результат классификации. Для данной модели СНС значение коэффициента ошибок в метрике top-5 составляет 6,7%. Однако, CoogLeNet не учитывает контекстную информацию обо всем изображении, так как на вход СНС подаются лишь области интереса. Также, выделение ограничивающих рамок расширенным методом выборочного поиска значительно замедляет обработку изображения.

В работе [52] предложена модификация блока Inception v2 для CoogLeNet за счет факторизации сверточного слоя 5×5 на 2 слоя с размером 3×3 , что позволило увеличить скорость вычислений в 9,36 раза. Дальнейшее применение разложения всех сверточных слоев $n \times n$ в два слоя с размерами $n \times 1$ и $1 \times n$ позволило уменьшить вычислительные затраты на 33%. Для исключения резкого уменьшения размерности карты признаков и снижения вычислительных затрат на ее обработку половина признаков, вычисленных на предыдущих этапах, подается на слой субдискретизации, а другая часть поступает на последующий сверточный слой. Данная модель также предполагает начальное задание минимальных весовых коэффициентов равномерно распределенных относительно каждого класса, сходное весовое значение которых составляет 0,1 для исключения переобучения и лучшей классификации.

Модификация Inception v3 характеризуется добавлением нормализации выборки из [53] на последних слоях и исключением технологии отсева (Dropout). Модель СНС состоит из 11 чередующихся блоков Inception v3 и при тестировании достигает коэффициента ошибки 4,2% для метрики top-5. Однако, по сравнению со своим ранним аналогом, затрачивает в 2.5 раз больше вычислительной мощности.

3.1.6 Модель ResNet и модификации

Рассмотренная в [54] архитектура СНС ResNet содержит начальный сверточный слой с размерностью ядра $[7 \times 7]$, чередующиеся сверточные слои с ядрами $[3 \times 3]$ и $[1 \times 1]$, а также обеспечивает возможность соединения по технологии быстрого доступа между входами чередующихся слоев и их выходами. В работе отмечено, что использование технологии быстрого доступа позволяет исключить ухудшение качества работы детектора при увеличении количества слоев СНС. Коэффициент ошибки составляет 3,57% в метрике top-5.

В 2016 году компания Google представила четвертую версию блока Inception v4, а также его модификации Inception-ResNet [55]. Применение технологии TensorFlow [56] позволило обучать модель GoogLeNet на одном устройстве, в отличие от предыдущих версий данной СНС, обучение которых выполнялось по частям с дальнейшим их соединением в полностью обученную модель. Inception-ResNet v1 использует модуль Inception v3 с включением технологии быстрого

доступа и сверточного слоя с ядром размерностью $[1 \times 1]$. Inception-ResNet v2 использует модуль Inception v4 с включением технологии быстрого доступа и сверточного слоя с ядром размерностью $[1 \times 1]$.

3.1.7 Модель YOLO и модификации

Представленная в 2016 модель СНС Yolo (You Only Look Once) [57] направлена на уменьшение вычислительных затрат при обработке. СНС включает сверточные слои с ядрами $[7 \times 7]$ и $[3 \times 3]$ для выделения признаков. Используются слои субдискретизации размером $[2 \times 2]$. За каждым сверточным слоем $[3 \times 3]$ размещен полносвязный слой $[1 \times 1]$, передающий координаты ограничивающих рамок и выходные вероятности о нахождении объекта и классовой принадлежности этого объекта. Два конечных полносвязных слоя выполняют задачу классификации. Размеры областей интереса во время обучения выбираются вручную. Модель делит входное изображение на сетку размером $[S \times S]$ частей (ячеек). Для каждой ячейки сетки предсказываются ограничивающие рамки и их весовые коэффициенты, которые характеризуют вероятность наличия объекта в ограничивающей рамке и величину этой вероятности ($Pr(Object)$). Если центр объекта попадает в ячейку сетки, то для нее устанавливается максимальная вероятность наличия рассматриваемого объекта. Каждая ограничивающая рамка характеризуется 5 параметрами: координатами центра относительно границ ячейки сетки (x , y), шириной (w) и высотой (h) ячейки, и ($Pr(Object)$). После разбиения каждая ячейка сети поступает на входной слой сети, масштабируясь под его размер. Во время тестирования вероятность наличия объекта в ограничивающей рамке, а также вероятность принадлежности объекта к конкретному классу умножаются, и их произведение дает классификационную оценку для каждого региона. Эти оценки кодируют как вероятность того, что этот класс появится в поле, так и насколько хорошо предсказанный регион подходит для объекта. В метрике top-5 точность данной модели составляет 88%. Недостатком модели является ручной выбор размера рамок, множество ошибок локализации для объектов малого размера, а также при перекрывающихся рамках проявляется нестабильность модели.

Для улучшения сегментации и классификации были предложены следующие модификации Yolo v2, Yolo9000[58], в которых предусмотрены:

- нормализация данных, позволяющая не использовать технологию отсева без опасения возникновения переобучения;
- повышение размерности классификатора для Yolo v2 до $[448 \times 448]$ для 10 эпох ImageNet;
- использование сети, выносящей предположение о нахождении регионов интереса (RPN) по аналогии с моделью Faster R-CNN;
- применение метода k -средних ($k=5$) для уточнения размера ограничивающих рамок и предварительной сегментации объектов в каждой области интереса;

- более точное предсказание местоположения осуществляется за счет передачи на последующие слои не собственных координат ограничивающих рамок, а их смещения относительно верхнего левого угла исходного изображения;
- вычисление детализированных признаков с разрешением карты признаков $[26 \times 26]$;
- многомасштабное обучение, которое позволяет сети сегментировать и классифицировать объекты при разных разрешениях, исключая полностью связанные слои в архитектуре СНС;
- использование новой классификационной модели Darknet-19, которая включает 19 сверточных слоя и 5 слоев субдискретизации.

Yolo9000 имеет ту же архитектуру, что и Yolo v2, однако количество выходных гипотез ограничивается 3. Модификация архитектуры улучшенной версии Yolo заключается в замене 2 последних полносвязных слоев исходной архитектуры на сеть, выносящую предположения о наличии регионов интереса (Region Proposal Network, RPN) и применении метода k -средних (причем, $k=5$) для каждой ячейки. Таким образом, каждая ячейка может включать 5 объектов, в то время как первая версия Yolo ограничивалась одним объектом для ячейки. Далее выделенные области интереса поступают на вход классификатора Darknet-19, в котором используется 19 сверточных слоев размером $[3 \times 3]$ и $[1 \times 1]$, а также 5 слоев субдискретизации $[2 \times 1]$. Обучение Yolo 9000 выполнено на совместной базе данных (ImageNet и COCO), состоящей из 9000 классов. Вероятность правильного обнаружения для данной модели достигает значения 91,2% в метрике top-5, однако при этом вычислительные затраты значительно больше, чем у СНС Yolo. Для некоторых классов объектов, например, «человек», «одежда» вероятность правильной классификации уменьшается.

3.2 Сверточные нейронные сети для обнаружения объектов с учетом их динамических признаков

3.2.1 Модель Seq-NMS

СНС Seq-NMS [59] разработана на основе СНС Faster R-CNN, ZFnet, VGG16 и предполагает выполнение следующих шагов: выделение областей интереса, их классификацию, поиск соответствующих областей в соседних кадрах, нахождение максимальных значений пересечения над объединением областей интереса (IoU) для каждой пары соседних кадров, построение трехмерной области интереса (3D-ОИ) с учетом движения для каждой области, изменение значений IoU для каждой области в 3D-ОИ, дальнейший поиск 3D-ОИ без учета уже имеющихся, а также без учета областей интереса, имеющих значение IoU выше порогового. Для детектирования использовалась СНС ZFnet [60] совместно с RPN (Region Proposal Network) по аналогии с СНС Faster R-CNN. Классификация осуществлялась с помощью СНС ZFnet и VGG16 [61], результаты экспериментов показали, что VGG16 лучше справляется с задачей классификации. Seq-NMS отличается большим

количеством ошибок локализации а также значительно проигрывает в значении средней точности второго порядка (Mean average precision, mAP).

3.2.2 Модель T-CNN

Модель T-CNN (Tube-Convolutional Neural Network) [62] предложена для создания быстро обучаемого и точного алгоритма детектирования и классификации объектов на видео. Данная модель вычисляет сверточные признаки в трех последовательных кадрах, а затем использует СНС для слежения за ними. Объекты детектируются путем использования улучшенного метода Faster R-CNN, в котором Fast R-CNN и RPN соединены каскадно. Для слежения за областью интереса, выделенной модифицированным Faster R-CNN, применяется метод [22], который использует пару параллельных СНС имеющих два сверточных слоя с ядрами $[9 \times 9]$ и $[5 \times 5]$ соответственно, выходом которых являются тепловые карты (heated map), характеризующие наличие объектов переднего фона на рассматриваемом кадре видеопоследовательности. Разное время обновления рассматриваемых кадров в этих СНС позволяет осуществлять слежение за движущимися объектами. Таким образом, формируются трехмерные области интереса (3D-ОИ), которые классифицируются Байесовским классификатором. В архитектуре используется контекстное ограничение на классификацию, т.е. предположение о том, что на видеопоследовательности присутствует небольшой набор коррелированных между собой классов, поэтому предположения о наличии отличных от данного набора классов можно не учитывать при классификации.

В работе [64] представлена новая архитектура сети T-CNN основанная на модели 3DConvNet, которая базируется на подходе Faster R-CNN. Для учета динамических изменений объекта относительно заднего фона видеокадра используются трехмерные сверточные признаки, а также трехмерные слои субдискретизации. Методика заключается разбиении видеопоследовательности на фрагменты по 8 кадров, определении для каждого фрагмента 3D-ОИ и дальнейшем их объединении. Предложенная архитектура состоит из семи слоев 3D-свертки и четырех слоев 3D субдискретизации. Размер окна установлен как $[1 \times 2 \times 2]$ для первого слоя 3D субдискретизации, $[2 \times 2 \times 2]$ для оставшихся слоев, а для сверточных слоев размер ядра составляет $[3 \times 3 \times 3]$. Так, после последнего сверточного блока, полученные двумерные области интереса дублируются на все 8 кадров, таким образом формируя 3D-ОИ. Для каждой выделенной области применяется метод k-средних (при $k=12$) для уточнения расположения ограничивающих рамок, а каждая рамка характеризуется бинарной оценкой присутствия движения. Ограничивающие рамки с уровнями движения меньше порога не рассматриваются в дальнейшем. Далее две выделенных 3D-ОИ поступают на усредняющий слой субдискретизации (average pooling) и передаются на сверточный слой $[1 \times 1]$. Три полносвязных слоя

обрабатывают каждую 3D-ОИ, выходной информацией являются смещение высоты, ширины и центральной координаты каждого ограничивающей рамки в каждом кадре. Описанный подход, учитывает оценку движения за счет использования 3D-ОИ, однако отличается большими временными затратами на обучение и обработку.

