

Инженерная школа информационных технологий и робототехники  
 Направление подготовки: Информатика и вычислительная техника  
 Отделение информационных технологий

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
<b>Разработка алгоритмов классификации объектов дорожного движения на изображениях и видеопоследовательностях</b>

УДК 004.424.5.021:004.932:656.1

Студент

Группа	ФИО	Подпись	Дата
8ВМ91	Кружков Денис Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Друки Алексей Алексеевич	К.Т.Н.		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Верховская Марина Витальевна	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Антоневич Ольга Анатольевна	К.Х.Н		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Кочегурова Елена Алексеевна	К.Т.Н.		

## ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код компетенции	Наименование компетенции
	<b>Универсальные компетенции</b>
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла
УК(У)-3	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
УК(У)-4	Способен применять современные коммуникативные технологии, в том числе на иностранном (-ых) языке (-ах), для академического и профессионального взаимодействия
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки
	<b>Общепрофессиональные компетенции</b>
ОПК(У)-1	Способен самостоятельно приобретать, развивать и применять математические, естественно-научные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
ОПК(У)-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач
ОПК(У)-3	Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ОПК(У)-4	Способен применять на практике новые научные принципы и методы исследований
ОПК(У)-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования
ОПК(У)-7	Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий
ОПК(У)-8	Способен осуществлять эффективное управление разработкой программных средств и проектов
	<b>Профессиональные компетенции</b>
ПК(У)-1	Способен разрабатывать и администрировать системы управления базами данных
ПК(У)-2	Способен проектировать сложные пользовательские интерфейсы

ПК(У)-3	Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов
ПК(У)-4	Способен осуществлять руководство разработкой комплексных проектов на всех стадиях и этапах выполнения работ
ПК(У)-5	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ВМ91	Кружков Денис Сергеевич

<b>Школа</b>	Инженерная школа информационных технологий и робототехники	<b>Отделение (НОЦ)</b>	Информационных технологий
<b>Уровень образования</b>	Магистратура	<b>Направление/специальность</b>	09.04.01 «Информатика и вычислительная техника»

Тема ВКР:

«Разработка алгоритмов классификации объектов дорожного движения на изображениях и видеопоследовательностях»	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<ul style="list-style-type: none"> <li>– Объектом исследования является алгоритмы классификации объектов дорожного движения по нескольким классам: автомобили, люди, дорожная разметка. Практическое применение алгоритмов возможно в беспилотных автомобилях, видеорегистраторах и системах видеонаблюдения.</li> <li>– Область применения: в беспилотных автомобилях, видеорегистраторах и системах видеонаблюдения.</li> <li>– Рабочая зона: помещение с персональными компьютерами.</li> </ul>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p><b>1. Правовые и организационные вопросы обеспечения безопасности:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<ul style="list-style-type: none"> <li>– ГОСТ 21889-76 Система "Человек-машина". Кресло человека-оператора. Общие эргономические требования. Обозначение.</li> <li>– ГОСТ 12.2.032-78 Система стандартов безопасности труда.</li> <li>– ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения</li> <li>– ГОСТ Р ИСО 9241-2-2009 Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT)</li> </ul>
<p><b>2. Производственная безопасность:</b></p> <p>2.1. Анализ выявленных вредных и опасных факторов</p>	<p>Рассмотрены вредные факторы:</p> <ul style="list-style-type: none"> <li>– отклонение показателей</li> </ul>

2.2. Обоснование мероприятий по снижению воздействия	микроклимата рабочей зоны; – повышенный уровень шума на рабочем месте; – отсутствие или недостаток естественного света; – недостаточная освещенность рабочей зоны; – монотонный режим работы. Рассмотрены опасные факторы: – повышенное значение напряжения в электрической цепи замыкание которой может произойти через тело человека.
3. Экологическая безопасность:	– Программное обеспечение не связано с производством, следовательно, влияние на окружающую среду минимально. – Рассмотрена утилизация бумажных отходов и неисправных комплектующих. – Решения по обеспечению экологической безопасности согласно нормативным документам.
4. Безопасность в чрезвычайных ситуациях:	– Возможные ЧС: Пожары и взрывы; – Наиболее типичная ЧС: Пожары и взрывы;

Дата выдачи задания для раздела по линейному графику	
--	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент (ООД, ШБИП)	Антоневич О. А.	к.б.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ91	Кружков Денис Сергеевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ВМ91	Кружков Денис Сергеевич

<b>Школа</b>	Инженерная школа информационных технологий и робототехники	<b>Отделение (НОЦ)</b>	Информационных технологий
<b>Уровень образования</b>	Магистратура	<b>Направление/специальность</b>	09.04.01 «Информатика и вычислительная техника»

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Человеческие ресурсы – 2 человека (руководитель и студент-дипломник). Тариф на электроэнергию – 6,59 руб. за 1 кВт·ч.
2. <i>Нормы и нормативы расходования ресурсов</i>	
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений во внебюджетные фонды – 30 %.

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Планирование и формирование бюджета научных исследований</i>	Необходимо составить календарный план НИИ и рассчитать затраты НИИ.
2. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Расчет интегральных показателей эффективности исследования, выбор наилучшего исполнения.

**Перечень графического материала:**

1. <i>Оценка конкурентоспособности технических решений</i>
2. <i>Матрица SWOT</i>
3. <i>График проведения и бюджет НИИ</i>
4. <i>Оценка ресурсной, финансовой и экономической эффективности НИИ</i>

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент	Верховская Марина Витальевна	Кандидат экономических наук		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ВМ91	Кружков Денис Сергеевич		

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

Инженерная школа информационных технологий и робототехники  
 Направление подготовки: Информатика и вычислительная техника  
 Отделение информационных технологий

УТВЕРЖДАЮ:  
 Зав. кафедрой

\_\_\_\_\_  
 (Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации
--------------------------

Студенту:

Группа	ФИО
8ВМ91	Кружков Денис Сергеевич

Тема работы:

«Разработка алгоритмов классификации объектов дорожного движения на изображениях и видеопоследовательностях»
--

Утверждена приказом директора (дата, номер)	
---	--

Срок сдачи студентом выполненной работы:	
--	--

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Набор кадров из видеопоследовательности, снятой с видеорегистратора.
---------------------------------	--

*(наименование объекта исследования; документы конференции и отчеты НИР; программное обеспечение).*

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> <li>1) Аналитический разбор литературы по теме “Распознавание объектов на видеопоследовательностях”.</li> <li>2) Постановка задачи.</li> <li>3) Изучение вопросов и тем, необходимых для понимания и использования алгоритма.</li> <li>4) Изучение и обоснование алгоритмов для решения задач классификации и детектирования</li> <li>5) Разработка системы распознавания объектов дорожного движения.</li> <li>6) Тестирование работоспособности алгоритма и программы.</li> <li>7) Применение программы для тестового примера.</li> <li>8) Заключение к работе</li> </ol>
--	--

<p><b>Перечень графического материала</b></p> <p><i>(с точным указанием обязательных чертежей)</i></p>	
--	--

<p><b>Консультанты по разделам выпускной квалификационной работы</b></p> <p><i>(с указанием разделов)</i></p>
---

Раздел	Консультант
1. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Верховская Марина Витальевна
2. Социальная ответственность	Антоневич Ольга Анатольевна

<p><b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b></p>	
--	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Друки Алексей Алексеевич	К.Т.Н		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ91	Кружков Денис Сергеевич		

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

Инженерная школа информационных технологий и робототехники  
 Направление подготовки Информатика и вычислительная техника  
 Отделение информационных технологий  
 Период выполнения \_\_\_\_\_ осенний / весенний семестр 2020/2021 учебного года

Форма представления работы:

магистерская диссертация

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы: \_\_\_\_\_

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
08.02.2021	Постановка целей и задач, получение исходных данных	
16.02.2021	Подбор и изучение материалов по тематике	
27.03.2021	Разработка календарного плана	
11.03.2021	Выбор методов для реализации	
27.03.2021	Разработка алгоритмов решения задачи	
10.04.2021	Реализация алгоритмов в виде программы	
30.04.2021	Проведение тестирования	
14.05.2021	Оформление расчётно-пояснительной записки	
21.05.2021	Оформление графического материала	
28.05.2021	Подведение итогов	

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Друки Алексей Алексеевич	к.т.н		

**СОГЛАСОВАНО:**

	ФИО	Ученая степень, звание	Подпись	Дата

## РЕФЕРАТ

Пояснительная записка выполнена на 130 листах машинописного текста, содержит 37 иллюстраций, 16 таблиц, 27 источников, 9 приложений.

Ключевые слова: классификатор, сегментация, компьютерное зрение, ограничения-неравенства, область допустимых значений,

Работа посвящена разработке и реализации алгоритмов классификации объектов дорожного движения на фото и видеопоследовательностях.

Объектом исследования является алгоритм детектирования и классификации объектов дорожного движения.

Целью работы является разработка системы обнаружения и распознавания объектов дорожного движения на видеопоследовательности.

В процессе работы были разработаны четыре модуля: модуль детектирования дорожной разметки, классификатор сигналов светофора, классификатор знаков дорожного движения, и классификатор участников дорожного движения. Алгоритм и программа были проверены на тестовом примере, подтверждающем корректность их работы.

Область применения: в беспилотных автомобилях, видеорегистраторах и системах видеонаблюдения.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ, НОРМАТИВНЫЕ ССЫЛКИ

В данной работе используются следующие термины с соответствующими определениями:

детектор – техническое средство, которое указывает на наличие определенного свойства объекта измерения при превышении порогового значения соответствующей величиной;

классификация – систематизированный перечень наименованных объектов, каждому из которых в соответствие дан уникальный код;

компьютерное зрение – это область искусственного интеллекта, связанная с анализом изображений и видео;

видеопоследовательность – совокупность, последовательность изображений на видеозаписи;

фрейм – фрагмент кино- или видеоряда, отдельное изображение или отрезок киноплёнки;

гистограмма направленных градиентов – дескрипторы особых точек, которые используются в компьютерном зрении и обработке изображений с целью распознавания объектов;

SVM (метод опорных векторов) – один из наиболее популярных методов обучения, который применяется для решения задач классификации и регрессии;

референс – вспомогательное изображение: рисунок или фотография;

интерфейс прикладного программирования – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

## ОГЛАВЛЕНИЕ

Введение .....	16
Обзор литературы .....	18
1. Анализ предметной области и средств разработки	19
1.1 OpenCV .....	19
1.2 Dlib .....	20
1.3 TensorFlow и Keras .....	23
1.4 ImageAI .....	25
2. Разработка алгоритмов	27
2.1 Детектирование полосы дорожной разметки .....	28
2.2 Детектирование светофоров и классификация их сигналов .....	30
2.3 Детектирование и классификация дорожных знаков .....	31
3. Программная реализация	34
3.1 Детектор дорожной разметки .....	36
3.1.1 Детектор границ Canny .....	36
3.1.2 Поиск области интереса .....	41
3.1.3 Вычисление и отрисовка направляющих .....	42
3.2 Модуль обнаружения и распознавания сигналов светофора .....	44
3.2.1 Обучение модели детектора .....	45
3.2.2 Реализация блока классификатора .....	47
3.3 Модуль распознавания дорожных знаков .....	48
3.3.1 Бинаризация изображения .....	48
3.3.2 Детектирование контуров .....	50
3.3.3 Референсная классификация .....	50

3.4 Обнаружение участников дорожного движения .....	53
3.4.1 Выбор архитектуры нейронной сети.....	54
3.4.2 Настройка модели детектора .....	54
3.4.3 Встраивание детектора в систему .....	55
4. Тестирование разработанных программных средств	57
4.1 Подготовка данных для тестирования .....	57
4.2 Тестирование системы .....	59
5. Социальная ответственность	63
Введение .....	63
5.1 Правовые и организационные вопросы обеспечения безопасности .....	63
5.1.2 Организационные мероприятия при компоновке рабочей зоны .....	64
5.2 Производственная безопасность .....	66
5.2.1 Анализ вредных и опасных факторов .....	66
5.2.2 Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов .....	71
5.3 Экологическая безопасность .....	72
5.4 Безопасность в чрезвычайных ситуациях .....	74
5.4.1 Анализ вероятных ЧС .....	74
5.4.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС .....	75
Выводы и рекомендации.....	77
6. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	78
6.1 Оценка коммерческого и инновационного потенциала научных исследований .....	78
6.1.1 Потенциальные потребители результатов исследований .....	78

6.1.2	Анализ конкурентных технических решений.....	79
6.1.3	SWOT-анализ.....	79
6.2	Планирование научно-исследовательских работ .....	81
6.2.1	Структура работ в рамках научного исследования .....	81
6.2.2	Определение трудоемкости выполнения работ .....	82
6.2.3	Разработка графика проведения научного исследования .....	83
6.3	Бюджет научно-технического исследования (НТИ) .....	85
6.3.1	Расчет затрат на специальное оборудование для научных (экспериментальных) работ .....	85
6.3.2	Основная заработная плата исполнителей темы .....	86
6.3.3	Дополнительная заработная плата .....	88
6.3.4	Отчисления во внебюджетные фонды (страховые отчисления).....	88
6.3.5	Прочие прямые затраты .....	89
6.3.6	Накладные расходы.....	90
6.3.7	Формирование бюджета затрат научно-исследовательского проекта ....	90
6.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .....	91
	Заключение .....	94
	Список использованной литературы.....	95
	Приложение А .....	99
	Приложение Б– Детектор дорожной разметки .....	120
	Приложение В «Lines.py» .....	121
	Приложение Г Обучение HOG-SVM детектора .....	123
	Приложение Д Модуль классификации сигналов светофора .....	124

Приложение Е Модуль детектирования и классификации знаков дорожного движения.....	125
Приложение Ж Модуль распознавания участников дорожного движения .....	126
Приложение З Подготовка детектора для модуля распознавания участников дорожного движения.....	127
Приложение И Полный код разработанной системы распознавания объектов дорожного движения.....	128

## ВВЕДЕНИЕ

В настоящее время набор задач, решаемых системами компьютерного зрения, приобретает всё большую область распространения.

Большинство таких задач подразумевает собой использование методов обработки изображений с помощью нейронных сетей. Такой подход позволяет создавать полноценные системы компьютерного зрения, способные в полной мере анализировать происходящее на исходном изображении и координировать внешнюю систему, основываясь на результатах проведённого анализа.

В области транспортных средств, технологии компьютерного зрения также имеют широкий диапазон использования. Наиболее полновесными задачами для таких систем являются задачи активной и пассивной помощи водителю.

Рассматриваемые в данной работе методы классификации объектов дорожного движения являются базой для создания системы анализа ситуации на дорожном полотне.

В технической литературе описаны методы детектирования отдельных типов объектов дорожного движения, позволяющие найти на изображении тот или иной объект, однако полноценных систем, позволяющих детектировать (а также классифицировать) достаточное количество объектов для комплексного анализа ситуации на дорожном полотне в открытом доступе обнаружить не удалось. Поэтому в процессе выполнения работы, алгоритм решения задачи создания классификации объектов дорожного движения был разделён на следующие этапы, позволяющие выделить распознавание наиболее ключевых объектов и провести их классификацию отдельно для каждого из таких объектов:

- исследование и реализация алгоритма детектирования дорожной полосы;
- исследование и реализация алгоритма детектирования светофоров;
- реализация алгоритма классификации сигналов светофора;

– исследование и реализация алгоритма детектирования и классификации участников дорожного движения;

– исследование и реализация алгоритма детектирования знаков дорожного движения;

– исследование и реализация алгоритма классификации знаков дорожного движения;

– исследование реализованного алгоритма на результативность и безошибочность выполнения с использованием тестовых примеров.