3.2.3 Модель DFFCNN

В работе [64] представлена модель сверточной нейронной сети DFFCNN (Deep Feature Flow CNN), которая является комбинацией нескольких методов для детектирования и классификации объектов на статичных изображениях совместно с CNN-подходом к построению оптического потока. Для вычисления оптического потока использована СНС FlowNet [65], а именно, ее упрощенная версия. Также предлагается уменьшенная в 2 раза модель FlowNet и модель FlowNet, дополненная моделью Inception, которая состоит из Inception и Reduction блоков, сверточных слоев с ядрами размером $[7 \times 7]$ и $[3 \times 3]$ и слоя субдискретизации для окна $[3 \times 3]$. Данная модификация позволила уменьшить сложность в 8 раз по сравнению со стандартной версией FlowNet. Для выделения признаков в DFFCNN использовалась модифицированная СНС ResNet с 50 и 100 слоями. В модифицированной СНС ResNet отсутствует последний полносвязный классификационный слой, для некоторых сверточных блоков изменен шаг, а для сохранения размерности карт признаков между слоями применен «дырочный алгоритм» (holing algorithm) [26]. Выходом модифицированной СНС ResNet является 1024 размерная карта признаков, которая обрабатывается сверточным слоем $[1 \times 1]$ и последующим слоем многопеременной логической функции (softmax layer) для выполнения семантической сегментации. Для детектирования объектов используется R-FCN [27], который представляет собой модель Resnet-101 без конечного полносвязного слоя, также в архитектуру введена RPN по аналогии с Faster R-CNN, а выходом последнего сверточного слоя является набор из 1024 признаков.

В модели FGFA [67] движение компенсируется на основе простой версии FlowNet. Для выделения признаков применяется СНС, которая использует модифицированный блок Inception-ResNet - Aligned-Inception-ResNet. Aligned-Inception-ResNet расширен слоями нормализации выборки (Batch Normalization) и полулинейной функцией активации (Rectifier Linear Unit). Для детектирования объектов используется R-FCN. В архитектуру FGFA включена встроенная трехслойная сверточная сеть с адаптивными весами, которая используется для характеристики отношения кадров, использующихся для вычисления оптического потока к исходному кадру. Если признак обрабатываемого кадра коррелирован с признаком, имеющим такие же координаты на исходном кадре, то нейрон характеризуется большим весом, чем признак с меньшей степенью корреляции. Архитектура встроенной трехслойной сети состоит из двух сверточных слоев с

ядрами $[1 \times 1]$ и входными размерностями сверточных карт 512 и 2048, соответственно, между которыми расположен слой с размером ядра $[3 \times 3]$, входная карта признаков которого 512 элементов.

В таблице 6 представлены основные характеристики СНС, учитывающие динамические признаки объектов.

Таблица 6 - Основные характеристики СНС, учитывающих движение объектов

Тип модели СНС	Seq-NMS	T-CNN & Bayesian classifier	T-CNN	DFFN	FGFA
mAP, %	48,2***	67,82	39,2...86,69	73,1	76,3
Кол-во слоев	18	22	11	Более 100	Более 100
Время на обучение, ч.	-	264	1344	-	-
Время на обработку кадра, с.	-	0,09	0,253	0,073	0,773
Размер кадра	-	386×386	300×400	600×600	600* 300**
Кол-во кадров для анализа движения	2	3	8	20	20

* Наименьший размер входного кадра для сети выделения признаков

** Наименьший размер входного кадра для сети оптического потока

*** Для классификации использовалась СНС *VGG16*

Применение данных моделей для обработки видеоизображений большого разрешения требует масштабирования входного изображения к размерам входного слоя. Поэтому для изображений разрешения 4K масштабирование к размерам входного изображения СНС приведет к невозможности обнаружения объектов небольшого размера. Следовательно, актуальным является разработка алгоритмов с учетом специфики существующих СНС для обработки видео большого разрешения.

3.3 Блочный алгоритм обнаружения объектов на видеопоследовательностях с большим разрешением на основе СНС YOLO

3.3.1 Описание алгоритма

Применение данных моделей для обработки видеоизображений большого разрешения требует применения масштабирования входного изображения к размерам входного слоя, что приводит к потере объектов небольшого размера. Для устранения данного недостатка предлагается модифицированный алгоритм

обнаружения объектов на видеопоследовательностях с большим разрешением на основе СНС YOLO, который требует выполнения следующих основных шагов:

1) Исходное изображение I размером $H \times W$ делится на пересекающиеся между собой ячейки $C_{i,j}$ размером $ch \times cw$, $i = \overline{0, H/ch-1}$, $j = \overline{0, W/cw-1}$ с шагом равным размеру ячейки и с 10% перекрытием соседними. На рисунке 1 представлен пример разбиения изображения для $S=3$.

2) Классификация изображений ячеек. Для изображения ячейки $C_{i,j}$ определяется вероятность принадлежности $\Pr(\text{Class}_l | C_{i,j})$ только одному из классов Class_l , где $l = \overline{0, classNumber-1}$, $classNumber$ - общее количество классов для классификации. Для каждой из ячеек $C_{i,j}$ объявляются регионы интереса $B_{i,j}^k$, где $bbNumber$ - количество регионов интересов для ячейки, $k = \overline{0, bbNumber-1}$. Регионы интереса представляют собой фрагменты изображения фиксированных размеров, которые для всех ячеек одинаковые, причем их центры устанавливаются на центр ячейки. Каждый регион интереса $B_{i,j}^k$ определяется: координатами $B_{i,j}^k(x, y)$ - центром изображения региона интереса относительно центра ячейки $C_{i,j}$; $B_{i,j}^k(w, h)$ - шириной и высотой относительно всего изображения I ; достоверностью прогноза $\Pr(B_{i,j}^k)$ - вероятностью наличия объекта в ограничивающей рамке. Если в ячейке $C_{i,j}$ не существует объекта, то $\Pr(B_{i,j}^k) = 0$.

3) Постобработка соседних ячеек. Поиск соседних ограничивающих рамок у которых существует общая пересекающаяся область, расположенная ближе? чем на 10% от края ячеек. Если такие ячейки найдены, то для них вычисляется параметр характеризующий степень наложения регионов интереса друг на друга, пересечение над объединением (intersection over union):

$$Iou = \frac{In}{C_{i_0, j_0}(w) \times 0.5 \cdot C_{i_0, j_0}(h) + C_{i_1, j_1}(w) + 0.5 \cdot C_{i_1, j_1}(h) - In},$$

где:

$$In = \begin{aligned} & \min(C_{i_0, j_0}(x) + 0.5 \cdot C_{i_0, j_0}(w), C_{i_1, j_1}(x) + 0.5 \cdot C_{i_1, j_1}(w)) - \\ & - \max(C_{i_0, j_0}(x) - 0.5 \cdot C_{i_0, j_0}(w), C_{i_1, j_1}(x) - 0.5 \cdot C_{i_1, j_1}(w)) \times \\ & \times \min(C_{i_0, j_0}(y) + 0.5 \cdot C_{i_0, j_0}(h), C_{i_1, j_1}(y) + 0.5 \cdot C_{i_1, j_1}(h)) - \\ & - \max(C_{i_0, j_0}(y) - 0.5 \cdot C_{i_0, j_0}(h), C_{i_1, j_1}(y) + 0.5 \cdot C_{i_1, j_1}(h)) \end{aligned}$$

Если $Iou > T$, то выполняется объединение ячеек C_{i_0, j_0} и C_{i_1, j_1} , где T - коэффициент задающий степень идентичности координат для объединения.

На рисунке 20 приведена обобщенная схема предложенной модификации.

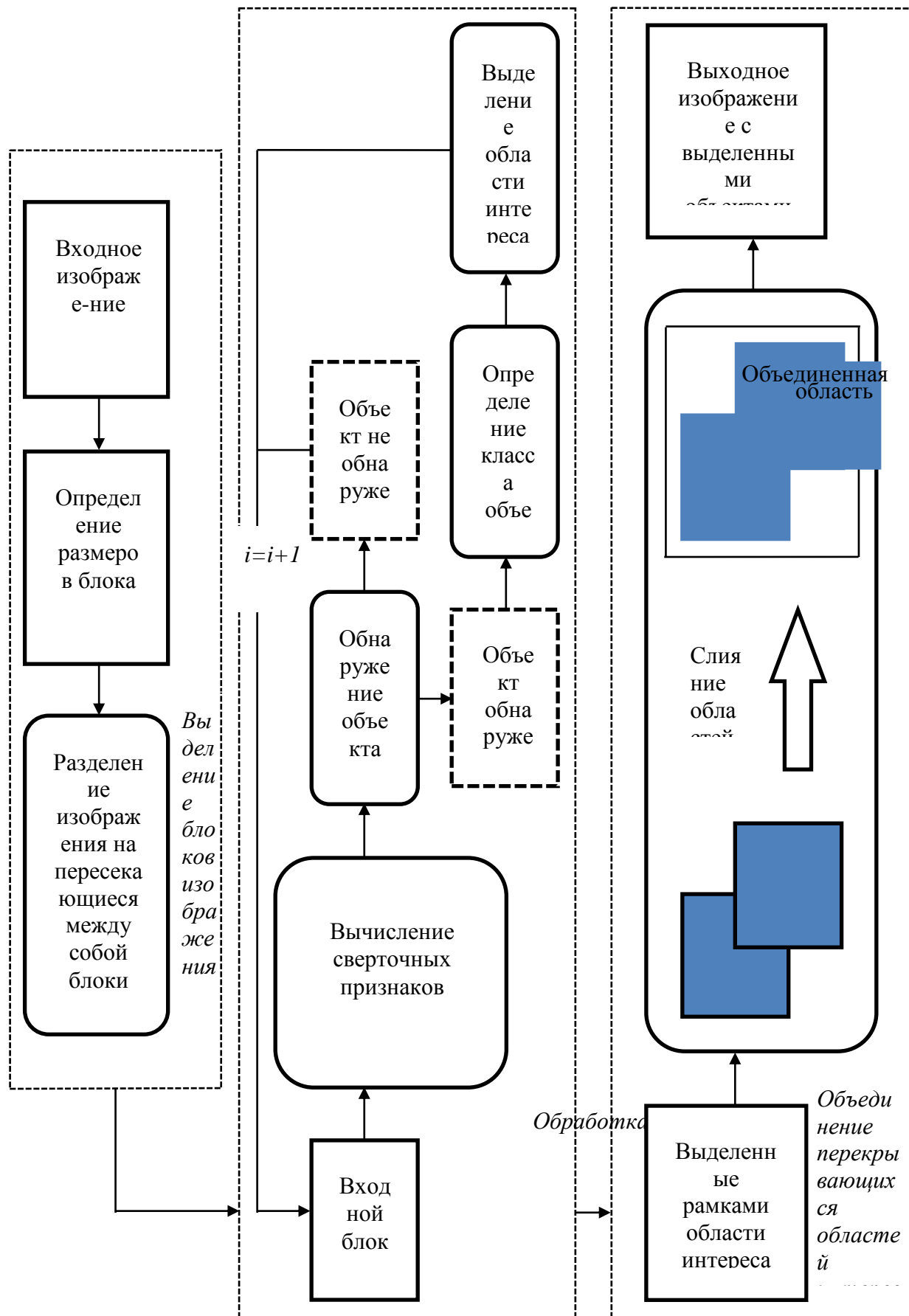


Рисунок 20 - Схема предложенного алгоритма обнаружения объектов

3.3.2 Результаты исследований

Для обучения СНС использовался набор изображений PASCAL VOC [68], размер сетки $S=7$, количество гипотез для каждой ячейки $bbNumber=2$, количество классов для классификации $classNumber=20$, а именно, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor. Примеры изображений из базы PASCAL VOC, обработанных алгоритмом YOLO показаны на рисунке 21.