Объектом исследования в рамках выполнения данной работы является система анализа ситуации на дорожной полосе. Предметом исследования являются методы детектирования и классификации объектов на изображении и видеопоследовательностях.

## ОБЗОР ЛИТЕРАТУРЫ

Алгоритмизация детектирования и классификация достаточно часто упоминается в определённых источниках литературы, однако, большая часть задач, связанных с распознаванием, не имеют комплексного освещения в научной литературе, а большая часть алгоритмов представлена в технической документации к наборам инструментов компьютерного зрения.

В источниках [1] и [5] собраны основные методы, используемые библиотекой «OpenCV» под Python, необходимые для реализации работы с изображениями в контексте распознавания объектов поставленной задачи.

В [7] и [8] рассмотрены основные аспекты теории обработки данных на изображениях с использованием технологии компьютерного зрения.

«Основные концепции нейронных сетей» Каллана Р. [12] раскрывают основные понятия, связанные с нейронными сетями, а также раскрывают тематику базовых моделей нейронных сетей, важные для понимания основных принципов работы нейронных сетей.

Книга [11] является относительно новой научно-технической литературой, которая освещает основные аспекты глубокого машинного обучения. В данном источнике описываются новые архитектуры и алгоритмы обучения, раскрывающие новые методологии и подходы к алгоритмам распознавания объектов.

## **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СРЕДСТВ РАЗРАБОТКИ**

Решение задачи анализа, детектирования и классификации объектов на изображениях и видеопоследовательностях со сложной фоновой структурой относится к числу наиболее распространённых областей использования алгоритмов компьютерного зрения.

Некоторые существующие на данный момент библиотеки и фреймворки позволяют разрешить ряд проблем с обработкой изображений или кадров из видеоряда, а также привести их к состоянию готовности к последующим процедурам обработки методами анализа или детектирования.

В сфере технологий работы с транспортными средствами в настоящее время алгоритмы компьютерного зрения имеют высокую степень интеграции, являясь базисом решения различного рода задач.

### **1.1 OpenCV**

OpenCV представляет собой разработанную и оптимизированную библиотеку компьютерного зрения, для обработки изображений с открытым исходным кодом. Данная библиотека также поддерживает выполнение моделей для машинного обучения и искусственного интеллекта.

Помимо кроссплатформенности, обусловленной поддержкой многих языков программирования (в том числе и Python), позволяющей использовать приложения на различных системах, OpenCV примечательна также высокой степенью эффективности проведения вычислений, которая достигается посредством векторизации большого числа операторов и функций.

Все алгоритмы в OpenCV реализованы на языке C++. Однако эти же алгоритмы могут использоваться в среде разработки под Python. Такая возможность реализована благодаря генераторам привязок. Такие генераторы создают «мост» между C++ и Python, который позволяет пользователю вызывать функции C++ из Python. Поэтому расширение всех функций в OpenCV на Python путем написания их функций-оболочек вручную – это трудоемкая задача. Таким образом, OpenCV делает это более разумным способом. OpenCV автоматически генерирует эти функции-оболочки из заголовков C++, используя некоторые скрипты Python, которые находятся в «modules/python/src2» [1].

В данной работе OpenCV исполняет роль базовой компоненты, предоставляющей инструментарий для работы с входными данными, представленными видеопоследовательностью. Данная библиотека позволяет представить видеоряд в качестве набора кадров (фреймов), каждый из которых предполагается рассматривать как отдельное изображение. Также OpenCV позволяет производить над полученными фреймами определённые действия, такие как сложение, смещение, сглаживание, аффинное преобразование, вращение, транспонирование, применение различного рода фильтров, переход от одного цветового пространства к другому и т.д.

Также OpenCV поддерживает работу с потоковым видео, полученным напрямую с камеры, что открывает для разрабатываемой системы потенциал обработки и анализа входных данных в режиме реального времени, что является крайне важным аспектом в проектировании системы компьютерного зрения в области транспортных средств.

## **1.2 Dlib**

Dlib – это современная библиотека, представляющая собой набор методов и алгоритмов машинного обучения, а также инструментарий для создания сложного программного обеспечения для решения задач в областях промышленности, робототехники, мобильных телефонов, а также высокопроизводительных вычислительных средах.

Не смотря на то, что Dlib реализована на C++, существует возможность использования большинства её методов из приложений Python, а открытый исходный код позволяет использовать данную библиотеку бесплатно в любом приложении.

В данной работе используется метод «simple\_object\_detector», который позволяет создать и обучить модель для детектирования определённых объектов на изображении.

Для тренировки такой модели в качестве обучающей выборки на вход подаются предварительно размеченные изображения, содержащие детектируемый объект. Далее такие изображения анализируются с помощью детектора гистограммы направленных градиентов (рис. 1.2.1), позволяющего рассчитать градиент в каждой точке пространства, и в зависимости от направления этого градиента записать в гистограмму некоторое значение.

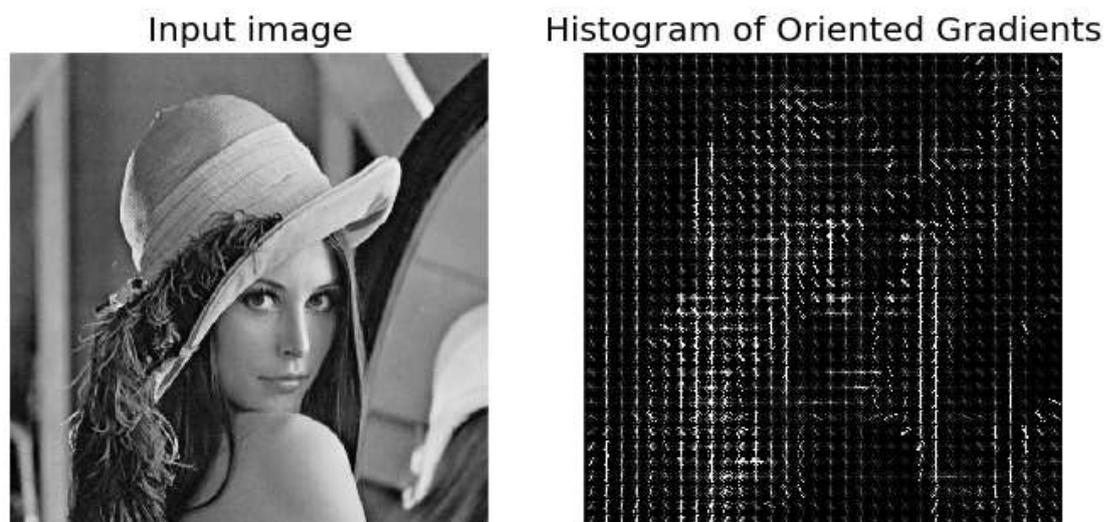


Рисунок 1.2.1 Гистограмма направленных градиентов (HOG)

С помощью такого анализа входных данных формируется набор векторов, каждый из которых описывает точку в  $n$ -мерном пространстве. Следующим шагом с помощью метода опорных векторов (SVM) наборы полученных точек делятся на два кластера (рис. 1.2.2).

Принцип работы SVM заключается в разделении гиперплоскостью векторов признаков, принадлежащих объектам и векторов признаков, принадлежащих фону, в многомерном пространстве. Результатом работы SVM является граница, проходящая через полученные точки, которая является критерием детектирования объекта. Таким образом, решение проблемы классификации сводится к определению того, по какую сторону полученной границы находится классифицируемый вектор признаков.

Оптимальной разделяющей гиперплоскостью принято считать гиперплоскость, расстояние от которой до ближайшего вектора с любой стороны – максимально. Зачастую векторы признаков объектов и векторы признаков фона линейно неразделимы в пространстве, образованном ими. В таком случае применяется нелинейное преобразование пространства в пространство большей размерности, где найденные вектора становятся линейно разделимыми.

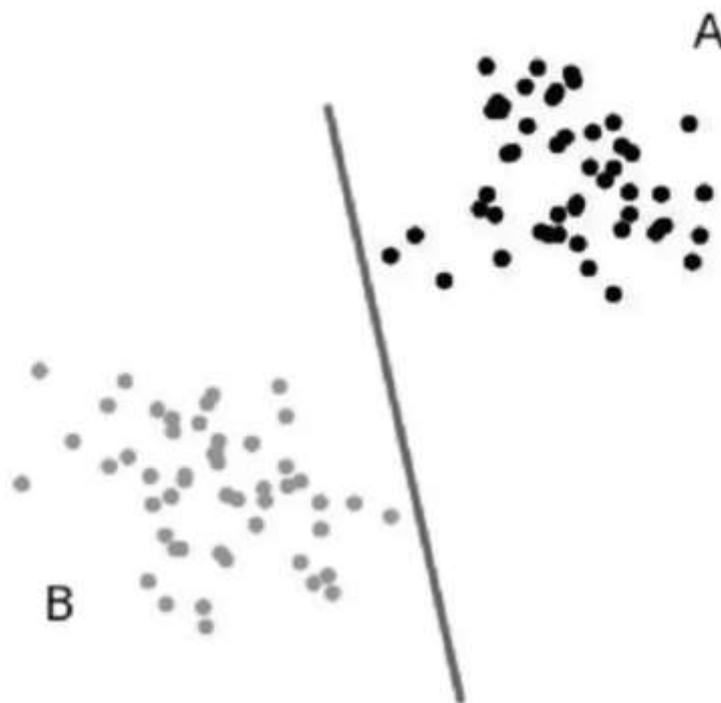


Рисунок 1.2.1 Метод опорных векторов

Таким образом формируется обученная модель. Во время работы такого детектора для каждого скользящего окна с помощью гистограммы направленных градиентов аналогичным образом рассчитываются точки и производится проверка, к какому из кластеров модели принадлежит найденная точка. Если оказывается, что точка принадлежит кластеру искомым объектов, детектор подаёт на выход информацию о найденном объекте.

Такой подход необходим в случае работы с объектами, детектирование которых весьма узконаправлено, и готовых обученных моделей, способных точно определить объекты, относящиеся к рассматриваемому в задаче конкретному классу, не существует. Данный метод позволяет обучить собственную модель для детектора и извлечь из изображения координаты целевого объекта.

### 1.3 TensorFlow и Keras

TensorFlow – это библиотека с открытым исходным кодом для численных вычислений, первоначально разработанная исследователями и инженерами, работающими в команде Google Brain. Данная библиотека использует реализацию ряда узлов обработки, представляющих собой математическую операцию. Ряд таких узлов называется «графом» (рис. 1.3.1).

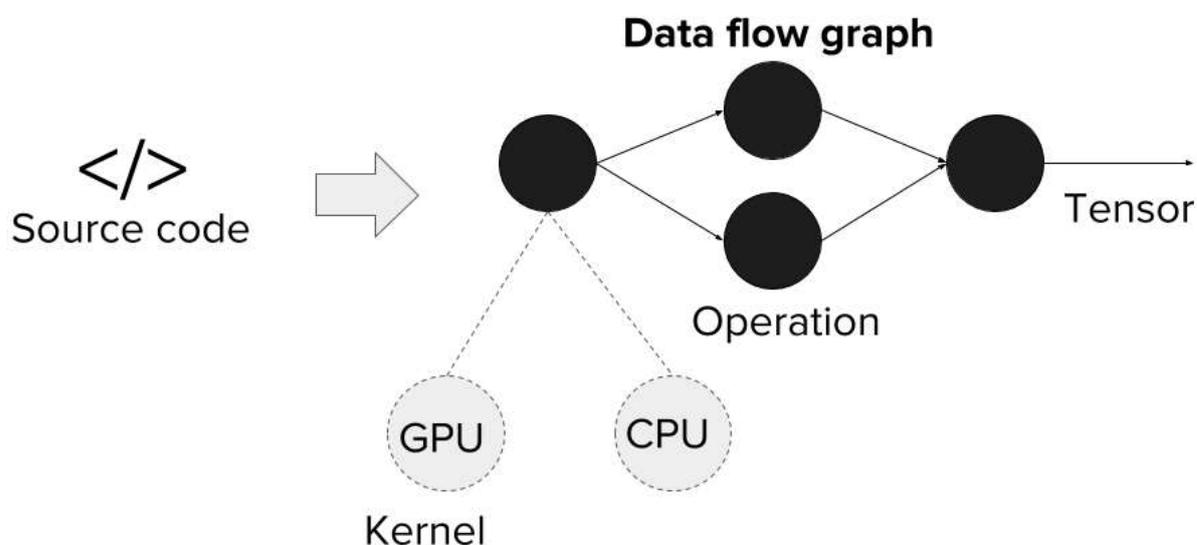


Рисунок 1.3.1 Схема работы TensorFlow

Основной целью создания рассматриваемой библиотеки является предоставление возможности наиболее простого в использовании интерфейса прикладного программирования при разработке алгоритмов машинного обучения и их развёртывания в кластере или на процессорах [2].

Keras – это высокоуровневый интерфейс прикладного программирования, использующий функционал TensorFlow, разработанный для реализации его наиболее мощных сложных функций, настроенный для работы с Python.

Базовая структура данных нейросети представлена слоями. Keras построен на основе представления базовой абстракции слоя. Слоем является объект, содержащий в себе некоторое состояние (веса) и вычисления. Веса объявлены в методе «build()», а вычисления – в методе «call()».

Распознавание изображений является задачей введения исходного изображения в нейронную сеть и выведения ею определяющей метки для анализируемого изображения. Полученная нейронной сетью метка будет сопоставлена с определённым заранее классом. Изображение может относиться к нескольким классам или же только к одному. Если такой класс идентификации только один, то задача обозначается как «Распознавание», если же таких классов несколько – как «Классификация».

Подмножеством классификации изображений является обнаружение объектов, когда конкретные экземпляры объектов идентифицируются как принадлежащие к определенному классу, например, животные, автомобили или люди [3].

## 1.4 ImageAI

ImageAI – это библиотека Python, созданная для того, чтобы дать разработчикам, исследователям и студентам возможность создавать приложения и системы с автономным глубоким обучением и возможностями компьютерного зрения, используя несколько строк кода. Эта документация предназначена для предоставления подробного представления обо всех классах и функциях, доступных в ImageAI, в сочетании с рядом примеров кода. ImageAI – это проект, разработанный Мозесом Олафенвой и Джоном Олафенвой [4].

Методы машинного обучения имеют высокую точность только в случае использования выборок точных осмысленных данных. Для наибольшей эффективности работы обучаемой модели необходимо предоставить как можно больше ресурсов для её обучения.

ImageAI поддерживает большое количество современных алгоритмов машинного обучения для распознавания изображений, детектирования и классифицирования объектов на изображении и видеопоследовательностях.

Данная библиотека поддерживает детектирование изображений с использованием четырёх различных алгоритмов машинного обучения, обученных на наборе данных «ImageNet-1000». ImageAI также поддерживает детектирование и классификация объектов с помощью моделей «RetinaNet», «YOLOv3», «TinyYOLOv3», обученных на данных «COCO».

ImageAI поддерживает также расширенный анализ видео с интервальными обратными вызовами и функциями для обучения моделей распознавания изображений на пользовательских наборах данных.

В данной работе были применены основные возможности данной библиотеки по детектированию и классификации для распознавания участников дорожного движения на изображении.

Выбор данной библиотеки обусловлен наиболее простым в использовании интерфейсом прикладного программирования для работы с моделями распознавания объектов на изображении, а также поддержкой наиболее популярных архитектур нейронной сети, таких как «RetinaNet» и «YOLO».

## 2 РАЗРАБОТКА АЛГОРИТМОВ

В процессе решения задачи алгоритмизации системы детектирования и классификации объектов дорожного движения процесс разработки алгоритмов был продифференцирован на несколько самостоятельных блоков, каждый из которых использует наиболее подходящие и оптимальные методы решения для исходной задачи. Дифференциация была проведена на основе класса, к которому принадлежит тот или иной детектируемый объект, а также на основе визуальных особенностей рассматриваемых объектов, таких как форма и цвет, а также изменчивость и необходимость проведения классификации после детектирования.

Чтобы произвести распознавание объекта в видеоряде необходимо рассмотреть исходные данные с точки зрения их представления в методах компьютерного зрения. Технически одно статическое изображение в видео называется видеокadre. В большинстве видео количество кадров в секунду видео колеблется от 20 до 32, и это значение называется частотой кадров в секунду (fps).

Для упрощения последующей работы алгоритмизации детекторов и классификаторов было принято решение рассматривать видеоряд как последовательный набор кадров, каждый из которых является самостоятельным изображением.

Входное изображение, являющееся частью исходного видеоряда, в свою очередь, является массивом пикселей. Таким образом действия детекторов ориентированы на преобразования и исследования массивов таких пикселей, каждый из которых имеет определённые координаты и цвет в качестве характеристик.

## 2.1 Детектирование полосы дорожной разметки

Определение расположения полосы движения относительно движущегося автомобиля является одной из наиболее весомых задач для систем помощи водителю и автономных автомобилей. Данный модуль несёт функционал слежения за траекторией движения машины по выбранному маршруту и призван предупреждать возможные съезды с дорожной полосы.

Первым шагом в процессе работы детектора является преобразование исходного цветового пространства изображения в цветовую схему «Градации серого». Такой подход позволяет выделить наиболее явно границы разметки, а также отбросить лишнюю информацию, не имеющую отношение к разметке дорожной полосы. Данное преобразование является следствием того, что разметка зачастую имеет белый (либо контрастный тёмному асфальту) цвет.

Чтобы уменьшить контраст от обесцвеченных участков дорожной полосы необходимо выполнить затемнение исходного изображения.

Необходимо также применить размытие по Гауссу, чтобы преобразовать лишние контуры в плавный градиент и избавиться от лишнего визуального шума. Гауссово размытие рассчитывается по формуле:

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.1.1)$$

где  $x$ ,  $y$  — координаты точки, а  $\sigma$  — среднеквадратическое отклонение нормального распределения.

При вычислении формулы по двум координатам в качестве результата будет получена поверхность, контуры которой образуют собой концентрические окружности с нормальным распределением относительно центра.



Рисунок 2.1.2 Пример использования размытия по Гауссу

Следующим шагом после предобработки к размытым снимкам применяется детектор границ – алгоритм, который обнаруживает края на основе изменения градиента (рис. 2.1.2).

Хотя первым шагом его работы по умолчанию является сглаживание изображения с размером ядра 5, наиболее корректным будет явное размытие по Гауссу на предыдущем этапе. Другие шаги работы детектора включают в себя: нахождение градиента интенсивности изображения, подавление максимумов, а также пороговый гистерезис.



Рисунок 2.1.2 Результат применения детектора границ

Даже после использования детектора границ на снимках остаётся много лишних линий и краёв, не связанных с разметкой. Для того, чтобы исключить все лишние границы необходимо определить форму и размеры для области интереса.

Область интереса – это многоугольник, охватывающий наиболее полезную часть изображения. Данный шаг является одним из ключевых в производстве анализа разметки, т.к. позволяет сформировать маску для входного изображения, тем самым отсеять все визуальные шумы и помехи, с которыми нет возможности справиться посредством предобработки кадров анализируемого видеопотока.

Заключительным шагом в работе детектора является анализ выделенной обработанной области на изображении и построение направляющих разметки.

## **2.2 Детектирование светофоров и классификация их сигналов**

Одними из важнейших объектов с точки зрения информативности на дорожном полотне являются сигналы светофора. Именно классификация сигналов светофора по признаку цвета сигнальной лампы позволяет системе автономного автомобиля принимать правильные решения о необходимости остановки или возможности продолжения движения.

Все светофоры имеют достаточно похожую форму, и внешне редко отличаются друг от друга. В такой ситуации имеет смысл использовать детектор, использующий гистограмму направленных градиентов и метод опорных векторов, для анализа входного изображения и детектирования на нём объектов, подлежащих последующей классификации.

Для обучения модели такого детектора необходимо сформировать масштабную выборку изображений, содержащих в себе детектируемый объект.

Сформировав выборку таких изображений, необходимо разметить их таким образом, чтобы координаты областей, содержащих искомые объекты, были указаны для каждого элемента выборки соответственно. Такой набор данных позволит сформировать массив векторов направлений градиента, каждый из которых представляет собой точку в  $n$ -мерном пространстве классификации, которое будет разделено на два кластера методом опорных векторов при обучении модели детектора.

Обученному детектору на вход подаётся исходное изображение, которым и является каждый кадр видеоряда. Изображение внутри скользящего окна детектора также преобразуется в набор векторов направлений градиента, после чего детектор определяет кластер, к которому принадлежит та или иная точка, на основе чего и формирует результат.

### **2.3 Детектирование и классификация дорожных знаков**

Начальным этапом построения алгоритма детектирования дорожных знаков является определение ключевых особенностей знаков относительно других объектов, находящихся на исходном изображении.

К таким особенностям можно отнести:

- форму знака;
- цвет знака;
- изображение на знаке.

Первым рубежом отсеивания в работе алгоритма детектирования был выбран именно цвет знака. Для того чтобы отсеять все изображения, цвета которых не фигурируют на знаках дорожного движения, было произведено преобразование исходного изображения с помощью цветового фильтра, который выполняет переход из rgb-пространства в бинаризованное (рис. 2.3.1).



Рисунок 2.3.1 Пример бинаризованного изображения

Бинаризация происходит с учётом допуска цветового диапазона, что позволяет привести искомые объекты к белому цвету, а отсеянные в процессе преобразования после применения фильтра – к чёрному.

Следующим шагом детектирования дорожных знаков является поиск контуров на полученном изображении. Так как после бинаризации было получено чёткое разграничение на чёрные и белые области, задача поиска контуров значительно упрощается.

Для того чтобы выделить из полученного множества контуров искомый объект, необходимо оставить тот контур, замкнутая которым область имеет наибольшую площадь. Таким образом, на данном этапе достигается детектирование объектов на изображении, принадлежащих определённому заранее классу по цветовому параметру.

Далее необходимо классифицировать найденные на изображении объекты, чтобы определить принадлежность к искомым объектам по остальным признакам, а также распознать, непосредственно, сам объект для дальнейшей обработки алгоритмами.

Первым шагом классификации является формирование референсной выборки для анализа полученных детектором результатов. Каждый элемент такой выборки должен представлять собой изображение, дающее наиболее полное представление об объекте классификации. Важно заметить, что каждое из таких изображений искомого объекта преобразуется в бинаризованное с использованием цветового фильтра, аналогичного использованному при предобработке входному изображению.

Следующим шагом необходимо выделить прямоугольную область наименьшей площади, в которую можно вписать найденный контур. Далее – привести изображение данной области к фиксированному разрешению, совпадающему с разрешением референсного изображения.

Заключительным этапом классификации является попиксельное сравнение полученного и ключевого изображения с последующим подсчётом совпадений. Основопологающим критерием функциональности такого сравнения является разительное различие количества совпадений при обнаружении искомого объекта и части изображения, не попадающей под классификацию.

### 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Для реализации системы распознавания объектов дорожного движения в рамках решения задачи магистерской диссертации был выбран высокоуровневый язык программирования общего назначения Python.

В основу данного выбора был положен ряд преимуществ, которыми обладает данный язык программирования:

- высокий уровень читаемости кода обеспечивает комфортный уровень концентрации внимания на синтаксисе, что, в свою очередь, позволяет сосредоточиться в большей степени именно на алгоритмизации;
- Python является бесплатным и открытым ресурсом, что является причиной быстрого развития и модернизации языка, а также высокой степени его популярности и поддержки со стороны комьюнити;
- совместимость с большим количеством платформ позволяет избежать проблем, связанных с кроссплатформенностью, типичных для большинства языков программирования;
- поддержка объектно-ориентированного программирования позволяет использовать объекты, в основе которых лежат данные и функционал;
- огромное количество библиотек, созданных или конвертированных для использования Python, позволяют облегчить работу с данными любого типа, в том числе и изображениями, а также являются основой решения различного рода задач.

Таблица 3.1 Сравнительная характеристика C++/Java/Python

Характеристика	C++11	Java 8	Python 3.5
Управление памятью	Ручное распределение Ручное освобождение	Ручное распределение Автоматическая сборка мусора	Автоматическое распределение Автоматическое освобождение
Длина кода	Более длинные строки кода по сравнению с Python	Примерно на 40% больше кода, чем в C++	В 3-5 раз короче, чем код на Java
Система типов	Статически типизированные код и компилятор	Статически типизированные код и компилятор	Динамически типизированный, интерпретатор строго типизирован, язык не содержит типов
Компилирование	Скомпилированный в машинный код	Компилируется в байтовый код, а затем интерпретируется JVM	Интерпретируется интерпретатором Python
Скорость выполнения (динамические массивы с числами, базовые математические операции)	100%	1176%	1800%
Портативность	Зависит от платформы, должен быть перекомпилирован для другой платформы	Независим от платформы, генерируемый байтовый код работает на каждой ОС	Независим от платформы, код интерпретируется при выполнении
Поддержка потоков	Встроенная поддержка начиная с C++11	Имеет встроенную поддержку потоков	Нет многопоточности
Сложность синтаксиса	Определение блоков с использованием {, окончание операторов с использованием ;	Определение блоков с использованием {, окончание операторов с использованием ;	Только использование отступов

### **3.1 Детектор дорожной разметки**

Дорожная разметка является одним из наиболее просто детектируемых объектов в силу своей однообразной формы и низкой степени изменчивости на различных изображениях. Обнаружение такого объекта можно реализовать посредством стандартных функций анализа изображений, не прибегая к обучению уникальной модели детектирования.

Построение детектора дорожной разметки было выполнено с использованием базовых методов обработки изображений библиотеки компьютерного зрения «OpenCV».

#### **3.1.1 Детектор границ Canny**

Перед тем, как перейти к детектированию границ, необходимо произвести предварительную подготовку исходного изображения.

Первым этапом предобработки является переход от цветового пространства BGR (для работы именно с таким форматом цвета изображения реализованы методы библиотеки «OpenCV») в пространство «Grayscale». Такой переход позволяет осуществить функция «cvtColor», с входным параметром «COLOR\_BGR2GRAY» (рис. 3.1.1.2).



Рисунок 3.1.1.1 Кадр исходного видеоряда



Рисунок 3.1.1.2 Результат перехода в цветовое пространство Graayscale

Чтобы применить фильтр Гауссова размытия, используется функция «GaussianBlur» (рис. 3.1.1.3).



Рисунок 3.1.1.3 Применение фильтра Гауссова размытия

Подготовленное изображение подаётся на вход функции «Canny», которая является непосредственной реализацией детектора границ для библиотеки «OpenCV».

Процесс работы детектора границ «Canny» включает в себя пять этапов обработки изображений:

- сглаживание изображение Гауссовым фильтром для подавления шума на входном изображении;
- поиск градиентов интенсивности изображения;
- применение методики «Не максимального подавления» для исключения ложной реакции на обнаружение края;
- определение потенциального края с применением двойного порога;
- завершение детектирования границ, подавление острых краёв, которые являются слабыми и не связаны с сильными краями (гистерезис).

Применив детектор границ «Canny» получим на выходе исходное изображение, представленное набором контуров (рис. 3.1.1.4).

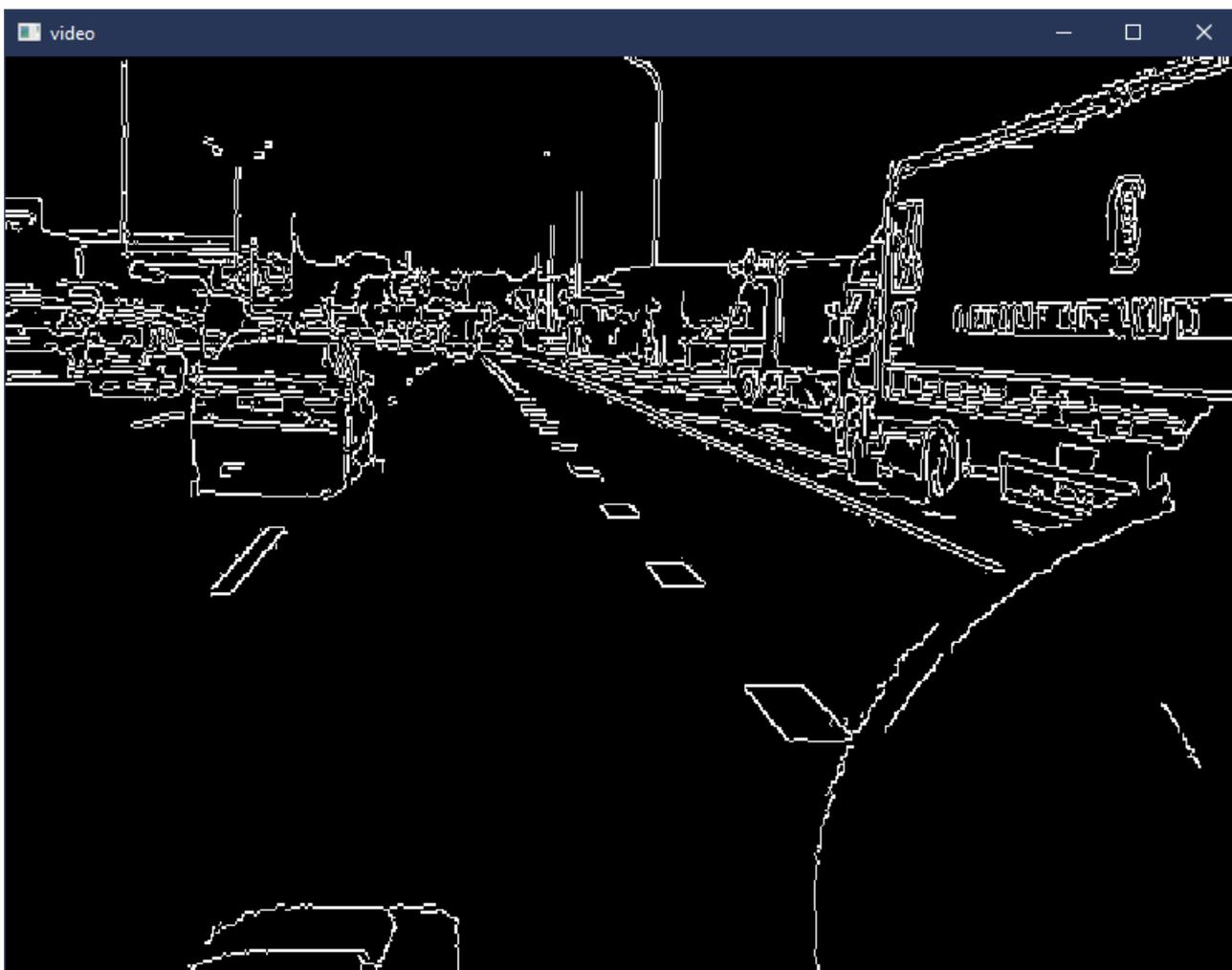


Рисунок 3.1.1.4 Результат применения детектора границ – «Canny»

### 3.1.2 Поиск области интереса

В случае анализа изображения, когда предметом поиска является полоса дорожной разметки, область интереса представляет собой вытянутую вдоль направления дороги трапецию.

Первым шагом в построении маски для области интереса является формирование массива координат вершин трапеции.