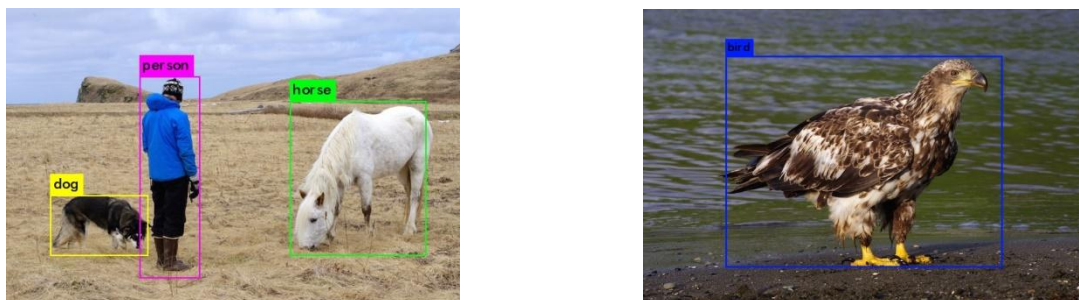


Рисунок 21 – Примеры используемых классов изображений

Для проведения экспериментов разработанный алгоритм реализован на языке C++ с использованием компилятора GCC и библиотеки компьютерного зрения OpenCV 3.1.

На рисунке 22 демонстрируется результат локализации регионов интереса на изображении размером 3872×2592 . Из рисунка 22 видно, что возможно деления одного объекта на две части, если они смещены относительно друга.

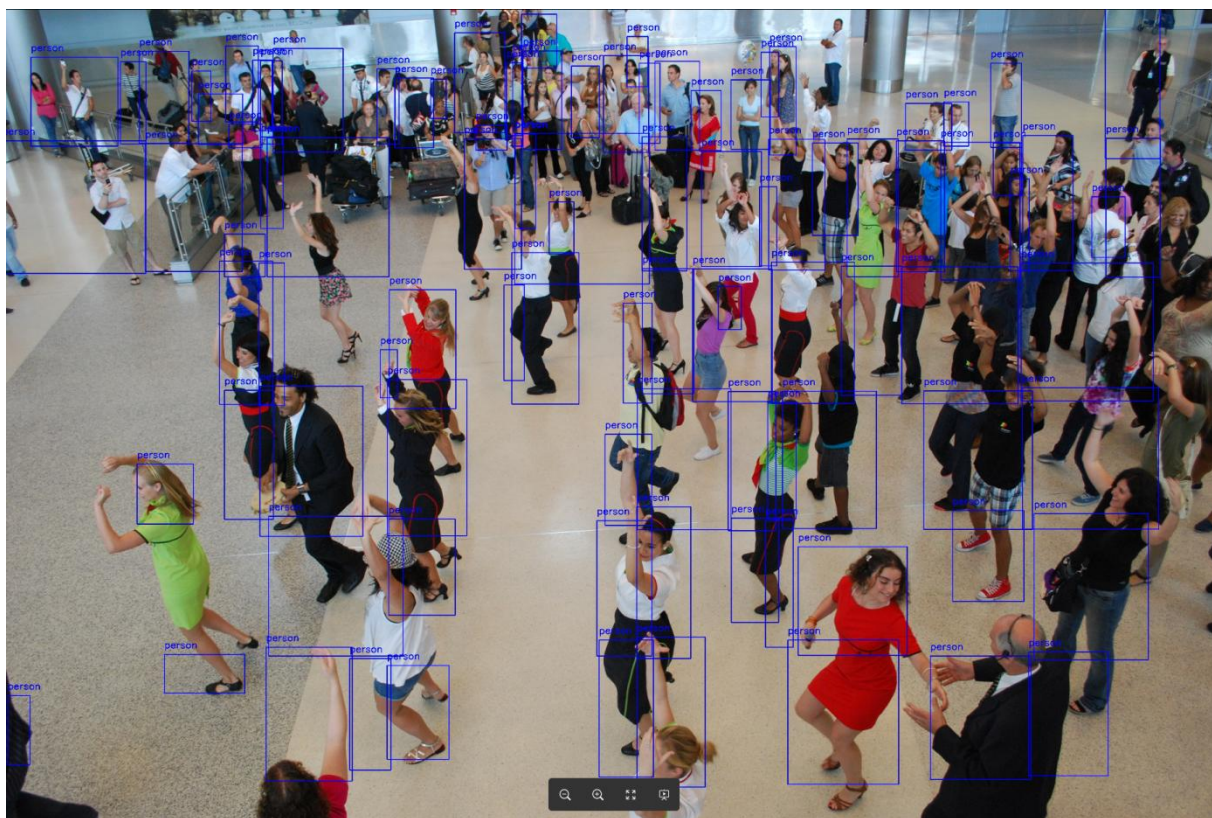


Рисунок 22 - Результат нахождения регионов интереса

На рисунке 23 представлены примеры локализованных объектов для кадра, приведенного на рисунке 22, после объединения регионов интересов.

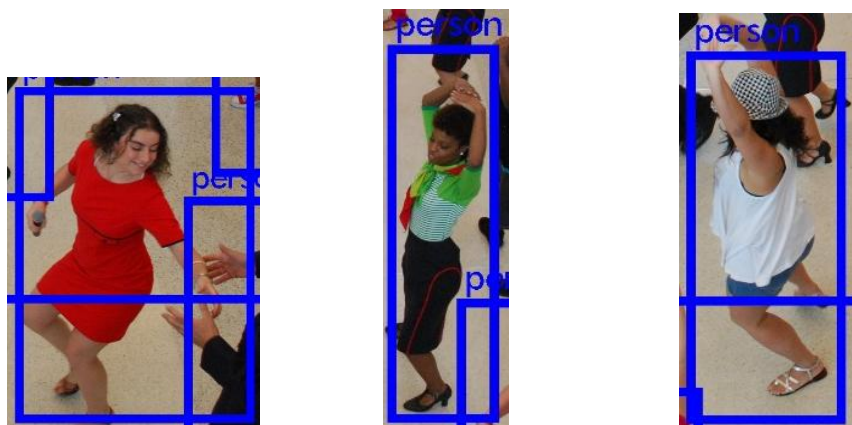


Рисунок 23 - Пример объединения регионов интересов

Для оценки ошибки локализации объекта использовалась формула:

$$\delta E = \sqrt{\frac{\sum_{i=1}^N (X_{i_b} - X_i)^2}{N}} \cdot 100\%$$

где i - номер изображения в наборе данных, N - число объектов на изображении, X_{i_b} - значение эталонной координаты, X_i - значение координаты, определенной на основе применяемого алгоритма.

Результаты тестирования алгоритма YOLO и предложенной модификации на 100 изображениях, приведены в таблице 7.

Таблица 7 – Экспериментальные результаты по оценке основных характеристик алгоритма YOLO и предложенной модификации

Тип алгоритма	Точность обнаружения			Время обработки кадра, с		
	1024×1024	2048×2048	4096×3072	1024×1024	2048×2048	4096×3072
На базе СНС YOLO	57%	48%	31%	0,11		
Блочный алгоритм	49%	47%	42%	0,46	1,72	2,64

Анализ таблицы показывает, что с возрастанием размерности входного кадра предложенная модификация работает эффективнее алгоритма YOLO.

Для оценки временных затрат работы алгоритма проведены экспериментальные исследования с использованием компьютера с основными параметрами i7 4.3 ГГц, ОЗУ 32 Гб. видеокарта Nvidia GeForce GTX 1070 .

На рисунке 24, демонстрируется сравнение работы оригинального алгоритма YOLO и модифицированного. Для исследований использовались динамические

изображения со сложным фоном больших размерностей и с разным количеством объектов.

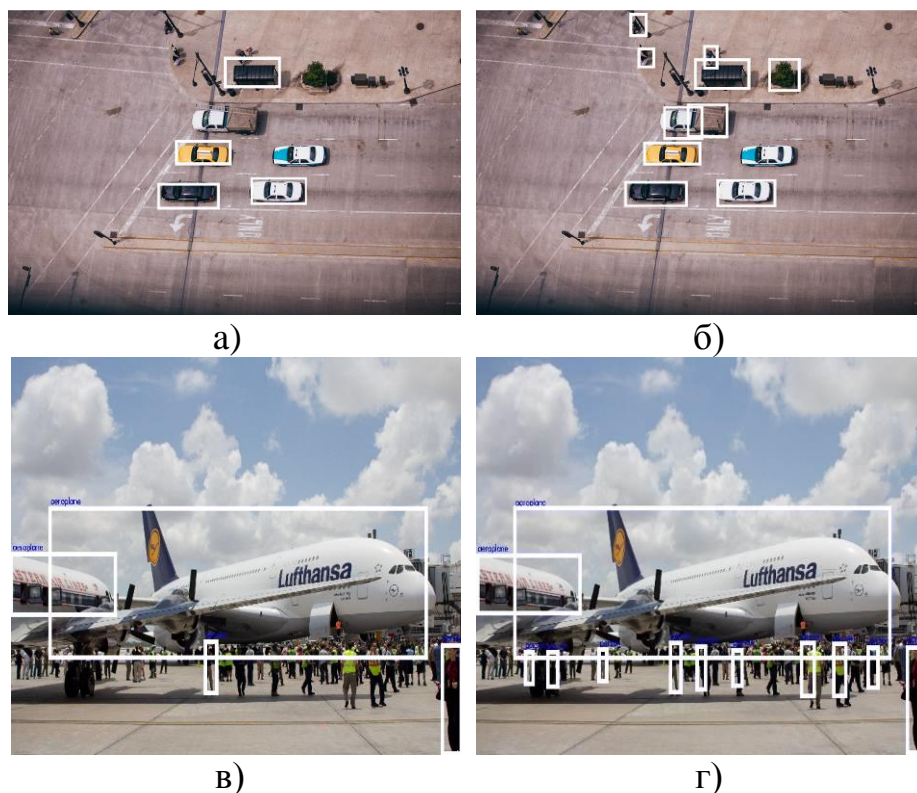


Рисунок 24 – Сравнение результатов обнаружения объектов: а, в) на основе СНС YOLO; б, г) на основе предложенного блочного

3.3.3 Модификация блочного алгоритма

В работе [69] для решения задачи обнаружения объектов на изображениях большого разрешения предложен алгоритм на основе разбиения входного кадра на фрагменты (блоки), которые подаются на вход СНС. Однако, при таком подходе один объект может быть разделен на несколько частей, которые в таком случае классифицируются как независимые различные объекты. На рисунке 25 показан пример ложной классификации носа самолета (класс «car») в качестве автомобиля. Это связано с тем, что вычисленные СНС признаки выделенного фрагмента более схожи с признаками, соответствующими объекту «автомобиль», чем объекту «самолет».

Для решения указанной проблемы предлагается построение и использование групп коррелированных классов объектов (объекты с большим количеством схожих признаков), что позволит значительно уменьшить вероятность фрагментарной классификации одного объекта для граничных областей блоков.



Рисунок 25 - Пример ложной классификации фрагмента объекта

Для базы данных Pascal VOC сформированы следующие группы коррелированных объектов: Aeroplane, Bird, Bus, Car, Train; Bicycle, Motobike; Bird, Aeroplane; Boat; Bottle; Bus, Car, Aeroplane, Train; Car, Aeroplane, Train, Bus; Cat, Dog; Chair, Sofa; Cow, Horse; Dinningtable; Person; Pottedplant; TVmonitor .