После корректного формирования точек, образующих трапецию, необходимо создать саму фигуру. Для этого функцией «zeros\_like» библиотеки «numpy» создаётся массив нулей, с формой и типом данных аналогичный исходному изображению и с помощью «fillPoly» заполняется область, образованная массивом координат вершин. Затем полученная фигура накладывается в виде маски на исходное изображение с помощью функции побитового «И» для каждого пикселя, отсекая, таким образом, те части изображения, которые находятся за пределами области интереса, представленной трапецией (рис. 3.1.2.1).



Рисунок 3.1.2.1 Результат наложения маски для отсечения области интереса

### 3.1.3 Вычисление и отрисовка направляющих

Для непосредственного вычисления направляющих дорожной разметки будем использовать функцию «HoughLinesP» библиотеки «OpenCV», которая представляет собой реализацию алгоритма поиска линий на изображении.

Алгоритм работы данной функции базируется на преобразовании линии Хафа. Линии на полотне изображения могут быть представлены двумя переменными (пусть  $t$  и  $b$  – для декартовой системы координат,  $r$  и  $\theta$  – для полярной системы координат, рис. 3.1.3.1).

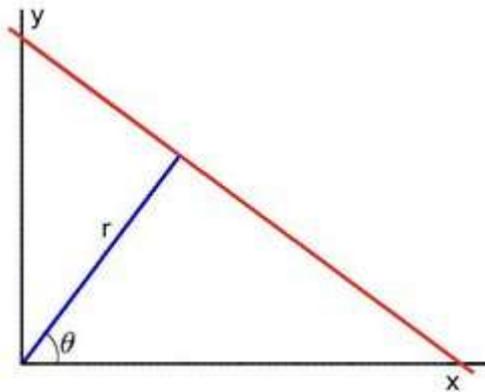


Рисунок 3.1.3.1 Представление линий в декартовой и полярной системах координат

Выразим линии в полярной системе для последующих преобразований Хафа:

$$y = \left(-\frac{\cos \theta}{\sin \theta}\right)x + \left(\frac{r}{\sin \theta}\right), \quad (3.1.3.1)$$

тогда, можно выразить:

$$r = x \cos \theta + y \sin \theta. \quad (3.1.3.2)$$

Следовательно, для каждой точки  $(x_0, y_0)$ , можно определить семейство линий, проходящих через эту точку как:

$$r_\theta = x_0 \cos \theta + y_0 \sin \theta. \quad (3.1.3.3)$$

Таким образом, каждая пара  $(r_\theta, \theta)$  представляет собой линию, проходящую через  $(x_0, y_0)$ .

Если для определённых  $(x_0, y_0)$  провести семейство линий, проходящих через эту точку, получим синусоиду (рис. 3.1.3.2).

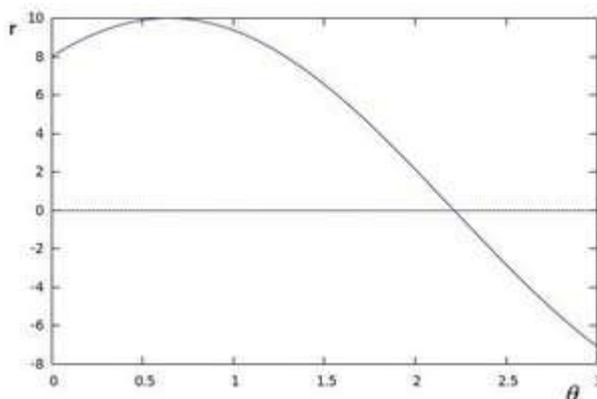


Рисунок 3.1.3.2 Синусоида в пространстве  $\theta - r$

Справедливо, также, отметить, что рассматриваются только точки такие, что  $r > 0$  и  $0 < \theta < 2\pi$ .

Если проделать ту же операцию для каждой из набора точек изображения, то в случае, когда кривые двух разных точек пересекутся на плоскости  $\theta - r$ , это будет означать, что обе точки принадлежат одной прямой (рис. 3.1.3.3).

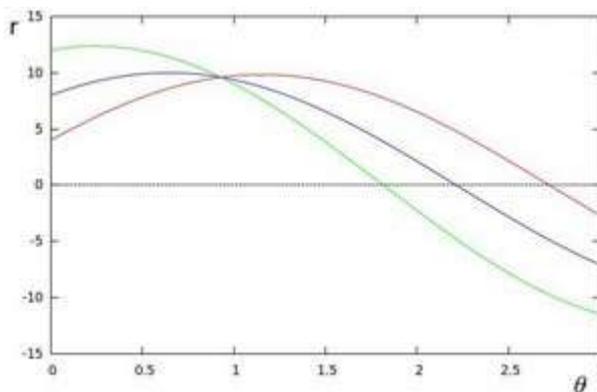


Рисунок 3.1.3.3 Пересечение двух синусоид на плоскости  $\theta - r$

Это значит, что в общем случае линию можно обнаружить, определив количество пересечений между кривыми. С увеличением количества пересекающихся кривых, увеличивается и количество точек, принадлежащих линии, представленной этим пересечением. Тогда можно определить порог минимального количества пересекающихся линий, необходимых, для обнаружения линии.

Это то, что делает преобразование линии Хафа. Оно отслеживает пересечение кривых каждой точки изображения. Если количество пересечений превышает установленный порог, то алгоритм считает это пересечение линией с параметрами  $(\theta, r_\theta)$  [5].

Произведя отрисовку линий по вычисленным координатам и заполнив пространство между ними, получим изображение, с задетектированной полосой разметки дорожного движения (рис. 3.1.3.4)

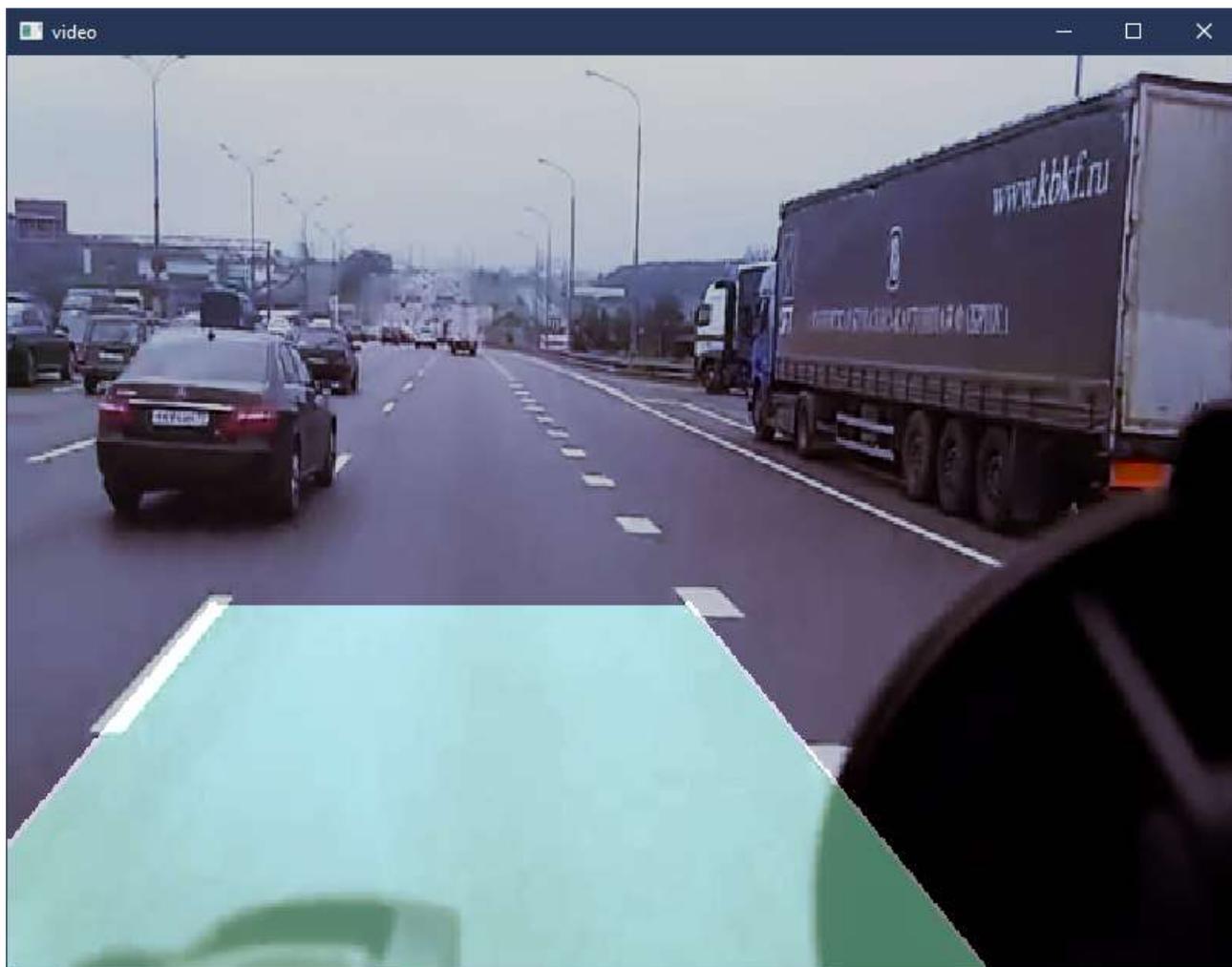


Рисунок 3.1.3.4 Работа детектора полосы разметки дорожного движения

### **3.2 Модуль обнаружения и распознавания сигналов светофора**

Для создания и обучения модели детектора были использованы методы библиотеки «Dlib», которые позволяют получить корректно работающий детектор для последующей реализации функционала классификации обнаруженных на изображении объектов.

### 3.2.1 Обучение модели детектора

Чтобы обучить модель для последующей работы детектора по распознаванию и классификации объектов, необходимо создать обучающую выборку с изображениями достаточно высокого разрешения.

На изображениях, включённых в выборку, необходимо произвести разметку детектируемых объектов. Данная работа была проведена вручную, посредством программного обеспечения, находящегося в открытом доступе, под названием «LabelImg» (рис. 3.2.1.1).

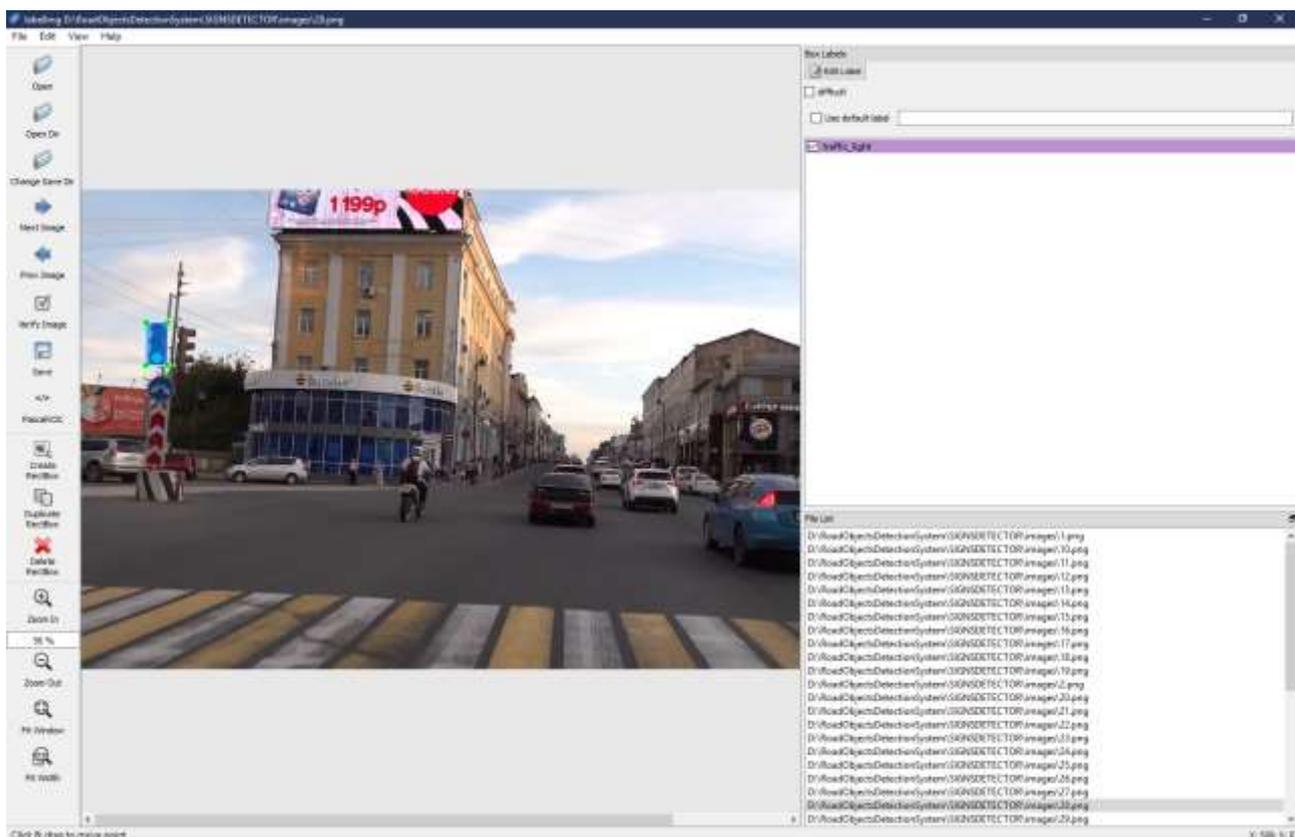


Рисунок 3.2.1.1 Интерфейс программы для формирования разметки обучающей выборки

Используемое программное обеспечение предоставляет возможность, с помощью визуализации изображения и определённого инструментария отметить на ключевом изображении область, где расположен детектируемый объект. При сохранении разметки, размеры изображения, координаты отмеченной области и название класса детектируемого изображения, принадлежащего к выделенной области, записываются в файл текстовой разметки разрешения «.xml» (рис. 3.2.1.2).

```

<?xml version="1.0"?>
- <annotation>
  <folder>images</folder>
  <filename>1.PNG</filename>
  <path>D:\RoadObjectsDetectionSystem\SIGNSDETECTOR\images\1.PNG</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>traffic_light</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>524</xmin>
      <ymin>423</ymin>
      <xmax>551</xmax>
      <ymax>485</ymax>
    </bndbox>
  </object>
  - <object>
    <name>traffic_light</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>556</xmin>
      <ymin>426</ymin>
      <xmax>579</xmax>
      <ymax>485</ymax>
    </bndbox>
  </object>
</annotation>

```

Рисунок 3.2.1.2 Пример аннотации к определённому изображению из обучающей выборки

При обучении модели детектирования, HOG-SVM детектор анализирует размеченные участки на изображениях и записывает проанализированные данные в модель.

Для того чтобы произвести обучение такой модели была использована функция библиотеки «Dlib» под названием «train\_simple\_object\_detector» с адресом расположения изображений из сформированной выборки и соответствующих им .xml-файлов-аннотаций, содержащих в себе данные разметки.

### 3.2.2 Реализация блока классификатора

Получив массив координат  $(x1, y1, x2, y2)$  области обнаруженного HOG-SVM детектором объекта, выполним подгонку изображения внутри области, принадлежащей прямоугольнику из массива полученных координат, под константно-заданное разрешение – 50px:100px с помощью метода «resize» библиотеки OpenCV. Также отсечём лишние края (по 10px с каждой стороны), на которых могут находиться частички фона с засветами (рис. 3.2.2.1).

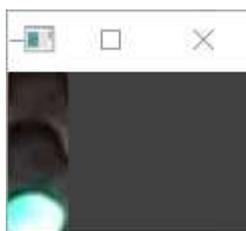


Рисунок 3.2.2.1 Результат работы детектора, преобразованный к константному разрешению

Используя функцию «cvtColor», с аргументом «COLOR\_BGR2HSV» преобразуем исходный кадр в цветовое пространство HSV.