На рисунке 26 показана разработанная схема взаимосвязей коррелированных классов объектов для базы данных Pascal VOC.

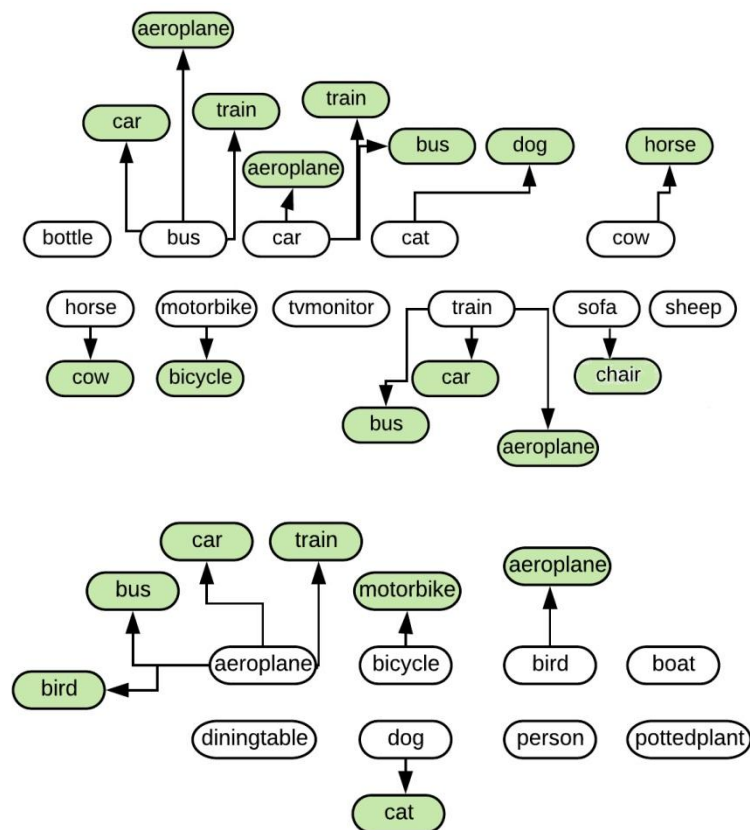


Рисунок 26 - Схема взаимосвязей коррелированных классов объектов для базы данных Pascal VOC

Разработанный алгоритм обнаружения объектов на видеоизображениях с разрешением 4K и выше, основанный на блочной обработке кадра с использованием сверточной нейронной сети Yolo в любой модификации и использовании коррелированных объектов требует выполнения следующих шагов:

1. Расчет размера блока. Размер блока B_i выбирается исходя из минимизации ложноотрицательного обнаружения и определяется как:

$$B_i = \frac{I_i}{I_1},$$

где: I_i –размер исходного изображения в пикселях, для разрешения 4K максимальный размер длинны или ширины варьируется от 3600 до 4200 пикселей; I_1 – размер входного слоя СНС, для СНС Yolo v3 он составляет [416×416] пикселей.

2. Обнаружение областей интереса на фрагментах кадра. Для этого каждый блок поступает на вход СНС Yolo. Выходным результатом для каждой области интереса, которая содержит объект или фрагмент объекта, является набор данных вектор $[X_1, Y_1, X_2, Y_2, Confidence, Class_index, Bn]$, где X_1, Y_1 — координаты левого верхнего угла области интереса, X_2, Y_2 — координаты правого нижнего угла области интереса, $Confidence$ — степень уверенности системы, $Class_index$ — индекс класса объекта, Bn — номер блока в котором осуществлялось обнаружение.

3. Нахождение координат обнаруженных областей на кадре. Обнаруженные координаты областей интереса на фрагментах кадра, записанные в текстовый документ, пересчитываются на координаты входного кадра:

$$Add_X = B_n \bmod s;$$

$$Add_Y = B_n \div s,$$

где Add_X и Add_Y – добавочные значения для X и Y координат соответственно, s – количество разбиений для одной стороны кадра

4. Объединение областей интереса. Для объединения соседних областей интереса рассчитывается коэффициент, характеризующий степень наложения областей интереса друг на друга. Если рассчитанное значение коэффициента выше порогового и объекты относятся к одной группе, то данные области интереса объединяются в один объект. В данной работе пороговое значение выбиралось аналитически и составило 50%. Класс объекта определяется максимальной величиной $Confidence$ для объединяемых областей.

Рисунок 27 свидетельствует о том, что предложенный алгоритм позволил исключить ложное обнаружение автомобиля, как показано на рисунке 25.

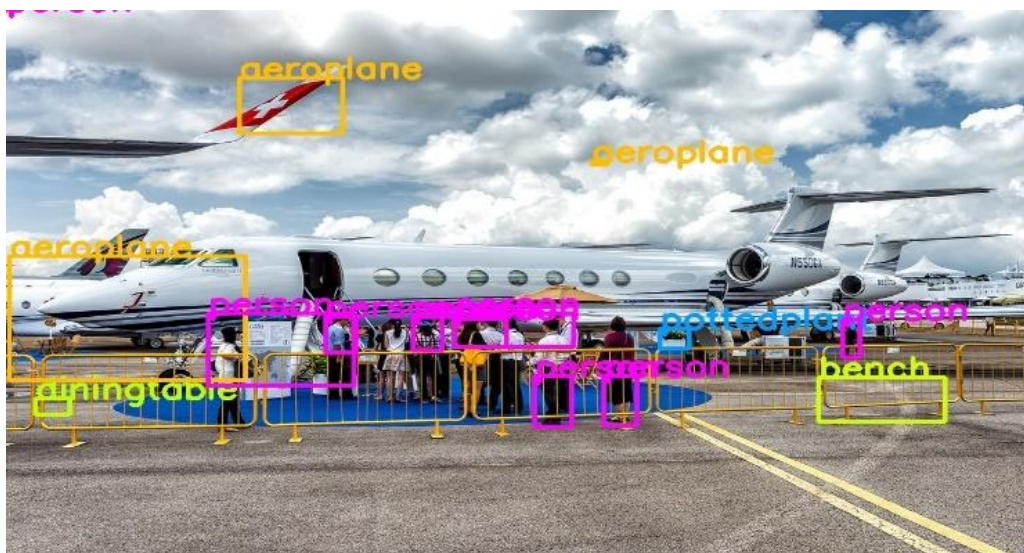


Рисунок 27 – Пример обнаружения и классификации объектов на основе предложенного алгоритма

В целом, результаты экспериментов показали, что предложенный алгоритм позволяет уменьшить вероятность фрагментарной классификации одного объекта для граничных областей блоков, возникающих из-за разбиения входного кадра на фрагменты. Однако, наблюдается некоторое количество ложноположительных результатов обнаружения. Этот недостаток можно исключить, переобучив нейронную сеть на выборке, включающей фрагменты изображений.

4 Анализ изображений микроструктуры металлов

Для распознавания структуры материала при металлографическом анализе используется ГОСТ 5639-82 «Стали и сплавы. Методы выявления и определения величины зерна», который содержит методы для выявления и определения величины зерна [70].

Метод подсчета зерен заключается в определении количества зерен на единице поверхности шлифа и расчете их средних площади и диаметра [71]. Для этого поле зрения на микроснимке ограничивают определенной плоской фигурой с заданными размерами, подсчитывают число зерен попавших внутрь данной фигуры и число, пересеченных ее границами. Далее определяют среднюю площадь сечения и средний диаметр зерна.

Метод подсчета пересечений границ зерен состоит в подсчете зерен, пересеченных отрезком прямой и определении среднего условного диаметра - в случае равноосных зерен или количества зерен на 1 мм^3 — в случае неравноосных зерен [70]. Для этого на микроснимках проводят несколько отрезков произвольной длины и подсчитывают точки пересечений отрезков прямых линий с границами зерен. После чего для равноосных зерен вычисляется средний условный диаметр. Для подсчета количество неравноосных зерен отрезки прямых проводят параллельно осям симметрии.

Метод измерения длин хорд основан на измерении линейных размеров отрезков - хорд, отсекаемых в зернах прямыми линиями, и применяется для определения величины зерна в разнотельной структуре [70]. На микроснимке проводится не менее трех прямых в произвольном направлении и определяется относительная доля зерен.

Для определения характеристик углеродистой стали важным является микроструктурный анализ, т.к. микроструктура углеродистых сталей, полученная после медленного охлаждения, характеризуется равновесным состоянием. Для этого случая можно с достаточной точностью для практических целей определить процентное содержание углерода на основе анализа площадей, занимаемых на микрошлифе ферритом и перлитом[71].

Проведение подобного анализа изображений требует большого количества времени, поэтому актуальной является задачей автоматизации вычислений на основе методов цифровой обработки видеоизображений.

4.1 Этапы обработки

4.1.1 Предварительная обработка

Изображения специально подготовленных микрошлифов металлов, полученные использованием микроскопа либо USB-камеры (пример показан на рисунке 28) требуют предварительной обработки для реализации методов анализа их структуры.

Так как исходные изображения могут характеризоваться различного типа шумами, а также с учетом используемых методов анализа, предварительная обработка включает: медианную фильтрацию, высокочастотную фильтрацию, выделение краев, бинаризацию, расширение, заполнение, очистку.

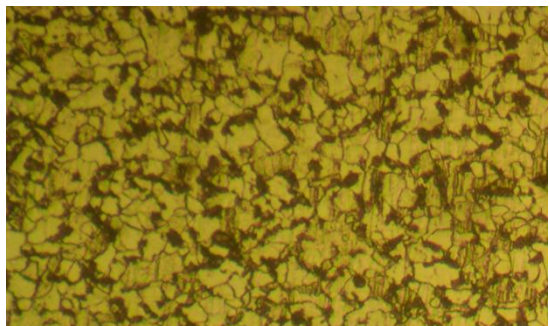
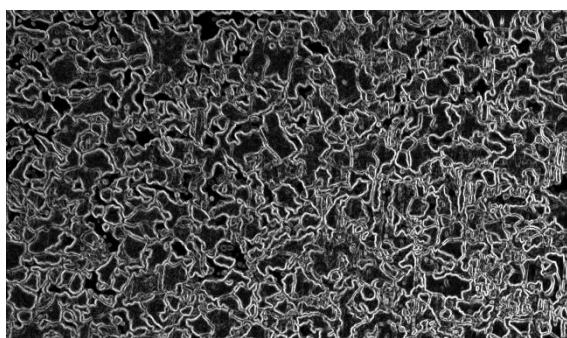
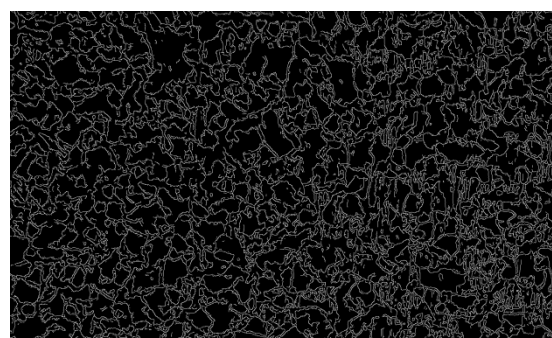


Рисунок 28 – Исходное изображение микрошлифа

Для выделения краев экспериментально установлено, что наиболее эффективен оператор Кэнни, т.к. позволяет сохранять мелкие детали и получать границы толщиной в один пиксель. На рисунке 29 показаны примеры выделения краев с использованием оператора Собела и Кэнни.



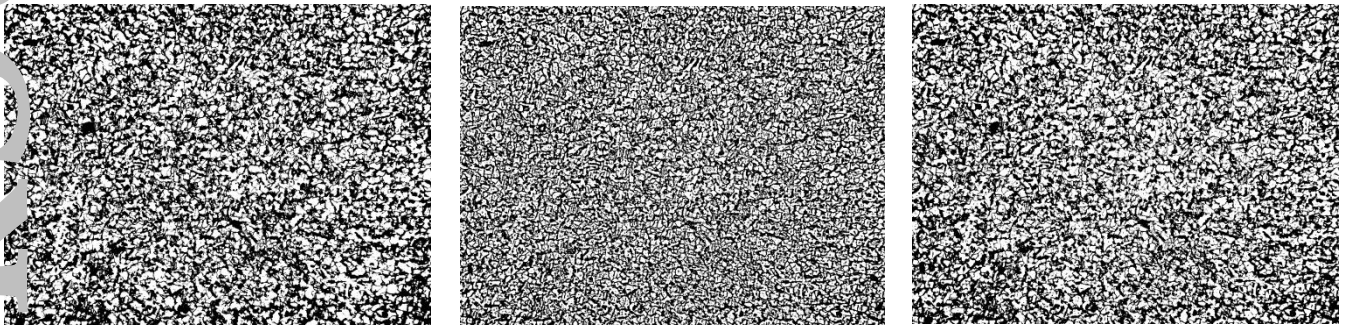
а)



б)

Рисунок 29–Результаты выделения краев: а) оператором Собела; б) оператором Кэнни

Для определения эффективного алгоритма бинаризации исследованы адаптивные алгоритмы Кристиана, Бернсена и Брэдли на реальных изображениях микрошлифов. С учетом особенностей такого типа изображений результаты экспериментов показали, что наиболее эффективен метод Брэдли (пример на рисунке 30а). После применения алгоритма Кристиана на изображении присутствуют шумы (рисунок 30б), после применения алгоритма Бернсена присутствуют разрывы границ (рисунок 30в), из-за этого при подсчете зерен несколько областей будут приняты за одну.



а)

б)

в)

Рисунок 30 – Примеры бинаризации: а) алгоритмом Брэдли; б) алгоритмом Ористиана; в) алгоритмом Бернсена.

4.1.2 Подсчет зерен

Метод подсчета зерен заключается в определении количества зерен на $0,5 \text{ мм}^2$ поверхности шлифа и расчете средней площади и среднего диаметра зерен. Таким образом, для более точного подсчета количества зерен, при предварительной обработке важно сохранить границы зерен. После бинаризации зерна представляются белым цветом на изображении, а их границы черным. Далее выполняется выделение связанных областей линейным алгоритмом, каждая из которых закрашивается своим определенным цветом.

Пусть c - цвет окраски, $I(y, x)$ – цвет пикселя изображения, M - ширина изображения, N - высота, b - флаг смены цвета

Алгоритм требует выполнения следующих шагов:

1. Задается произвольное значение c ;
2. Для каждой строки изображения, включающей M пикселей:
 - а) если $I(i, j) == 0$, $b = 1$, т.е. признак для изменения цвета;
 - б) если $I(i, j) == 255$ и $b == 1$ значение c меняется, а текущему пикселю присваивается текущее значение c . Если проверялся последний пиксель строки, то $b = 1$.
3. Для каждой строки изображения, включающей M пикселей сравнивается цвет текущего пикселя $I(i, j)$ с цветом вышестоящего $I(i-1, j)$:
 - а) если $I(i, j) == 0$ или $I(i-1, j) == 0$, то $j = j + 1$;
 - б) если $I(i, j) <> 0$ и $I(i-1, j) <> 0$ и $I(i, j) == I(i-1, j)$, то $j = j + 1$;
 - в) если $I(i, j) <> 0$ и $I(i-1, j) <> 0$ и $I(i, j) <> I(i-1, j)$, то $I(i-1, j) == I(i, j)$.
4. если хотя бы один раз выполнялось условие 3в, то происходит возврат к шагу 3.

Далее выполняется подсчет количества пикселей для каждой связанной цветовой области для всего изображения и для выбранной области анализа и определяется число зерен. Определение количества зерен, входящих в исследуемую область выполняется с учетом частичного вхождения зерен.

4.1.3 Определение длин хорд

Метод измерения длин хорд основан на измерении линейных размеров отрезков, отсекаемых в зернах прямыми линиями, и применяется для определения величины зерна в разнотельной структуре. Таким образом, на микроснимке необходимо провести не менее трех прямых в произвольном направлении. Далее можно определить относительную долю зерен.

Для данного метода, при предварительной обработке, нежелательно применять операцию расширения границ, так как это изменит размер зерен и может внести погрешность в результат обработки.

Спецификой реализации данного метода является предоставление возможности пользователю задать количество исследуемых областей. В ручном режиме пользователь должен иметь возможность самостоятельно провести линии.

Таким образом, на вход алгоритма поступает бинарное изображение и набор линий, каждая задана координатами двух точек.

Пусть (x_1, y_1) - координаты начала линии, (x_2, y_2) - координаты конца линии, $I(y, x)$ - цвет пикселя изображения, M - ширина изображения, N - высота, I_{last} - хранит цвет последнего рассмотренного пикселя, C - количество найденных переходов цвета, C_{fil} - количество отфильтрованных точек перехода.

Для каждой пары точек требуется выполнить следующие шаги:

1. вычислить коэффициенты уравнения прямой и выполнить проход по пикселям изображения, лежащим на линии, с единичным шагом. При этом, для каждого такого пикселя сохраняются координаты и тип перехода - с черного на белый, либо с белого на черный;
2. выполнить фильтрацию точек по расстоянию:
 - а) если $i=0$, то рассматриваемая точка заносится в выходной набор: $Res(C_{fil}) = points(i)$, $C_{fil} = C_{fil} + 1$;
 - б) если $i \neq 0$, то вычисляется расстояние от текущей точки до предыдущей в списке: $L = ((points(i).y - points(i-1).y)^2 + (points(i).x - points(i-1).x)^2)^{1/2}$. Если $L > 7$, то: $Res(C_{fil}) = points(i)$, $C_{fil} = C_{fil} + 1$.
3. Выполнить фильтрацию шумовых переходов за счет анализа полученного набора точек на предыдущем шаге путем выделения пар начало - конец, несколько подряд идущих переходов отбрасываются. Таким образом, если линия будет проходить через пиксель с шумом, это не будет восприниматься как граница зерна.
4. Подсчитываются расстояния каждой из полученных хорд.

4.1.4 Подсчет пересечения границ

Метод подсчета пересечений границ зерен состоит в подсчете, пересеченных срезами прямой и определении среднего условного диаметра - в случае равноосных зерен или количества зерен на 1 мм^3 - в случае неравноосных зерен. Для данного метода, при предварительной обработке, так же нежелательно применять операцию расширения границ для исключения дополнительной погрешности к результату.

Таким образом, при реализации данного метода необходимы два типа вычислений, различие между ними будет на этапе выбора области для анализа, так как для неравноосных зерен требуется загрузить второе изображение шлифа. Перед анализом необходимо провести перпендикулярные линии на изображении шлифа. Далее требуется подсчитать количество пересеченных зерен, для этого может быть использован рассмотренный алгоритм нахождения и фильтрации границ зерен. После нахождения всех пересечений необходимо подсчитать общую длину пересеченных зерен и вычислить средний условный диаметр зерна, как общая длина, деленная на их количество.

4.1.5 Определение содержания углерода

Для определения процентного содержания углерода в стали необходимо определить величины площадей, занимаемых на микрошлифе ферритом и перлитом.

Содержание углерода определяется по формуле:

$$C = P * 0,8/100 ,$$

где: P - процент перлитной составляющей.

Перлит на микроструктуре выглядит как темные зерна, соответственно по соотношению светлых и темных зерен можно определить процент перлита.

Входным для данной итерации является бинарное изображение. Для него выполняется подсчет числа черных пикселей, на основе которого и вычисляется процентное содержание перлита.

4.2 Программная реализация алгоритмов обработки

На основе рассмотренных методик и методов цифровой обработки изображений разработано программное обеспечение для анализа изображений микроструктуры металлов [72], которое имеет функции:

- работа с изображениями, полученными с USB камеры: получение изображения с камеры, вывод в интерфейсе программы в реальном времени с возможностью захвата кадра и последующей его обработкой;

- пакетная обработка изображений заданным методом: возможность загрузки группы снимков для обработки однотипным алгоритмом;
- предварительная обработка изображения:
 - применение медианной фильтрации, с возможностью выбора размера фильтра (3×3, 5×5, 7×7);
 - применение высокочастотной фильтрации, с возможностью выбора размера фильтра (3×3, 5×5, 7×7);
 - применение оператора Канны, с возможностью настроить параметр сглаживания, а также пороги бинаризации;
 - применение бинаризации, с возможностью настройки порога белого, размера окна, коэффициента порога;
 - применение операций математической морфологии (расширение, отмыкание и замыкание), с возможностью выбора размера структурирующего элемента типа квадрат со стороной от 3 до 5 для расширения и 3 до 19 для отмыкания и замыкания.
- предпросмотр при предварительной обработке изображения.
- сохранение изображений в указанный пользователем файл формата PNG на этапах предварительной обработки.
- реализация металлографических методов анализа в соответствии с ГОСТ 5639-82:
 - метод подсчета зерен, используемый для равноосных зерен;
 - метод подсчета пересечения границ зерен, применяемый как для равноосных зерен, так и для неравноосных;
 - метод измерения хорд, применяемый для определения величины зерна в разнородной структуре.
- определение содержания углерода в сталях.
- формирование отчета в виде документа MS Word или PDF

Диаграмма вариантов использования для данного программного продукта представлена на рисунке 31.