Получим массив значений яркости пикселей исследуемого изображения.

Таким образом, чтобы определить, в какой зоне светофора зажжена сигнальная лампа, посчитаем сумму яркостей пикселей, принадлежащих каждой из зон.

Завершающим этапом классификации является сравнение полученных значений и выведение на изображение соответствующей информации о полученном результате (рис. 3.2.2.2).

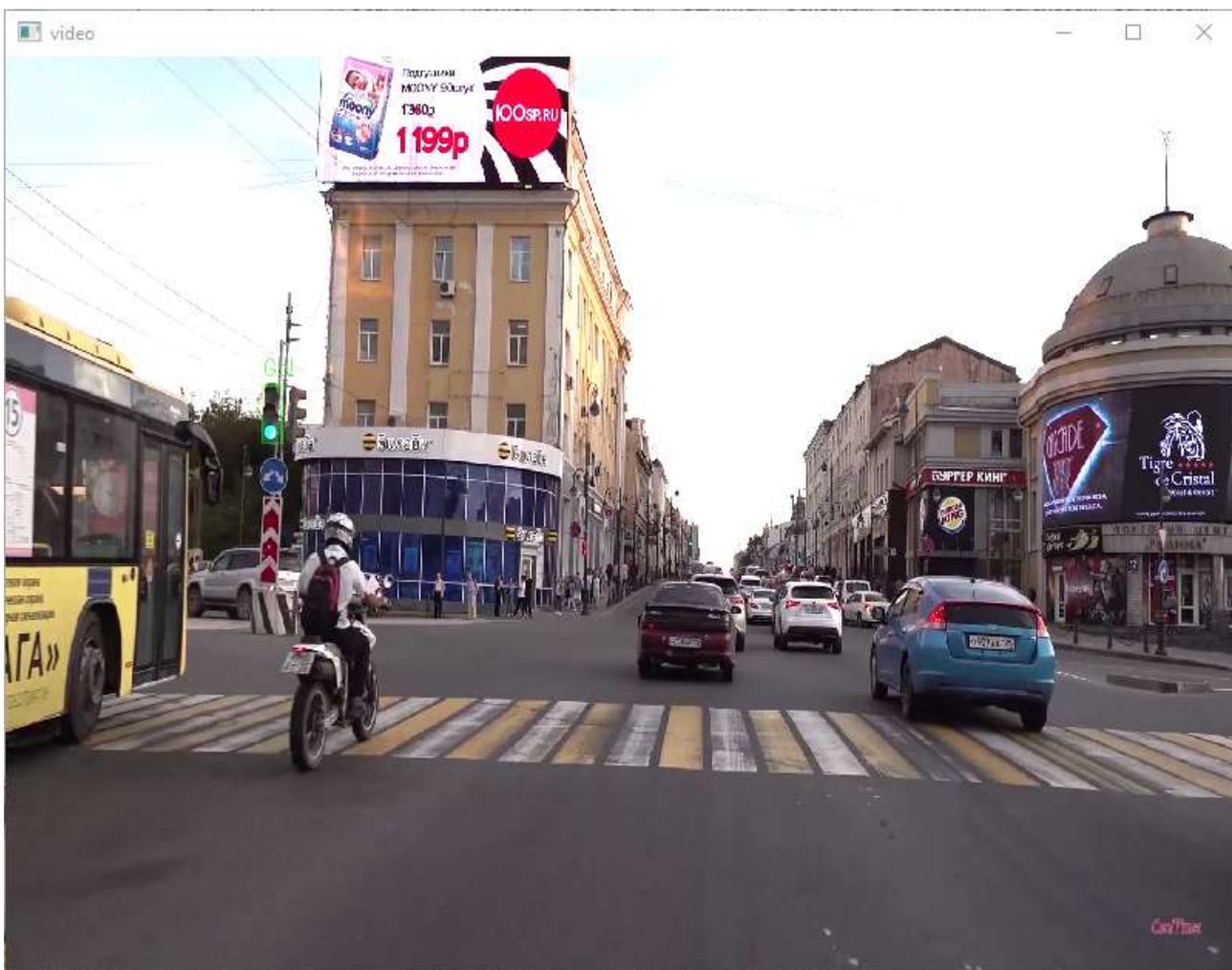


Рисунок 3.2.2.2 Результат работы детектора и классификатора сигналов светофора

### 3.3 Модуль распознавания дорожных знаков

Для детектирования и последующей классификации дорожных знаков были использованы стандартные методы цветового преобразования библиотеки OpenCV.

#### 3.3.1 Бинаризация изображения

Для того чтобы подготовить изображение к детектированию контуров необходимо произвести его бинаризацию с использованием цветового фильтра, позволяющего исключить определённый диапазон цветов.

Для бинаризации изображения был использован метод «InRange», входными параметрами которого являются начальные и конечные значения диапазона допустимой цветовой палитры.

Для увеличения плавности линий контура была использована функция «erode», с количеством итераций равным двум, которая позволяет размыть границы объектов переднего плана. Также, для увеличения границ изображений переднего плана, к полученному изображению была применена функция «dilate», с количеством итераций равным двум.

Такой подход позволяет увеличить плотность и толщину границ, что весьма существенно влияет на качество последующего детектирования контуров (рис. 3.3.1.1).

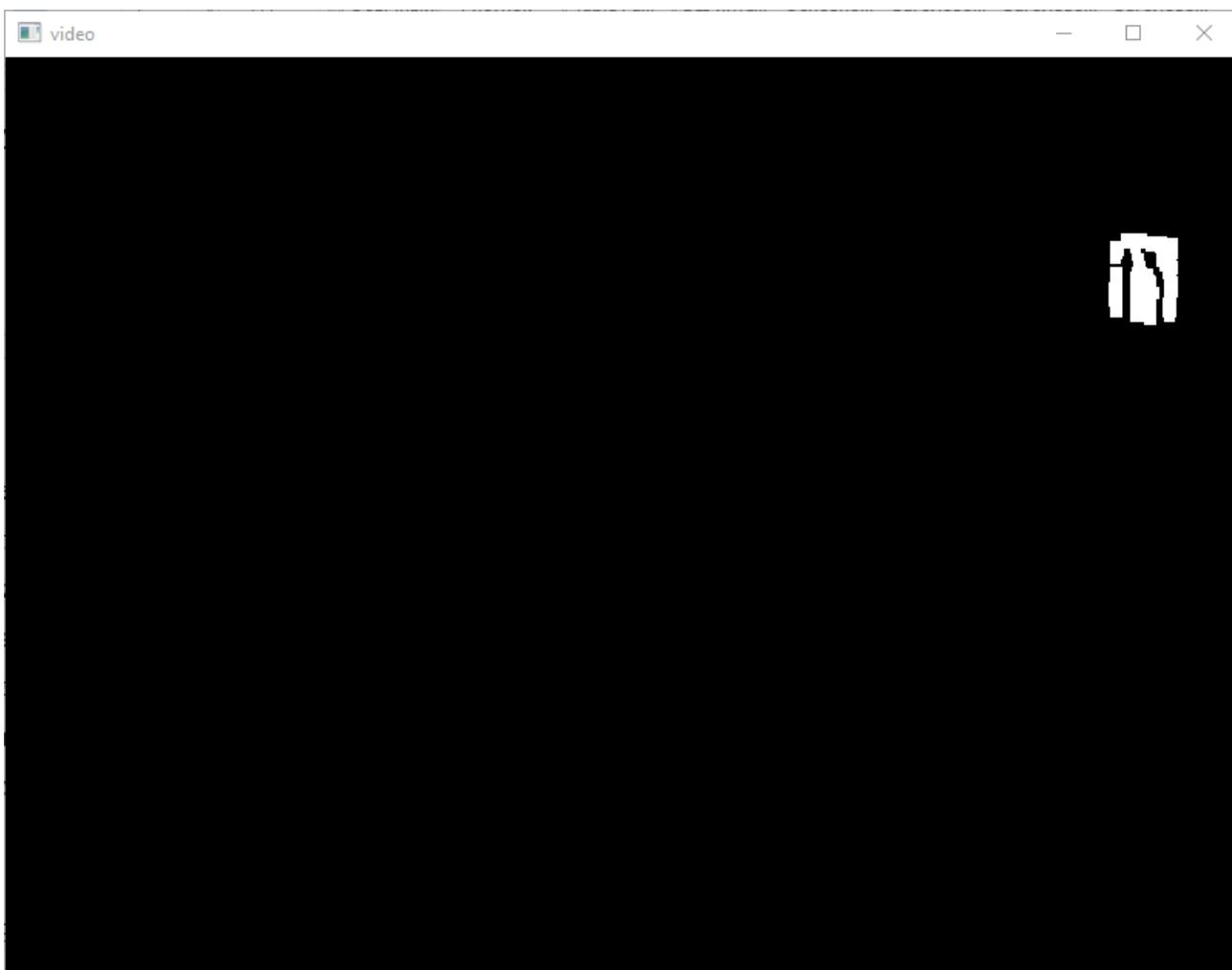


Рисунок 3.3.1.1 Результат бинаризации изображения

### 3.3.2 Детектирование контуров

Следующим шагом в построении детектора является поиск контуров, который реализован с помощью функции «findContours». Данная функция возвращает массив свойств всех найденных контуров на входном изображении. Из данных контуров нас интересуют только их координаты, которые находятся в первой ячейке полученного массива. Таким образом мы можем отсечь всю лишнюю информацию.

Для того чтобы получить массив координат наибольшего контура необходимо произвести сортировку по площади области изображения, замыкаемой каждым из контуров. Поиск такой площади можно произвести функцией «contourArea». Так как нас интересует элемент с наибольшей площадью, отсекаемой замкнутым контуром, будем использовать только первый элемент отсортированного массива.

Для последующей классификации отсечём область, сформированную полученными координатами, и приведём её к фиксированному размеру – 64px:64px (рис. 3.3.2.1а и 3.3.2.1б).

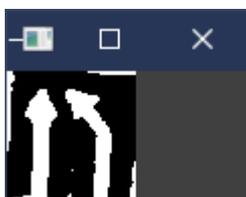


Рисунок 3.3.2.1а Область №1, определённая детектором

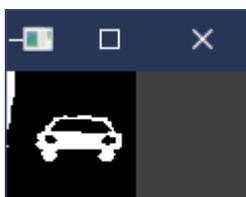


Рисунок 3.3.2.1а Область №2, определённая детектором

### 3.3.3 Референсная классификация

Для проведения классификации задетектированных дорожных знаков, необходимо реализовать алгоритм попиксельного сравнения полученного изображения с шаблонным.

Для того чтобы произвести сравнение, необходимо привести полученное и референсное изображение к единому разрешению. В данном случае было выбрано наиболее оптимальное разрешение, обеспечивающее достаточную эффективность сравнения, при допустимой погрешности – 64px:64px. Преобразование выполняется с помощью функции «resize» (рис. 3.3.3.1а и 3.3.3.1б).

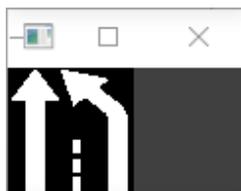


Рисунок 3.3.3.1а Референсное изображение №1

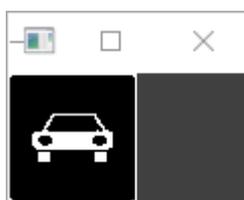


Рисунок 3.3.3.1б Референсное изображение №2

Финальным этапом классификации задетектированных знаков дорожного движения является реализация цикла для прохода по каждому пикселю полученного изображения и наполнение счётчиков совпадений для каждого ключа. После выхода из цикла, на основе данных, записанных в счётчики, необходимо сделать вывод о классификации задетектированного изображения и вывести информацию о результатах анализа на изображение в виде прямоугольной области с подписью, если объект принадлежит какому-либо классу (рис 3.3.3.2а и 3.3.3.2б).



Рисунок 3.3.3.2а Результат №1 работы классификатора

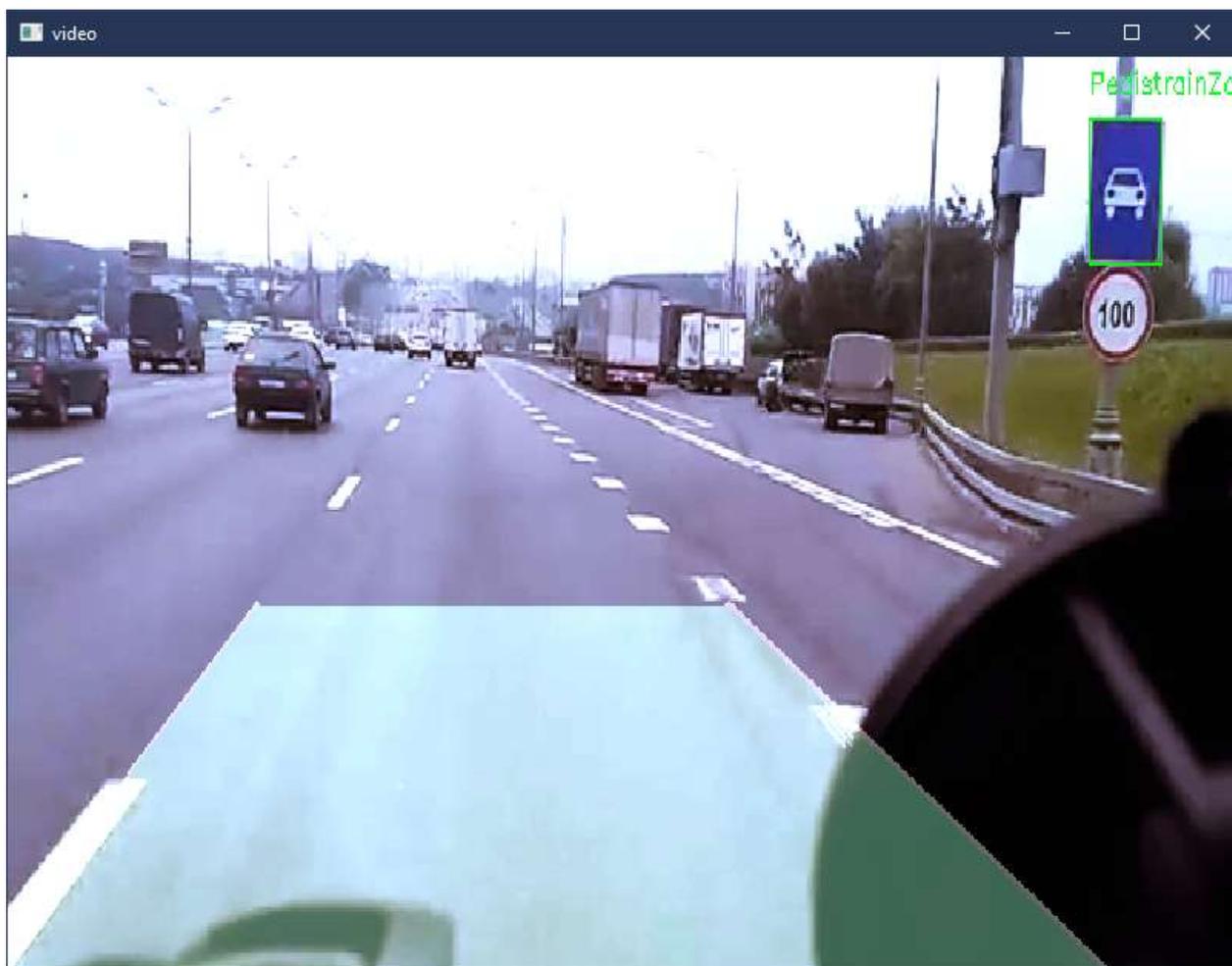


Рисунок 3.3.3.26 Результат №2 работы классификатора

### 3.4 Обнаружение участников дорожного движения

Задача обнаружения участников дорожного движения, таких как различные виды транспорта, велосипедисты и мотоциклисты, пешеходы и т.д. является слишком сложной для реализации её функционала посредством простых детекторов. Детектирование и классификация такого большого количества объектов, различных по своим свойствам и характеристикам, является достаточно масштабной задачей. В качестве программной компоненты для реализации алгоритма классификации было решено использовать надстройку для TensorFlow и Keras, упрощающую реализацию алгоритмов распознавания объектов – ImageAI.