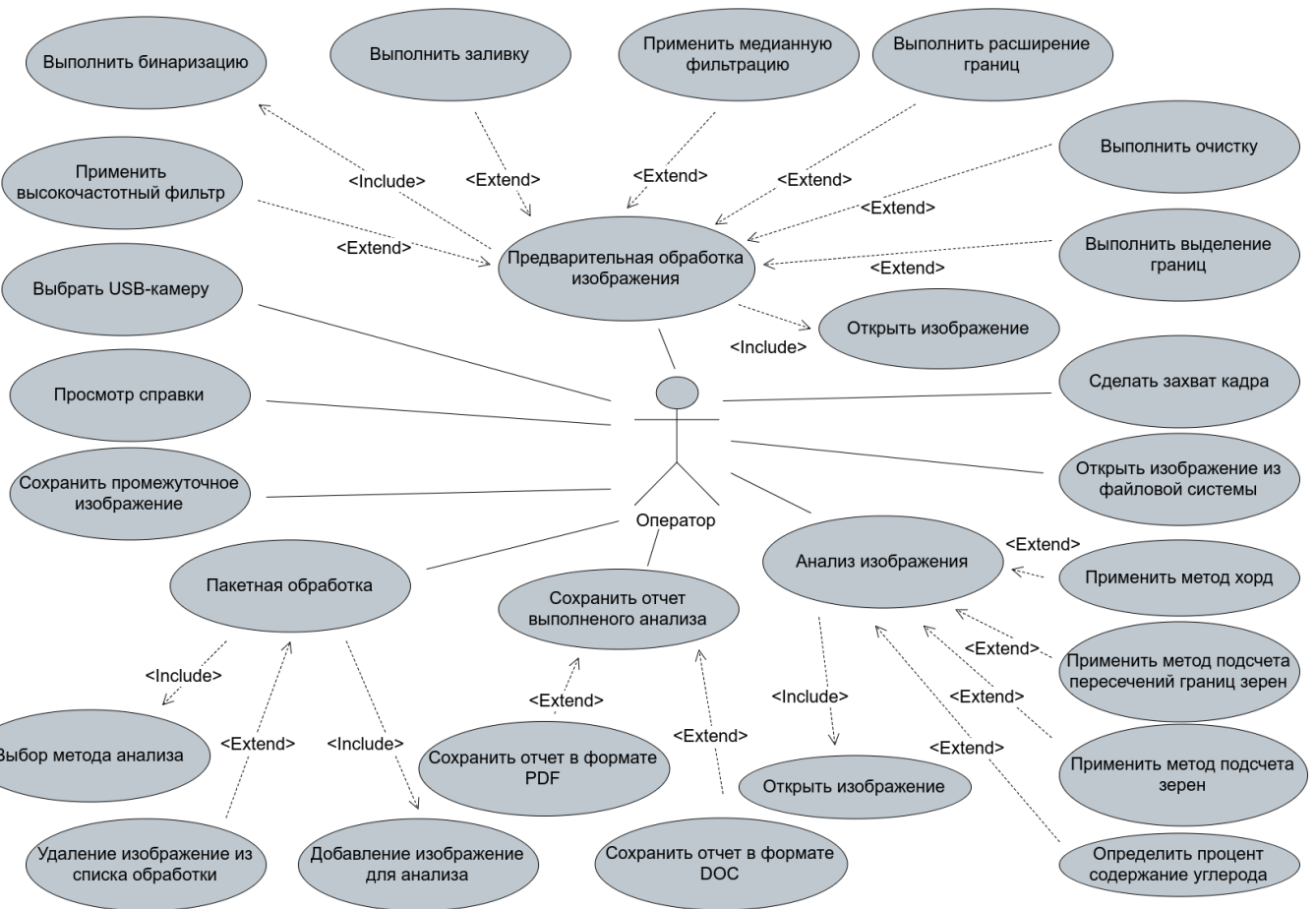


Рисунок 31 – Диаграмма вариантов использования

После запуска программы, пользователь должен выбрать сохраненное изображение либо получить его с USB-камеры и далее можно проводить предварительную обработку и необходимый анализ. Пример результата работы программы представлен на рисунке 32.

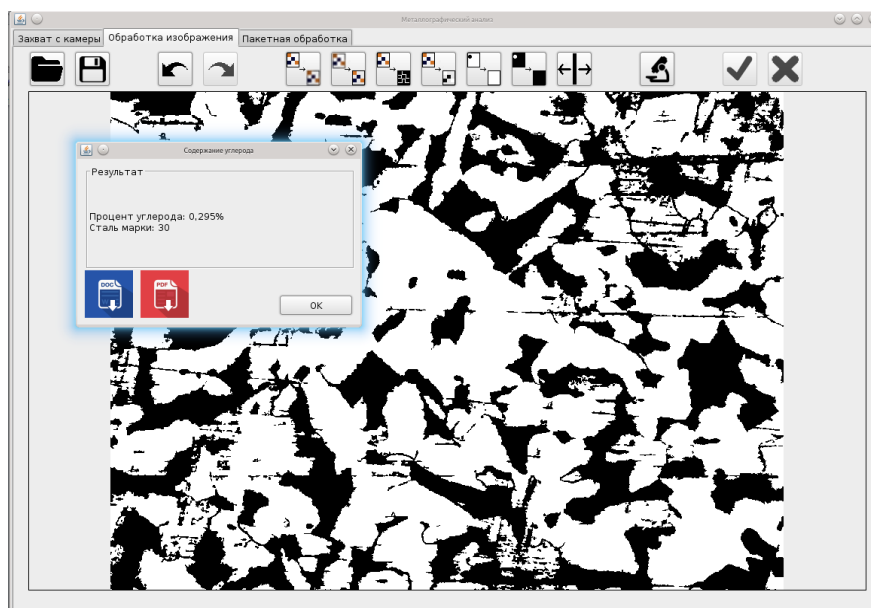


Рисунок 32– Пример результат обработки

Для исходного изображения, представленного на рисунке 33, показан результата его анализа методом подсчет зерен (рисунок 34) и методом хорд (рисунок 35).

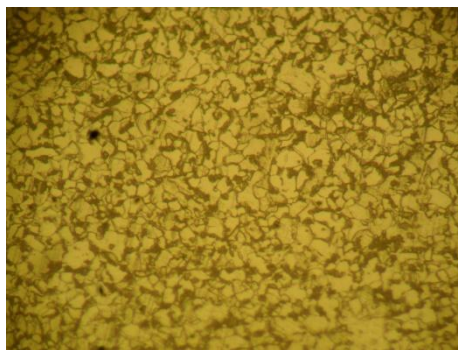


Рисунок 33 – Изображение неравноосных зерен

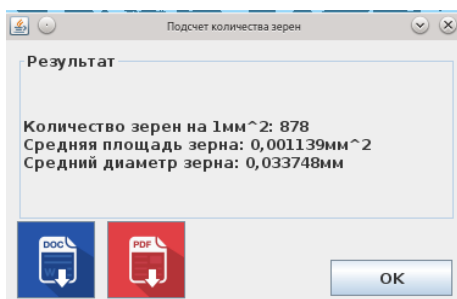


Рисунок 34 – Результат анализа методом подсчета зерен

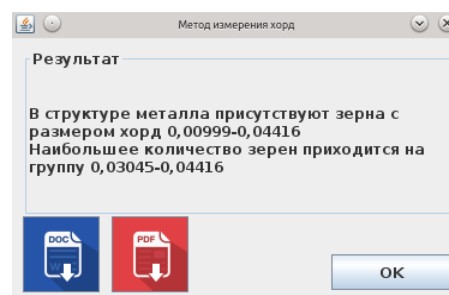


Рисунок 35 – Результат анализа методом измерения хорд

Данный программный продукт внедрен в контрольно-испытательную лабораторию учреждения образования «Полоцкий государственный университет», акт о внедрении от 19.06.2017г.

ЗАКЛЮЧЕНИЕ

В ходе исследований за 2016-2018гг. получены следующие основные результаты:

- предложен алгоритм классификации изображений парковочных мест на основе вычисления гистограмм ориентированных градиентов и метода опорных векторов. Признаки изображения парковочного места формируются на основе гистограмм ориентированных градиентов путем выполнения следующих шагов: вычисление горизонтальных и вертикальных градиентов для исходного изображения парковочного места, нахождение модуля вектора градиента и ориентаций, накопление мощностей градиентов в соответствии с ориентациями в ячейках, объединение ячеек в блоки, вычисление второй нормы ориентаций ячеек блока и нормализация ориентаций ячеек блока. Классификация дескрипторов парковочного места выполняется с использованием метода опорных векторов с функцией ядра на основе пересечения гистограмм. Определены наиболее эффективные значения параметров дескриптора: размер изображения парковочного места - 64×72 , размер ячейки - 8×8 , количество ориентаций в ячейке - 16. При этом вероятность правильной классификации 0,997;

- разработан алгоритм обнаружения дыма лесного пожара на динамических изображениях, который требует выполнения следующих основных этапов: детектирование медленно движущихся областей, цветовую сегментацию обнаруженных областей, вычисление ковариационных признаков сегментированных областей, классификацию полученных ковариационных признаков. Отличительной особенностью алгоритма является также использование цветового пространства HSV, что позволяет уменьшить число ложных обнаружений на первом этапе алгоритма. Малый размер пространственно-временного блока при вычислении ковариационных признаков позволяет обнаруживать области дыма незначительного размера, что очень важно для раннего обнаружения лесного пожар.

- разработан алгоритм обнаружения объектов на видеопоследовательностях с разрешением 4K на основе сверточной нейронной сети YOLO. При этом требуется выполнить: разбиение исходного кадра на перекрывающиеся ячейки; классификацию ячеек с использованием нейронной сети YOLO; выделить регионы интереса с центрами в соседних ячейках, имеющих общую пересекающуюся область; объединить регионы интереса при перекрытии объектов однотипных классов более заданного значения; удалить регион в случае его вхождения в новый большего размера. Предложенная методика позволяет повысить точность обнаружения объектов небольших размеров на изображениях большого разрешения, что достигается за счет исключения масштабирования исходного изображения к размерам первого слоя нейронной сети и обеспечить удовлетворительные вычислительные затраты за счет использования блочной обработки входного кадра. Алгоритм направлен на решение важной практической задачи – разработку

современных систем обнаружения произвольных объектов на видеопоследовательностях, получаемых с камер большого разрешения. Предложенный алгоритм позволяет повысить точность обнаружения объектов на 35% по сравнению с моделью YOLO. Время обработки видеокadra разрешения 4k с использованием компьютера с параметрами CPU i7 4.3 ГГц, ОЗУ 32 Гб, GPU Nvidia GeForce GTX 1070 составляет 2,64с.;

- разработано программное обеспечение для анализа структуры материала при металлографическом анализе в соответствии с ГОСТ 5639-82, которое включает реализацию метода подсчета зерен, метод подсчета пересечений границ зерен, метода хорд, применяемых для определения величины зерна в разнотернистой структуре и методики определения содержания углерода в сталях;

Выполненные исследования проводились в соответствии с планом работ по теме на 2016-2018 год. План работ выполнен в полном объеме.

Результаты НИР используются в контрольно-испытательной лаборатории учреждения образования «Полоцкий государственный университет», а также при организации учебного процесса по курсам «Цифровая обработка сигналов и изображений», «Цифровая обработка изображений» и при проведении научно-исследовательской работы студентов первой ступени высшего образования специальностей 1-400201 «Вычислительные машины, системы и сети», 1-400101

Программное обеспечение информационных технологий» и второй ступени высшего образования специальности 1-408004 «Математическое моделирование, численные методы и комплекс программ».