### 3.4.1 Выбор архитектуры нейронной сети

Оценка скорости обработки видео производилась с использованием 10 различных видео с цветами, средняя продолжительность которых равнялась 20 секундам.

В таблицах 2.1, 2.2 представлены значения average precision по каждому классу для каждой архитектуры СНС, mAP, скорость обработки видео в кадрах в секунду (FPS) и объема весовых коэффициентов в МБ [6].

Таблица 3.4.1.1 Сравнение архитектур свёрточных нейронных сетей

Архитектура СНС	Скорость обработки видео СНС, FPS	mAP, %	Объём коэффициентов СНС, МБ
YOLOv3	35	0.6757	241
RetinaNet	29	0.6809	426
YOLOv3-tiny	65	0.3529	33

В качестве вывода к проведённым тестам можно сказать, что архитектура YOLOv3 показывает менее точный результат, по сравнению с RetinaNet, однако расходует меньшее количество времени на обработку видео. YOLOv3-tiny, в свою очередь, является намного более быстрой и производительной, чем её полновесный вариант, однако пропорционально увеличению скорости обработки, теряет в точности.

### 3.4.2 Настройка модели детектора

Для реализации корректной обработки потокового видео, необходимо предварительно настроить детектор, указав в качестве входных следующие характеристики:

- скорость детектирования (существует выбор между вариантами «normal», «fast», «faster» и «flash») определяет соотношение между скоростью обработки изображения свёрточной нейронной сетью и эффективностью детектирования и классификации;
- набор классов объектов, который представляет собой список типов объектов, которые будут детектироваться свёрточной нейронной сетью во время обработки изображения;
- в качестве входных и выходных данных указать массив точек;

- в качестве входного изображения указать извлечённый из видеопоследовательности анализируемый фрейм.

### **3.4.3 Встраивание детектора в систему**

В целях оптимизации производительности и снижения нагрузки на систему было принято решение обрабатывать нейронной сетью не каждый фрейм видеопотока, а разграничивать интервалы временем его воспроизведения.

Таким образом, было подобрано оптимальное время запуска обработки изображения свёрточной нейронной сетью – одна секунда времени воспроизведения. Данный подход к реализации позволяет избавиться от лишних итераций пересчёта координат детектируемых изображений и сократить нагрузку на систему в несколько раз.

После введение временного интервала запуска необходимо по истечении выбранного промежутка времени выводить результаты детектирования и классификации на изображения.

Для реализации данной цели, для каждого найденного объекта необходимо получить список координат области детектирования, после чего каждую из таких областей вписать в прямоугольник соответствующего размера, а также вывести над найденной областью информацию о классе, которому принадлежит инициализированный объект (рис. 3.4.3.1).

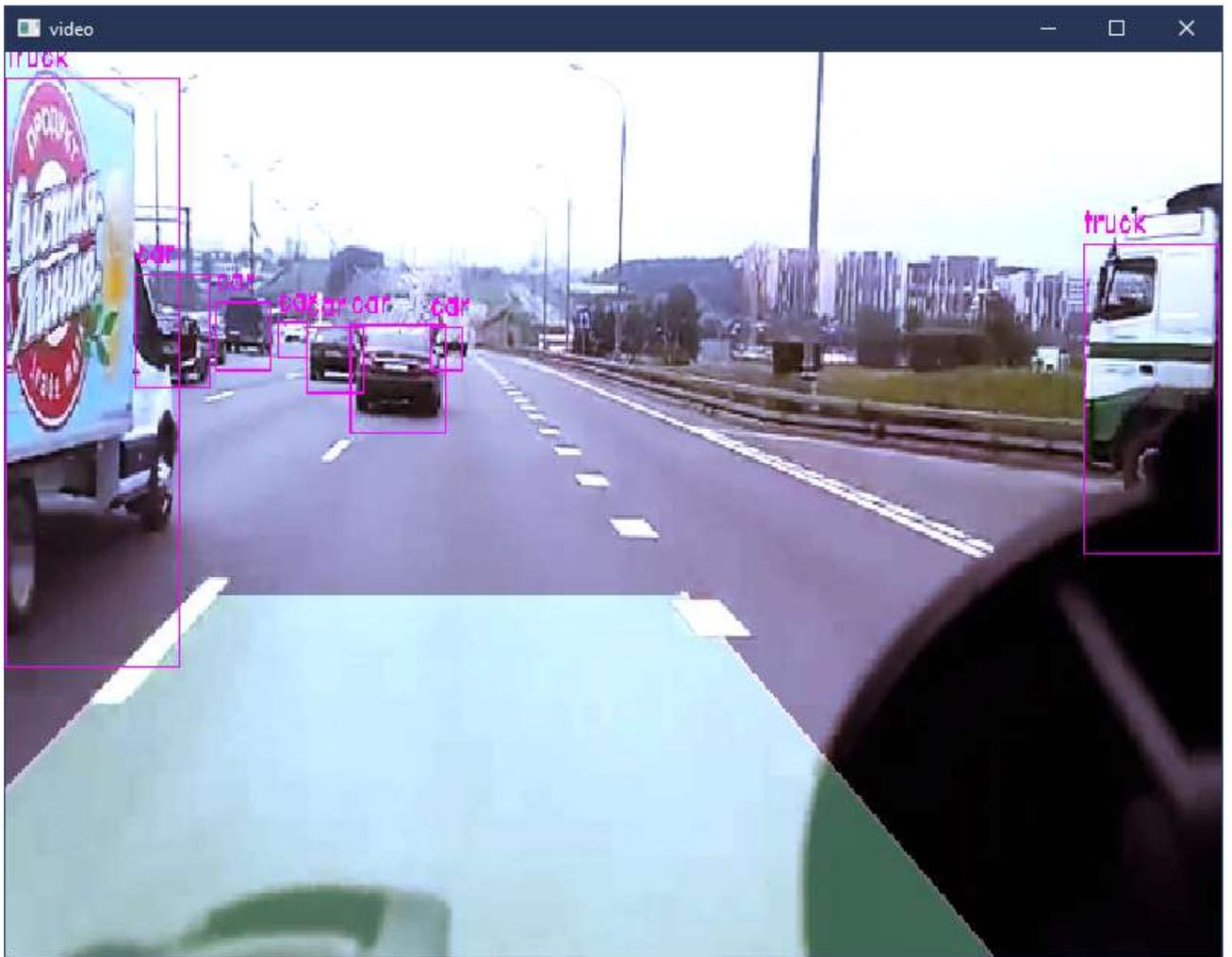


Рисунок 3.4.3.1 Результат работы классификатора участников дорожного движения

## 4 ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ ПРОГРАММНЫХ СРЕДСТВ

### 4.1 Подготовка данных для тестирования

Для тестирования работоспособности системы было выбрано два видеоряда содержащие в себе следующие объекты для детектирования:

- дорожную разметку;
- светофоры;
- знаки дорожного движения;
- непосредственных участников дорожного движения.

Выбранные видеопоследовательности были ограничены по времени для ограничения нагрузки на систему, а также для удобства проведения тестирования разработанной системы.

Разрешения, используемых в качестве тестовых данных видеопоследовательностей, были специально подобраны таким образом, чтобы протестировать работоспособность системы на данных различного качества и формата (рис. 4.1.1.1а и 4.1.1.1б).

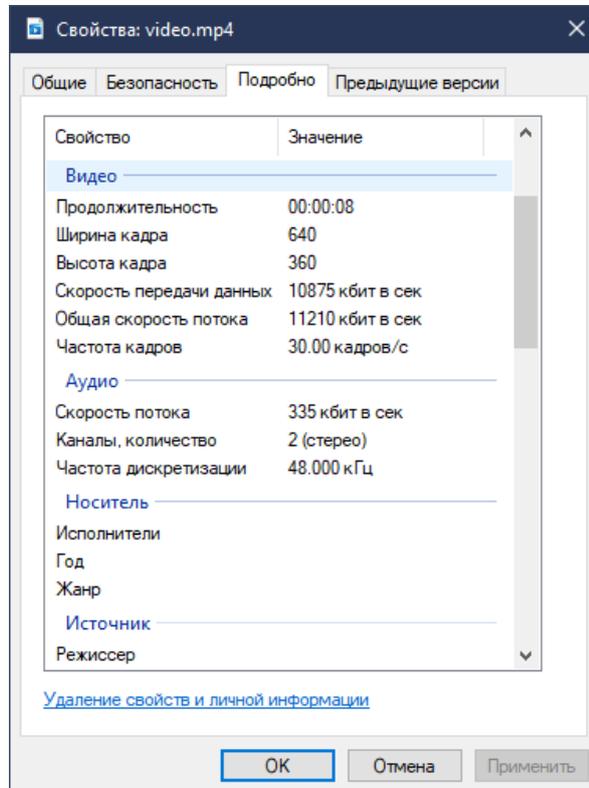


Рисунок 4.1.1.1а Свойства тестового видеоряда №1

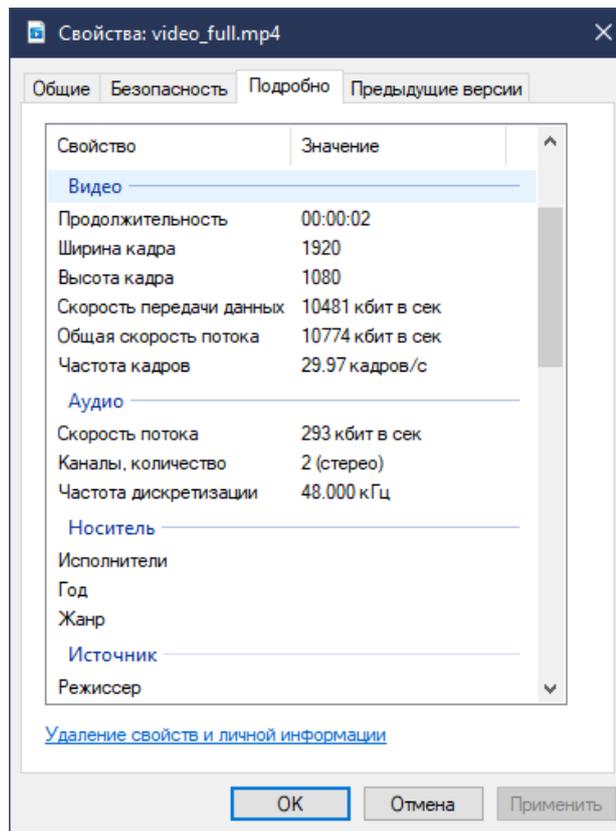


Рисунок 4.1.1.1б Свойства тестового видеоряда №2

Тестирование было произведено на ПК со следующими характеристиками:

Таблица 4.1.1.1 Характеристики ПК для тестирования

Параметры конфигурации	Характеристика
Процессор	AMD Ryzan 5 1600
ОЗУ	16GB
Видеокарта	MSI GeForce GTX 1660 Gaming X 6G
Материнская плата	GIGABYTE B450
Жесткий диск	500GB

## 4.2 Тестирование системы

При выполнении одновременного тестирования всех модулей в качестве полноценной системы-анализатора изображения с видеорегистратора были получены следующие результаты обработки:

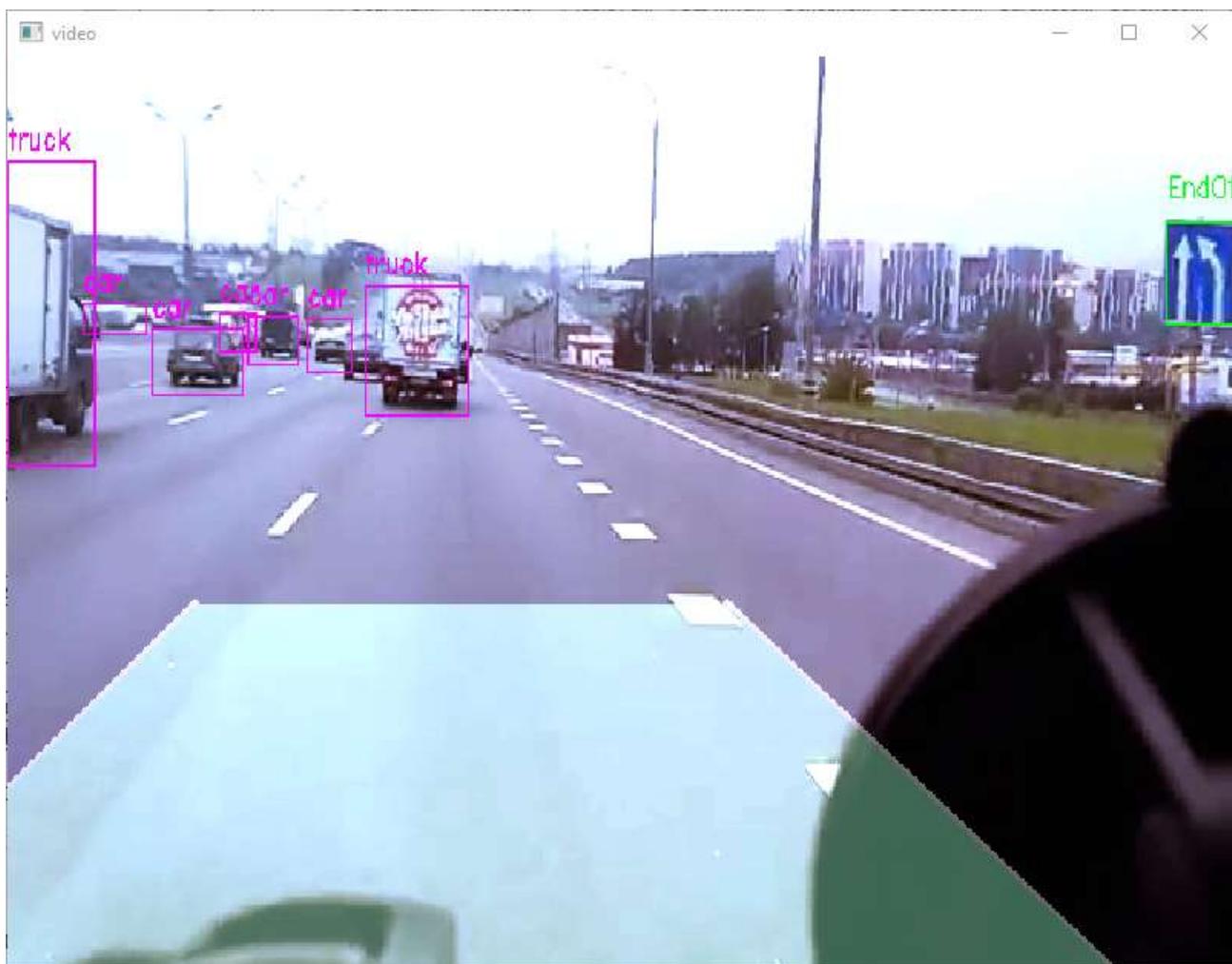


Рисунок 4.2.1.1а Результат работы системы распознавания объектов дорожного движения №1