Разработанные алгоритмы могут использоваться при разработке современных систем обнаружения произвольных объектов на видеопоследовательностях, в том числе получаемых с камер большого разрешения, для обнаружения и классификации объектов в видеосистемах управления автомобильными парковками, системах мониторинга лесных пожаров.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Васин, Н.Н. Система технического зрения для контроля состояния железнодорожного пути / Н.Н. Васин, Р.Р. Дязитдинов // Компьютерная оптика. – 2016. – Т. 40. - №3. – С. 410-415
2. Brovko, N. Smoke detection algorithm for intelligent video surveillance system / N. Brovko, R. Bogush, S. Ablameyko // Computer Science Journal of Moldova. – 2013. – Vol.21. - №1. – P. 142-156
3. Ngan, K. Video Segmentation and Its Applications / K. Ngan, H. Li // Springer. – 2011. – 164p.
4. Luming, F. A study of the key technology of forest fire prevention based on a cooperation of video monitor and GIS/ F. Luming, X. Aijun, and T. Lihua // Proc. of the 10th Int. Conf. on Natural Computation. –2008.-Vol. 5.-p. 391–396
5. Idris, M.Y. Car park system: A review of smart parking system and its technology / M.Y.I. Idris, Y.Y. Leng, E.M. Tamil, N.M. Noor, Z. Razak // Information Technology Journal. – 2009. – Vol. 8(2). – P. 101–113
6. Managing and Monitoring of a Parking Lot by a Video Camera [Electronic resource]– Mode of access: [http:// bib.irb.hr/datoteka/509872.Automatizacija_u_prometu_2010.pdf](http://bib.irb.hr/datoteka/509872.Automatizacija_u_prometu_2010.pdf)
7. R. Yusnita, Fariza Norbaya, and Norazwinawati Basharuddin Intelligent Parking Space Detection System Based on Image Processing. International Journal of Innovation, Management and Technology.2012. - Vol. 3. - N3. – p. 232-235
8. Tschentscher M., Neuhausen M. Video-based parking space detection // Proceedings of the Forum Bauinformatik. -2012. p. 159-166
9. True, N. Vacant parking space detection in static images /N.True // [Electronic resource]–2007.-Mode of access: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.1444&rep=rep1&type=pdf> - Date of access: 08.06.16
10. Bong, D.B.L. Car-Park Occupancy Information System / D.B.L. Bong, K.C. Ting, N. Rajae // Proc. of the 3rd Real-Time Technology and applications symposium. – 2006. - p. 65-70
11. Sastre, R. Computer algebra algorithms applied to computer vision in a parking management system /R. Sastre, P. Gil Jimenez, F. Acevedo, S. Maldonado Bascon// Proc. of the IEEE Int. Symposium on Industrial Electronics. -2007. - p. 1675–1680
12. Almeida, P.R. Parking space detection using textural descriptors / P.R. de Almeida, L.S. Oliveira, A.S.B.Jr., E.J.S.Jr., A.L. Koerich // Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics. - 2013. - p. 3603–3608
13. Huang, C.-C. Vacant parking space detection based on plane-based bayesian hierarchical framework / C.-C. Huang, Y.-S Tai, S.-J Wang // IEEE Transactions on Circuits and Systems for Video Technology. - 2013. –Vol.23 - №9. - p. 1598–1610
14. Wang, X. Parking lot analysis and visualization from aerial image / X. Wang, A. Hanson // Proc. 4th IEEE Workshop on Applications of Computer Vision. 1998. - p.36-41
15. Bong, D.B.L. Integrated approach in the design of car park occupancy information system / D.B.L. Bong, K.C. Ting, K.C. Lai // IAENG International Journal of Computer Science. – 2008. - № 35.-p.1–8

16. Seo, Y.-W. Utilizing prior information to enhance selfsupervised aerial image analysis for extracting parking lot structures /Y.-W. Seo, C. Urmson // Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. - 2009. - p. 339–344
17. Ярошевич, П.В. Сегментация парковочных мест на изображениях эстакад /П.В. Ярошевич, Р.П. Богущ// Вестник Полоцкого государственного университета, серия С. Фундаментальные науки. - 2016. - №12 – с. 6-12
18. Otsu, N. A threshold selection method from gray-level histogram // IEEE Transactions on Systems, Man and Cybernetics – 1979. -№9. – pp. 62-66
19. Dalal, N. Histograms of oriented gradients for human detection / N. Dalal, B. Triggs // Proc of the. IEEE Conference on Computer Vision and Pattern Recognition. – 2005. – vol. 1. – P. 886-893
20. Вапник В.Н. Теория распознавания образов / В.Н. Вапник, А.Я. Червонекис – М.: Издательство «Наука», 1974. – 416 с.
21. PKLot - A robust dataset for parking lot classification - [Электронный ресурс] - 2015. URL: <http://www.inf.ufpr.br/vri/databases/PKLot.tar.gz> (дата обращения 5.12.2015).
22. Huang, C.-C. A surface-based vacant space detection for an intelligent parking lot / Ching-Chun Huang, Yu-Shu Dai, Sheng-Juh Wang // Proc. of 12th Int. Conf. on ITS Telecommunications. – 2012. – p. 284-288.
23. Huang, C.-C. A multiclass boosting approach for integrating weak classifiers in parking space detection / Ching-Chun Huang, Hoang Tran Vu, Yi-Ren Chen // Proc. of the IEEE Int. Conf. on Consumer Electronics, – 2015. – p. 314-315.
24. Fusek, R. AdaBoost for parking lot occupation detection / Fusek R, Mozdren K, Surkala M, Sojka E //Proc. of the 8th Int. Conf. on Computer Recognition Systems. – 2013. – P. 681-690.
25. Tschentscher, M. Scalable real-time parking lot classification an evaluation of image features and supervides learning algorithms / M. Tschentscher, M. Neuhausen, C. Koch, M. Konig, J. Salmen, M. Schlipf // Proc. of the Int. Joint Conf. on Neural Networks. – 2015. – p. 1-8.
26. Baroffio, L. A visual sensor network for parking lot occupancy detection in Smart Cities / L. Baroffio, L. Bondi, M. Cesana, A. Redondi, M. Tagliasacchi // Proc. of the IEEE 2nd World Forum on Internet of Things (WF-IoT). – 2015. – p. 745-750
27. Кудрин, А.Ю. Современные методы обнаружения и мониторинга лесных пожаров / А.Ю. Кудрин, А.И. Запорожец, Ю.В. Подрезов // Технологии гражданской безопасности. - 2006.- №3. – С. 66-67
28. Narwade, A Result analysis of smoke detection in video for early warnings using static and dynamic features/ A. Narwade, A. Chakkarwar// International Journal of Computer Applications Understanding. – 2014.-Vol. 95. - №24. – p.12-18
29. Xiong, Z. Video-based Smoke Detection: Possibilities, Techniques, and Challenges / Z. Xiong, R. Caballero, H. Wang, A. Finn // Proc. of the Supression and Defection Research and Applications Conference. - 2007. p. 157–164.
30. Gomes-Rodriguez, F. Smoke monitoring and measurement using image processing. Application to Forest Fires/ F. Gomes-Rodriguez, B. Arrue, A. Ollero // Proc. SPIE 5094, Automatic Target Recognition.–2003.– p. 404-411

31. Genovese, A. Wildfire smoke detection using computational intelligence/ A. Genovese, R. D. Labati, V. Piuri, F. Ccotti// Proc of the IEEE Int. Conf. on Computational Intelligence for Measurement Systems and Applications.- 2011. - p. 1-6
32. Cetin, A. Wildfire detection using LMS based active learning / A. Cetin, B. Toreyin // Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. – 2009. - p. 1461-1464, 2009,
33. Kim, D. Smoke detection in video/ D. Kim. Y. Wang/ /Proc. of the World Congress on Computer Science and Information Engineering. - 2009. - p. 759-763
34. Vicente, J. An image processing technique for automatically detecting forest fire/ J. Vicente, P. Guillemant// International Journal of Thermal Sciences. 2002. – Vol.41. - №12. - p. 1113-1120.
35. Gunay, O. Entropy functional based online adaptive decision fusion framework with application to wildfire detection in video/ O. Gunay, B. Ugur Toreyin, A. Enis Cetin// IEEE Transactions on Image Processing. - 2012 – Vol.21. - №5. - p. 2853 - 2865
36. Habiboglu, Y. Real-time wildfire detection using correlation descriptors / Y. H. Habiboglu, O. Gunay // Proc. of the 19th European Signal Processing Conference. - 2011. - p. 894 – 898
37. Toreyin, B. Wavelet based real-time smoke detection in video/B. Toreyin, Y. Dedeoglu and A. E. Cetin // Proc. of the Signal Processing Conference. -2005.- p.1-4.
38. Toreyin, B. Contour based smoke detection in video using wavelets/B. Toreyin, Y. Dedeoglu and A. E. Cetin // Proc. of the 14th European Signal Processing Conference. – 2006. - p.1-5
39. Piccinini, P. Reliable smoke detection in the domains of image energy and color/P. Piccinini, S. Calderara, R. Cucchiara//Proc. of the 15th IEEE Int. Conf. on Image Processing. -2008. - p.1376-1379
40. Calderara, S. Smoke detection in video surveillance: A MoG model in the wavelet domain/ S. Calderara, P. Piccinini and R. Cucchiara // Proc. of the 6th International Conference on Computer Vision Systems. – 2008. - p.119-128
41. Yuan, F. A fast accumulative motion orientation model based on integral image for video smoke detection/F. Yuan//Pattern Recognition Letters.- 2008 - Vol.29. - p.925-932
42. Morerio, P. Early fire and smoke detection based on colour features and motion analysis/P. Morerio, L. Marcenaro, C. Regazzoni, G. Gera//Proc. of the IEEE Int. Conf. on Image Processing - . 2012. – p. 1041-1044
43. Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction / Z. Zivkovic //Proc. of the 17th Int. Conf. on Pattern Recognition. - 2004. – Vol. 2. – p. 28 – 31
44. Krizhevsky, A. ImageNet classification with deep convolutional neural networks/ A. Krizhevsky, I. Sutskever, G.E. Hinton //Proc. Of 26th Annual Conference on Neural Information Processing, Lake Tahoe, Nevada, United States. – 2012. – p. 1106-1114
45. Han, X. Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification/ X. Han, Y. Zhong, L. Cao, L. Zhang //Remote Sensing. – 2017. - №9 - T.8-p.2-22

46. R-CNN for Object Detection [Electronic resource]– Mode of access: https://courses.cs.washington.edu/courses/cse590v/14au/cse590v_wk1_rcnn.pdf- Date of access: 09.03.17
47. Girshick, R. Fast R-CNN/R. Girshick // Proc. of the IEEE International Conference on Computer Vision. - 2015.- p.1440-1448
48. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [Electronic resource]– Mode of access: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>- Date of access: 10.03.17
49. Is Faster R-CNN Doing Well for Pedestrian Detection[Electronic resource]– Mode of access: <https://arxiv.org/pdf/1607.07032.pdf>- Date of access: 10.03.17
50. Going Deeper with Convolutions [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1409.4842.pdf>- Date of access: 10.03.17
51. Erhan, D. Scalable object detection using deep neural networks/ D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov // Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA. – 2014. – p. 2155-2162
52. Rethinking the Inception Architecture for Computer Vision [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1512.00567.pdf>- Date of access: 10.03.17
53. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1502.03167.pdf>- Date of access: 10.03.17
54. He, K. Deep Residual Learning for Image Recognition/ K. He, X. Zhang, S. Ren, J. Sun //Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA. – 2016. – p. 770 - 778
55. Szegedy, C. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning/ C. Szegedy, S. Ioffe, V. Vanhoucke // Proc. of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), 2017. – 4278-4284
56. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1603.04467.pdf>- Date of access: 22.09.17
57. Redmon, J. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi // Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA. - 2016. –p.779-788
58. YOLO9000 Better, Faster, Stronger[Electronic resource]– Mode of access: <https://arxiv.org/pdf/1612.08242.pdf>- Date of access: 15.06.17
59. Han, W. Seq-NMS for Video Object Detection/ W. Han and ather // [Electronic resource]– Mode of access: <https://openreview.net/pdf?id=2xwPBxZoQTpKBZvXtQng> - Date of access: 15.06.18
60. Zeiler M. D., Visualizing and Understanding Convolutional Networks/ M. D. Zeiler, R. Fergus // ECCV. – 2014. - Vol. 8689. - p. 818-833
61. Simonyan K., Very Deep Convolutional Networks for Large-Scale Image Recognition/ K. Simonyan, A. Zisserman // [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1409.1556.pdf>- Date of access: 15.06.18

62. T-CNN Tubelets with Convolutional Neural Networks for Object Detection from Videos [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1604.02532.pdf> - Date of access: 15.06.18
63. Hou, R. Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos/ R. Hou, C. Chen, M. Shah // [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1703.10664.pdf> - Date of access: 15.06.18
64. Zhu, X. Deep Feature Flow for Video Recognition/ X. Zhu, Y. Xiong, J.Dai, L. Yuan, Y. Wei // Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA. - 2017. - p. 4141 - 4150
65. Flownet: Learning optical flow with convolutional networks [Electronic resource]– Mode of access: https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Dosovitskiy_FlowNet_Learning_Optical_ICCV_2015_paper.pdf- Date of access: 15.06.18
66. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs [Electronic resource]– Mode of access: <https://arxiv.org/pdf/1606.00915.pdf>] - Date of access: 15.06.18
67. Flow-Guided Feature Aggregation for Video Object Detection [Electronic resource] – Mode of access: <https://arxiv.org/pdf/1703.10025.pdf>- Date of access: 15.06.18
68. The PASCAL Visual Object Classes Homepage [Electronic resource]– Mode of access: <http://host.robots.ox.ac.uk/pascal/VOC/>- Date of access: 15.06.17
69. Vorobjov, D. An effective object detection algorithm for high resolution video by using convolutional neural network/D. Vorobjov, I. Zakharava, R. Bohush, S. Ablameyko// Proc. of the 15th Int. Symposium on Neural Networks (ISNN 2018), June 25-28 2018, Minsk. – 2018. – p.503-508
70. Стали и сплавы. Методы выявления и определения величины зерна: ГОСТ 5639-82. – Введ. 01.01.83. – М.: Государственный комитет СССР по стандартам: Министерство черной металлургии СССР, 2003. – 8 с.
71. Анисович А.Г. Микроструктуры черных и цветных металлов / Анисович А.Г., Андрушевич А.А. – Мн.: Беларуская навука, 2015. – 131с.
72. Богуш, Р.П. Программное обеспечение для анализа изображений микроструктуры металлов / Богуш Р.П., Туркова В.В., Глухов Д.О. // Информатика: проблемы, методология, технологии. Материалы XVIII Междун. конф., 8-9 февраля 2018 г. – Воронеж: Воронежский гос. ун-т, 2018. –Т.4 - с.69-73

ПРИЛОЖЕНИЕ
(информационное)

Перечень публикаций по заданию «Информатика и космос» 1.7.05

НАУЧНЫЕ СТАТЬИ В ЖУРНАЛАХ И СБОРНИКАХ НАУЧНЫХ ТРУДОВ

1. Ярошевич, П.В. Сегментация парковочных мест на изображениях автостоянок /П.В. Ярошевич, Р.П. Богуш// Вестник Полоцкого государственного университета, серия С. Фундаментальные науки. - 2016. - №12 – с. 6-12
2. Ярошевич, П.В. Анализ алгоритмов обработки динамических изображений для систем мониторинга свободных парковочных мест [Электронный ресурс] / П.В. Ярошевич, Р.П. Богуш // Электронный сборник трудов молодых специалистов Полоцкий государственный университет. Сер. Прикладные науки. –2016ю - Выпуск 15(85). – С. 71-74. Электронный оптический диск-1 диск.
3. Ярошевич, П.В. Классификация изображений парковочных мест автостоянки с использованием гистограмм ориентированных градиентов и метода опорных векторов/ П.В. Ярошевич, Р.П. Богуш // Техническое зрение. – 2016. – Вып.1. – С. 24-28
4. Ye, Sh. An Effective Algorithm to Detect Both Smoke and Flame Using Color and Wavelet Analysis/ Sh. Ye, Zh. Bai, H. Chen, R. Bohush and S. Ablameyko // Pattern Recognition and Image Analysis, 2017, Vol. 27, No. 1, pp. 131–138
5. Ярошевич, П.В. Алгоритм классификации изображений парковочных мест автостоянки на основе гистограмм ориентированных градиентов и метода опорных векторов / П.В. Ярошевич, Р.П. Богуш // Компьютерная оптика. – 2017. – Т. 41, № 1. – С. 110-117
6. Захарова, И.Ю. Анализ архитектур сверточных нейронных сетей для обнаружения и классификации объектов на видеопоследовательностях большого разрешения/ И.Ю. Захарова, Д.Э. Воробьев // Труды молодых специалистов ПГУ. Промышленность– 2018. – Вып.25(95) . – с.94-97
7. Захарова, И.Ю. Блочный алгоритм обнаружения и классификации объектов с использованием сверточной нейронной сети для видеопоследовательностей большого разрешения/ И.Ю. Захарова // Труды молодых специалистовПГУ. Промышленность – 2018. – Вып.25(95) . – с.91-93

НАУЧНЫЕ СТАТЬИ В МАТЕРИАЛАХ КОНФЕРЕНЦИЙ

8. Авсеенко, Д.В. Алгоритм детектирования дыма для обнаружения лесных пожаров по видеоданным / Д.В. Авсеенко, Р.П. Богуш // Информатика: проблемы, методология, технологии. Материалы XVI Междун. конф., 11-12 февраля 2016 г. – Воронеж: Воронежский гос. ун-т, 2015. –Т.2 - с. 24-27
9. Ярошевич, П.В. Обработка динамических изображений в системах видеомониторинга для определения свободных парковочных мест / П.В. Ярошевич, Р.П. Богуш // Информатика: проблемы, методология, технологии. Материалы XVI

Междун. конф., 11-12 февраля 2016 г. – Воронеж: Воронежский гос. ун-т, 2015. –Т.2 с. 324-327

10. Kurilovich, A. Content recognition of bank cards in ios mobile devices / A. Kurilovich, R. Bohush // European and National Dimension in Research: materials of the III Junior Researchers' Conf., Novopolotsk, April 27-28, 2016 / Polotsk State University.-Novopolotsk, 2016 – V.3 - p.220-223

11. Vorobyov, D. Object localization and classification based on convolutional neural networks/D. Vorobyov, R. Bohush //European and National Dimension In Research. Technology: Electronic collected materials of IX Junior Researchers' Conference, Novopolotsk, April 26–27, 2017 / Polotsk State University; ed. D. Lazouski [et al.]. – Novopolotsk, 2017. –1 CD-ROM. – p. 323-326

12. Богуш, Р.П. Моделирование алгоритма раннего обнаружения лесного пожара на основе анализа пространственных и временных признаков видеопоследовательностей / Р.П.Богуш, А.В.Авсеенко, Д.О.Глухов// Интеллектуальные и информационные технологии в формировании цифрового общества. Сборник научных статей международной научной конференции. 20–21 июня 2017 г. Санкт- Петербург. Conference of St.-Petersburg State University of Economics/Под ред. проф. В.В. Трофимова, В.Ф. Минакова. – СПб.: Изд-во СПбГЭУ, 2017. – с. 50-54

13. Богуш, Р.П. Программное обеспечение для анализа изображений микроструктуры металлов / Богуш Р.П., Туркова В.В., Глухов Д.О. // Информатика: проблемы, методология, технологии. Материалы XVIII Междун. конф., 8-9 февраля 2018 г. – Воронеж: Воронежский гос. ун-т, 2018. –Т.4 - с.69-73

14. Kurilovich, A. Software for scan bank cards in iOS mobile devices / A. Kurilovich, R. Bohush // European and National Dimension in Research: materials of the X Junior Researchers' Conf., Novopolotsk, May 10-11, 2018 / Polotsk State University.-Novopolotsk, 2018 – p.272-276

15. Богуш, Р.П. Структура интеллектуальной видеосистемы управления парковками/ Р.П. Богуш, С.В. Абламейко, П.В. Ярошевич // Информационно-коммуникационные технологии: достижения, проблемы, инновации (ИКТ-2018) [Электронный ресурс]: электронный сборник статей I международной научно-практической конференции, посвященной 50-летию Полоцкого государственного университета, Новополоцк, 14–15 июня 2018 г. / Полоцкий государственный университет. – Новополоцк, 2018, стр. 254-256

16. Vorobjov, D. An effective object detection algorithm for high resolution video by using convolutional neural network/D. Vorobjov, I. Zakharava, R. Bohush, S. Ablameyko// Proc. of the 15th Int. Symposium on Neural Networks (ISNN 2018), June 25-28 2018, Minsk. – 2018. – p.503-508

17. Bohush,R. Parking spaces images localization and classification for intelligent parking management system /R.Bohush, P. Yarashevich, S. Ablameyko // Міжнародна

науково-технічна конференція «Комп'ютерна графіка та розпізнавання зображень» (Вінниця, 2018): збірник доповідей. – 2018. – Т. 2. – с. 26-30

ТЕЗИСЫ ДОКЛАДОВ

18. Ярошевич, П.В. Классификация парковочных мест автостоянки с использованием гистограмм ориентированных градиентов и метода опорных векторов / П.В. Ярошевич, Р.П. Богуш // Техническое зрение в системах управления-2015. Тезисы докладов 7 науч.-техн. конф., 15–17 марта 2016г. – Москва: Институт космических исследований РАН, 2016. – с.72-73

19. Захарова, И.Ю. Современные архитектуры сверточных нейронных сетей для обнаружения и классификации объектов на видеопоследовательностях / И.Ю. Захарова, Р.П. Богуш // Техническое зрение в системах управления-2018. Тезисы докладов 9 науч.-техн. конф., 13–15 марта 2018г. – Москва: Институт космических исследований РАН, 2018. – с.51-52

ПРИЛОЖЕНИЕ Б
(информационное)

Список конференций, на которых докладывались результаты исследований

1. XVI Международная конференция «Информатика: проблемы, методология, технологии», 11-12 февраля 2016 г. Воронежский государственный университет
2. Седьмая научно-техническая конференция «Техническое зрение в системах управления-2016», 15–17 марта 2016г., Институт космических исследований РАН, г. Москва
3. VIII Junior Researchers' Conf. European and National Dimension in Research, Novopolotsk, April 27-28, 2016, Polotsk State University
4. IX Junior Researchers' Conf. European and National Dimension in Research, Novopolotsk, April 26-27, 2017, Polotsk State University
5. Международная научно-практическая конференция «Интеллектуальные и информационные технологии в формировании цифрового общества», 19-20 июня 2017г. Санкт-Петербургский государственный экономический университет, г. Санкт-Петербург
6. XVIII Международная конференция «Информатика: проблемы, методология, технологии», 8-9 февраля 2018 г. Воронежский государственный университет
7. Девятая научно-техническая конференция «Техническое зрение в системах управления-2018», 13–15 марта 2018г., Институт космических исследований РАН, г. Москва
8. X Junior Researchers' Conf. European and National Dimension in Research, Novopolotsk, April May 10-11, 2018, Polotsk State University
9. Международная научно-практическая конференция «Информационно-коммуникационные технологии: достижения, проблемы, инновации», 14–15 июня 2018 г., Полоцкий государственный университет
10. International Symposium on Neural Networks, 25-28 June 2018, Belarusian State University
11. Міжнародна науково-технічна конференція «Комп'ютерна графіка та розпізнавання зображень» 4 грудня 2018, Вінницький національний технічний університет, Україна

ПРИЛОЖЕНИЕ В
(информационное)

Акты внедрения результатов НИР