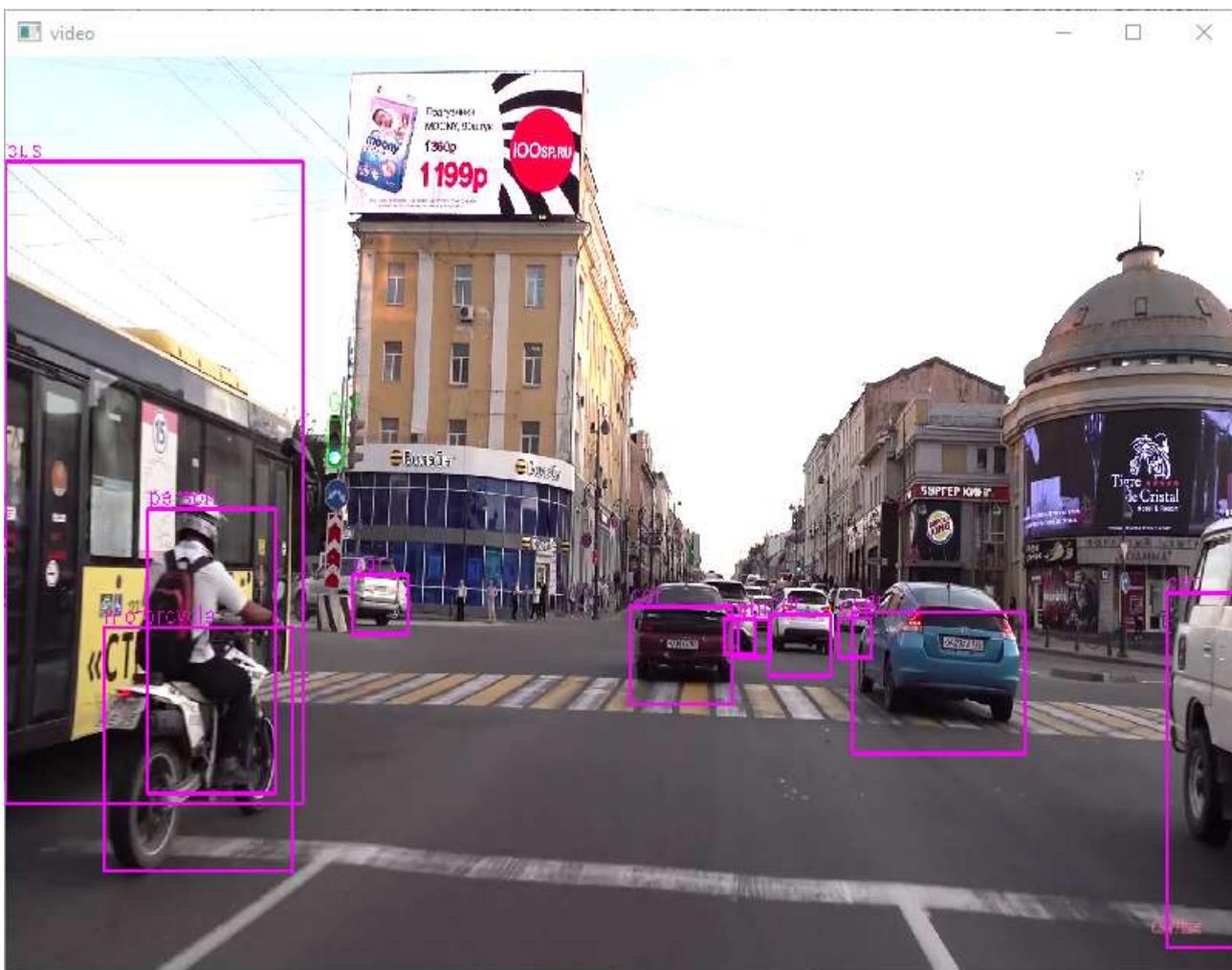


Рисунок 4.2.1.16 Результат работы системы распознавания объектов дорожного движения №2

Были сформированы также тестовые версии системы, для измерения точности детектирования. Так как, каждый из модулей системы позволяет настроить минимальный порог детектирования, то точность обнаружения каждого объекта не может быть ниже такого порога.

В процессе проведения тестов было выявлено, что «комфортная» точность (с минимальным количеством потерь при детектировании и максимальной эффективностью) для всей системы составляет 70%. При детектировании объектов на тестовой выборке все объекты были определены с точность более 70% с запасом 10-20% (рис. 4.2.1.2а, 4.2.1.2б и 4.2.1.2в).



Рисунок 4.2.1.2а Точность детектирования светофора



Рисунок 4.2.1.2б Точность детектирования знаков

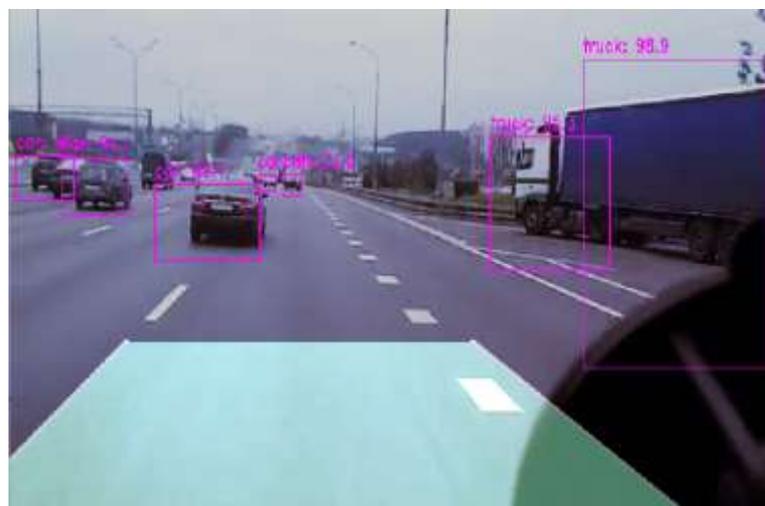


Рисунок 4.2.1.2в Точность детектирования автомобилей

Тестирование отдельных блоков системы показало следующие нюансы её работы:

- необходимость корректировки области интереса для детектора разметки в соответствии с расположением камеры записывающего устройства в автомобиле;

- прямая зависимость между выборкой обучающих данных и эффективностью работы детектора сигналов светофора является как большим потенциалом для последующего развития данного модуля в системе, так и уязвимостью на этапе, когда достаточная выборка для обучения модели ещё не сформирована;

- определение контура по наибольшей отсекаемой им области не всегда является корректным критерием выделения области, на которой расположен знак дорожного движения;

- из-за высокой вычислительной нагрузки при использовании нейронной сети, для наиболее эффективного функционирования модуля распознавания участников дорожного движения, существует необходимость реализации переноса вычислений на GPU.

## **5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

### **Введение**

Выпускная квалификационная работа представляет собой разработку программного обеспечения по классификации объектов дорожного движения на фото и видеопоследовательностях. Во время выполнения данной работы были разработаны и запрограммированы алгоритмы классификации объектов дорожного движения по нескольким классам: автомобили, люди, дорожная разметка. Все расчеты и алгоритмы реализованы в среде PyCharm.

Предполагаемое место работы – компьютерный класс Кибернетического центра ТПУ. Основные средства работы – персональный компьютер и локальная вычислительная сеть с выходом в Интернет.

В разделе будут рассмотрены опасные и вредные факторы, оказывающие влияние на производственную деятельность технологического персонала, работающего с автоматизированной системой управления технологическим процессом, рассмотрены воздействия разрабатываемой системы на окружающую среду, правовые и организационные вопросы, а также мероприятия в чрезвычайных ситуациях.

### **5.1 Правовые и организационные вопросы обеспечения безопасности**

Правовые и организационные вопросы задают специфичные требования для выбранного рода деятельности, а именно сфера программирования. Для учета всех требований законодательства по организации работы необходимо провести анализ организационных мероприятий по компоновке рабочей зоны для обеспечения эффективности работы трудящихся.

#### **5.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства**

По ТК РФ №197 – ФЗ работник имеет право на рабочее место, отвечающего требованиям охраны труда, обязательное соц. страхование от несчастных случаев и заболеваний, связанных с производством и профессией,

получение информации от работодателя и гос. органов об условиях труда на рабочем месте и возможных рисках повреждения и утраты здоровья, а также о методах защиты и предотвращение опасных производственных факторов.

Право на отказ от выполнения работ при нарушении требований охраны труда, приводящих к возникновению опасности для жизни или здоровья работника кроме случаев, предусмотренных федеральными законами до ее устранения.

Право на средства индивидуальной защиты во время проведения работ за счет работодателя.

Право на обучение методам работы за счет работодателя.

Право на прямое или косвенное участие в обсуждении вопросов, касающихся безопасных условий труда на рабочем месте и участие в 63 рассмотрении произошедшего с ним несчастного случая или профессионального заболевания.

Право на прохождение мед. осмотра с сохранением средней ЗП и должности на время прохождения.

Право на гарантии и компенсации, установленные договором при работе во вредных и опасных условиях, эти гарантии устанавливаются договором с учетом финансового положения работодателя.

### **5.1.2 Организационные мероприятия при компоновке рабочей зоны**

Требования к организации рабочих мест пользователей:

Рабочее место должно быть организовано с учетом эргономических требований согласно ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» и ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».

Конструкция рабочей мебели (рабочий стол, кресло, подставка для ног) должна обеспечивать возможность индивидуальной регулировки соответственно росту пользователя и создавать удобную позу для работы. Вокруг ПК должно быть обеспечено свободное пространство не менее 60-120см;

На уровне экрана должен быть установлен оригинал-держатель. На рисунке 20 схематично представлены требования к рабочему месту.



Рисунок 20 – Организация рабочего места

Работа программиста связана с постоянной работой за компьютером, следовательно, могут возникать проблемы, связанные со зрением. Также неправильная рабочая поза может оказывать негативное влияние на здоровье. Таким образом, неправильная организация рабочего места может послужить причиной нарушения здоровья и появлением психологических расстройств.

СанПиН 2.2.3670-20 «Санитарно-эпидемиологические требования к условиям труда»:

- яркость дисплея не должна быть слишком низкой или слишком высокой;
- размеры монитора и символов на дисплее должны быть оптимальными;
- цветовые параметры должны быть отрегулированы таким образом, чтобы не возникало утомления глаз и головной боли.

- опоры для рук не должны мешать работе на клавиатуре;
- верхний край монитора должен находиться на одном уровне с глазом, нижний – примерно на 20° ниже уровня глаза;
- дисплей должен находиться на расстоянии 45-60 см от глаз;
- локтевой сустав при работе с клавиатурой нужно держать под углом 90°;
- каждые 10 минут нужно отводить взгляд от дисплея примерно на 5-10 секунд;
- монитор должен иметь антибликовое покрытие;
- работа за компьютером не должна длиться более 6 часов, при этом необходимо каждые 2 часа делать перерывы по 15-20 минут;
- высота стола и рабочего кресла должны быть комфортными.

## **5.2 Производственная безопасность**

### **5.2.1 Анализ вредных и опасных факторов**

В данном подразделе производится анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте при проведении исследований и использовании результатов исследования.

Выявленные факторы представлены в таблице 5.2.1.1

Таблица 5.2.1.1 – Опасные и вредные факторы при проведении сравнительного исследования результатов оценивания учебных достижений

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработ ка	Изготовл ение	Эксплуат ация	
1. Повышенный уровень шума на рабочем месте	+	+	+	Шум. Общие требования безопасности, СН 2.2.4/2.1.8.562–96;
2. Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.3670-20 «Санитарно-эпидемиологические требования к условиям труда» СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»
3. Недостаточная освещенность рабочей зоны	+	+	+	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»
4. Повышенное образование электростатических зарядов	+	+	+	СанПиН 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ»

### 5.2.1.1 Повышенный уровень шума на рабочем месте

На компьютеризированных рабочих местах основными источниками шума являются вентиляторы системного блока, накопители, принтеры ударного действия. Шум создает значительную нагрузку на нервную систему человека, оказывая на него психологическое воздействие.

Уровень звука на рабочих местах, связанных с творческой деятельностью, научной деятельностью, программированием, преподаванием и обучением не должен превышать 50 дБА согласно СН 2.2.4/2.1.8.562–96 [16].

Меры, которые необходимо принять, для того чтобы помещение было менее зашумленным – это обеспечить нормальную вентиляцию системного блока. Для охлаждения необходимо оборудовать со стороны вентиляционных отверстий 20-30 см свободного пространства.

### **5.2.1.2 Отклонение показателей микроклимата рабочей зоны**

Как центральное отопление, так и обогреватели сушат воздух. Пересушенный воздух при критических показателях создает неоспоримую опасность для здоровья человека: способствует возникновению инфекций, обострение аллергических заболеваний и астмы.

В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма.

Оптимальные микроклиматические при воздействии на человека в течение рабочей смены обеспечивают сохранение теплового состояния организма и не вызывают отклонений в состоянии здоровья. Допустимые микроклиматические условия могут приводить к незначительным дискомфортным тепловым ощущениям. Возможно временное (в течение рабочей смены) снижение работоспособности, без нарушения здоровья.

Все категории работ разграничиваются на основе интенсивности энергозатрат организма в ккал/ч (Вт). Работа, производимая сидя и сопровождающаяся незначительным физическим напряжением, относится к категории Ia – работа с интенсивностью энергозатрат до 120 ккал/ч (до 139 Вт). Для данной категории допустимые нормы микроклимата представлены в таблице 5.2.1.2.1.

Таблица 5.2.1.2.1 Допустимые нормы микроклимата в рабочей зоне производственных помещений

Период года	Категория тяжести выполняемых работ	Температура, °С		Относительная влажность, %		Скорость движения воздуха, м/с	
		Фактическое значение	Допустимое значение	Фактическое значение	Допустимое значение	Фактическое значение	Допустимое значение
Холодный	Ia	(20÷24)	(19÷24)		(15÷75)	0.1	≤ 0.1
Теплый	Ia	(23÷25)	(20÷28)		(15÷75)	0.1	≤ 0.2

Анализируя таблицу 5.2.1.2.1, можно сделать вывод, что в рассматриваемом помещении параметры микроклимата соответствуют нормам СанПиН 2.2.548-96 [17]. Допустимый уровень микроклимата помещения обеспечивается системой водяного центрального отопления и естественной вентиляцией.

### 5.2.1.3 Отсутствие или недостаток естественного света, а также недостаточная освещенность рабочей зоны

Соответствующее производственное освещение способствует улучшению условий зрительной работы, как следствие снижает утомляемость и способствует повышению производительности труда.

В компьютерных залах используется комбинированное освещение: искусственное и естественное. Искусственное освещение создается люминесцентными лампами типа ЛД.

Естественное освещение создается прямыми солнечными лучами. Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк согласно СанПиН 1.2.3685-21

«Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [20].

Для оценки использования естественного света введено понятие коэффициента естественной освещенности (КЕО) и установлены минимальные допустимые значения КЕО — это отношение освещенности  $E_v$  внутри помещения за счет естественного света к наружной освещенности  $E_n$  от всей полусферы небосклона, выраженное в процентах:

$$КЕО = \left( \frac{E_v}{E_n} \right) 100\%.$$

При боковом естественном освещении в аудитории, лаборатории на рабочих столах и партах должен обеспечиваться  $КЕО = 1,5 \%$ . К средствам нормализации освещения производственных помещений и рабочих мест относятся: источники света, осветительные приборы, световые проемы, светозащитные устройства, светофильтры

#### **5.2.1.4 Повышенное образование электростатических зарядов**

Электризация заключается в следующем: нейтральные тела, в нормальном состоянии не проявляющие электрических свойств, при условии отрицательных контактов или взаимодействий становятся электростатически заряженными. Опасность возникновения статического электричества проявляется в возможности образования электрической искры и вредном воздействии его на человеческий организм, и не только в случае непосредственного контакта с зарядом, но и за счет действий электрического поля, которое возникает при заряде. При включенном питании компьютера на экране дисплея накапливается статическое электричество. Электрический ток искрового разряда статического электричества мал и не может вызвать поражение человека. Тем не менее, вблизи экрана электризуется пыль и оседает на нем. В результате чего искажается резкость восприятия информации на экране. Кроме того, пыль попадает на лицо работающего и в его дыхательные пути.

Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели, в отсутствии повреждений и наличии заземления приэкранного фильтра.

Основные способы защиты от статического электричества следующие: заземление оборудования, увлажнение окружающего воздуха. Также целесообразно применение полов из антистатического материала.

Чтобы избежать поражения электрическим током, необходимо выполнять правила, указанные в «ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты» [21]

### **5.2.2 Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов**

При работе с ПЭВМ и его периферией существует угроза поражения электрическим током. Электробезопасность представляет собой систему организационных и технических мероприятий и средств, которые обеспечивают защиту людей от вредного и опасного для жизни воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Согласно ПУЭ выполнение ВКР велось в помещении без повышенной опасности, поскольку в нем отсутствовали следующие условия:

сырость или токопроводящая пыль;

токопроводящие полы;

высокая температура;

возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

В ГОСТ 12.1.019-2009 приведены способы и средства защиты, обеспечивающие электробезопасность электроустановок различного назначения.

Согласно данному стандарту, для обеспечения защиты от случайного прикосновения к токоведущим частям, необходимо применять следующие способы и средства технической электробезопасности: защитные оболочки и ограждения, безопасное расположение токоведущих частей, изоляцию токоведущих частей или рабочего места, малое напряжение и защитное отключение.

Для обеспечения защиты от поражения электрическим током при прикосновении к металлическим нетоковедущим частям, которые могут оказаться под напряжением в результате повреждения изоляции, применяют следующие способы: защитное заземление, зануление, выравнивание потенциала, систему защитных проводов, защитное отключение, изоляцию нетоковедущих частей, электрическое разделение сети, малое напряжение, контроль изоляции, компенсацию токов замыкания на землю, средства индивидуальной защиты. Описанные способы и средства применяют отдельно или в сочетании друг с другом так, чтобы обеспечивалась оптимальная защита.

К работе с электроустановками допускаются лица, прошедшие инструктаж и обучение безопасным методам труда, проверку знаний правил безопасности и инструкций в соответствии с занимаемой должностью с присвоением соответствующей квалификационной группы по технике безопасности и не имеющие медицинских противопоказаний.

### **5.3 Экологическая безопасность**

Основным источником вреда для окружающей среды являются отходы, полученные при проведении работ. Их необходимо утилизировать. Так как израсходованная бумага не содержала никаких закрытых сведений, она была направлена на утилизацию без использования shreddera, а люминесцентные лампы собраны и направлены на утилизацию в соответствующую организацию.

Израсходованные картриджи аналогично были переданы производителю для централизованной утилизации в соответствии с требованиями ГОСТ 30775-2001.

Бытовой мусор помещений организаций несортированный, образованный в результате деятельности работников предприятия (код отхода 91200400 01 00 4). Агрегатное состояние отхода твердое; основные компоненты: бумага и древесина, металлы, пластмассы и др. Для сбора мусора рабочее место оснащается урной. При заполнении урны, мусор выносится в контейнер бытовых отходов. Предприятие заключает договор с коммунальным хозяйством по вывозу и размещению мусора на организованных свалках.

Современное, электронное оборудование, даже бытовые приборы, нередко содержит драгметаллы, токсичные и прочие опасные для экологии и здоровья человека вещества. Фактически, порядок утилизации (избавления от) компьютеров и оргтехники производится с учетом двух федеральных законов:

- N 89-ФЗ от 24.06.1998 – «Об отходах производства и потребления»;
- N 41-ФЗ от 26.03.1998 – «О драгоценных металлах и драгоценных камнях».

Утилизация компьютерного оборудования осуществляется по специально разработанной схеме, которая должна соблюдаться в организациях. На первом этапе необходимо создать комиссию, задача которой заключается в принятии решений по списанию морально устаревшей или не рабочей техники, каждый образец рассматривается с технической точки зрения;

- Разрабатывается приказ о списании устройств. Для проведения экспертизы привлекается квалифицированное стороннее лицо или организация;
- Составляется акт утилизации, основанного на результатах технического анализа, который подтверждает негодность оборудования для дальнейшего применения;
- Формируется приказ на утилизацию. Все сопутствующие расходы должны отображаться в бухгалтерии;

- Утилизацию оргтехники обязательно должна осуществлять специализированная фирма;

- Получается специальная официальной формы, которая подтвердит успешность уничтожения электронного мусора.

После оформления всех необходимых документов, компьютерная техника вывозится со склада на перерабатывающую фабрику. Все полученные в ходе переработки материалы вторично используются в различных производственных процессах (ГОСТ 30494-2011, Здания жилые и общественные. Параметры микроклимата в помещениях, 2011)

## **5.4 Безопасность в чрезвычайных ситуациях**

### **5.4.1 Анализ вероятных ЧС**

Наиболее вероятной ЧС в рамках рассматриваемого помещения является пожар. Помещение, в котором велась работа по степени пожаробезопасности относится к категории Г (умеренная пожароопасность), т.е. к помещению, в котором находятся негорючие вещества и материалы в горячем, раскаленном или расплавленном состоянии, процесс обработки которых сопровождается выделением лучистого тепла, искр и пламени (СП 12.13130.2009).

Возникновение пожара может возникнуть от следующих источников воспламенения:

- 1) искра при разряде статистического электричества;
- 2) искра от электрооборудования;

Также на рабочем месте запрещается иметь огнеопасные вещества и выполнять следующие действия [22]:

- 1) курить;
- 2) зажигать огонь;
- 3) включать электрооборудование, если в помещении пахнет газом;
- 4) сушить что-либо на отопительных приборах;
- 5) закрывать вентиляционные отверстия в электроаппаратуре.

#### **5.4.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС**

К мерам по предупреждению пожара относятся следующие пожарно-профилактические мероприятия:

- 1) соблюдение эксплуатационных норм оборудования;
- 2) обучение персонала правилам техники безопасности;
- 3) издание противопожарных инструкций, планов эвакуации.

Основными мерами по повышению устойчивости помещения к данной ЧС являются в первую очередь исключение образования благоприятной для пожара среды (контроль воздухообмена), а также использование трудно сгораемых материалов при отделке рабочего помещения (ГОСТ 12.1.004-91 ССБТ).

Необходимо предусмотреть безопасную эвакуацию людей на случай возникновения пожара. При пожаре люди должны покинуть помещение в течение минимального времени. Помещение, в котором выполнялась работа,

входит в общий план эвакуации этажа, который предусматривает выход из всех помещений этажа в основной или запасной эвакуационные выходы здания. Эвакуация проводится согласно плану эвакуации, который выставлен на всеобщее обозрение в нескольких местах на каждом этаже.

В каждом кабинете установлен углекислотный огнетушитель ОУ-2 и табличка с указанием лица, ответственного за пожарную безопасность.

ПЛАН  
эвакуации сотрудников и студентов  
кафедры ПМ ИК в случае пожара в корпусе ИК - 1 этаж

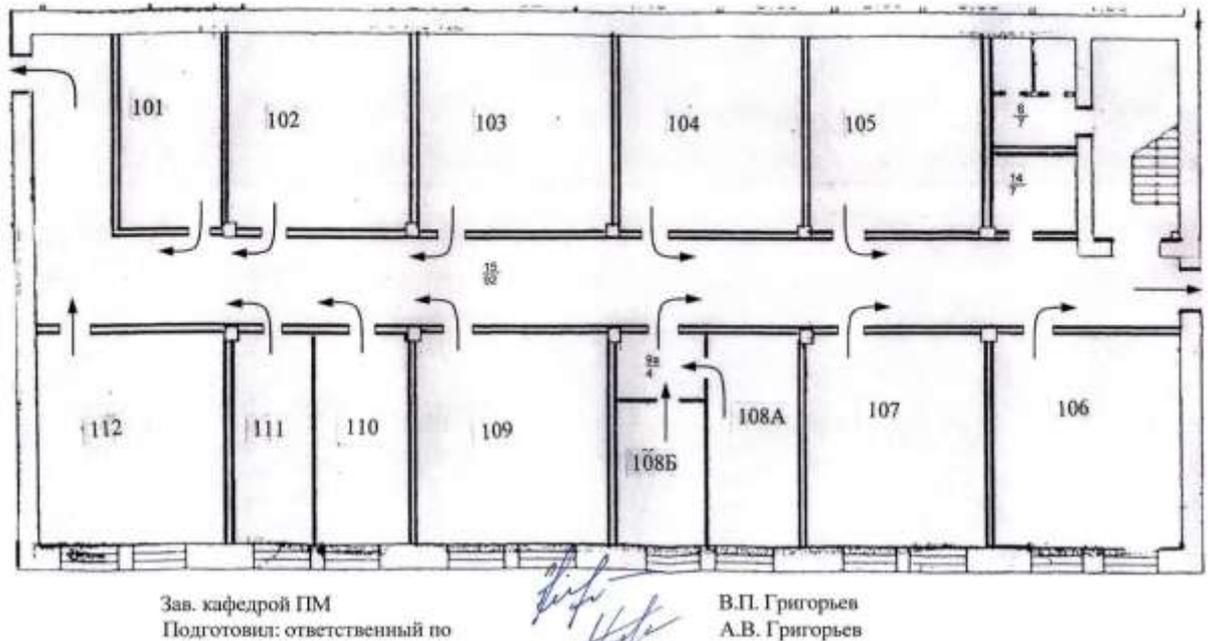


Рисунок 10.4.2.1 – План эвакуации

Необходимыми действиями в результате возникшей ЧС и мерами по ликвидации её последствий являются:

1. Передать сигнал «Тревога» голосом, задействовать систему оповещения людей о пожаре.
2. Сообщить по телефону 01, с сотового 010 адрес объекта, место возникновения пожара, свою фамилию. Сообщить по телефону 03, с сотового 030 адрес объекта, что случилось, информацию о пострадавших, свою фамилию, оказать помощь пострадавшим.
3. Открыть все эвакуационные выходы, направить людей к эвакуационным выходам согласно знакам направления движения.
4. Отключить от электропитания оборудование, механизмы и т.п., обесточить помещение.
5. По возможности принять меры по тушению пожара используя средства противопожарной защиты.

По возможности предотвратить развитие аварии, обозначить место аварии.

## **Выводы и рекомендации**

Проанализировав условия труда на рабочем месте, где была разработана магистерская работа, можно сделать вывод, что помещение удовлетворяет необходимым нормам и в случае соблюдения техники безопасности и правил пользования компьютером работа в данном помещении не приведет к ухудшению здоровья работника. Само помещение и рабочее место в нем удовлетворяет всем нормативным требованиям. Кроме того, действие вредных и опасных факторов сведено к минимуму, т.е. микроклимат, освещение и электробезопасность соответствуют требованиям, предъявленным в соответствующих нормативных документах.

Работу с ЭВМ можно отнести к экологически безопасным видам деятельности, если должным образом утилизировать отходы данной деятельности.

Относительно рассмотренного вопроса об экологической безопасности можно сказать, что деятельность в ходе выполнения выпускной квалификационной работы не представляет опасности окружающей среды

## **6 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

Целью написания раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

### **6.1 Оценка коммерческого и инновационного потенциала научных исследований**

#### **6.1.1 Потенциальные потребители результатов исследований**

Выполненная работа направлена на разработку и реализацию алгоритма классификации объектов на изображениях и видеопоследовательностях, который будет распознавать объекты дорожного движения на записях с различных видеорегистраторов.

Целевой рынок – сегменты рынка, на котором будет продаваться в будущем разработка. В свою очередь, сегмент рынка – это особым образом выделенная часть рынка, группы потребителей, обладающих определенными общими признаками.

Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар (услуга). Можно применять географический, демографический, поведенческий и иные критерии сегментирования рынка потребителей, возможно применение их комбинаций с использованием таких характеристик, как возраст, пол, национальность, образование, любимые занятия, стиль жизни, социальная принадлежность, профессия, уровень дохода.

В зависимости от категории потребителей (коммерческие организации, физические лица) необходимо использовать соответствующие критерии сегментирования. Например, для коммерческих организаций критериями сегментирования могут быть: месторасположение; отрасль; выпускаемая продукция; размер и др.

Потенциальными потребителями результатов исследования являются крупные организации, специализирующиеся в нефтегазовой отрасли, а именно газодобывающие компании, которые имеют в своём распоряжении уже эксплуатируемые УКПГ или планируют внедрение новых установок подготовки газа. Для этих компаний разрабатывается автоматизированная система управления узла осушки газа установки комплексной подготовки газа.

### 6.1.2 Анализ конкурентных технических решений

Анализ конкурентных технических решений проводится с помощью оценочной карты для сравнения конкурентных технических решений, приведенной в таблице 6.1.2.1. Основным средством разработки, используемым в данной работе, является библиотека компьютерного зрения OpenCV. Конкурентные средства разработки: SimpleCV, Face Recognition.

Таблица 6.1.2.1 Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы			Конкурентоспособность		
		Б <sub>очн</sub>	Б <sub>scv</sub>	Б <sub>fr</sub>	Б <sub>очн</sub>	Б <sub>scv</sub>	Б <sub>fr</sub>
Технические критерии оценки ресурсоэффективности							
Эффективность	0,3	4	4	4	1,2	1,2	1,2
Устойчивость	0,2	5	5	5	1	1	1
Временные затраты	0,2	5	4	4	1	0,8	0,8
Новизна метода	0,1	5	3	3	0,5	0,3	0,3
Простота реализации	0,1	5	4	5	0,5	0,4	0,5
Универсальность	0,1	4	4	4	0,5	0,4	0,4
Итого	1	29	24	25	4,7	4,1	4,2

По полученным результатам можно сделать вывод, что разрабатываемый алгоритм для оценки информативности является по конкурентоспособности наиболее эффективным.

### 6.1.3 SWOT-анализ

SWOT-анализ – это эффективный инструмент стратегического менеджмента. SWOT-анализ заключается в исследовании внешней и внутренней сред проекта.

метод, которые заключается в комплексном анализе научно-исследовательского проекта. SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (Угрозы).

Разработанная для алгоритма поиска информативного атрибута матрица SWOT представлена в таблице 6.1.3.1.

Таблица 6.1.3.1 SWOT-анализ

	<b>Сильные стороны:</b>	<b>Слабые стороны:</b>
	<p>С1. Открытый код использованных средств разработки.</p> <p>С2. Гибкость алгоритма, в области модификаций.</p> <p>С3. Простая реализация.</p>	<p>Сл1. Высокая вычислительная нагрузка на систему.</p> <p>Сл2. Прямая зависимость эффективности от качества обучения модели.</p>
<p><b>Возможности:</b></p> <p>В1. Дообучение моделей.</p> <p>В2. Перенос вычислений на GPU.</p>	<p>Благодаря гибкости и открытому исходному коду модели могут быть дополнены разнообразными решениями.</p>	<p>Перенос вычислительной нагрузки на GPU позволит снизить минимальные требования к характеристикам оборудования, тем самым расширив целевую аудиторию.</p>
<p><b>Угрозы:</b></p> <p>У1. Отсутствие спроса на продукт на рынке.</p> <p>У2. Развитие и появление аналогов алгоритма.</p>	<p>Наглядные результаты использования алгоритма, а также его низкая стоимость могут увеличить спрос на него.</p>	<p>Разработка распараллеленного алгоритма для повышения оптимизации вычислений.</p> <p>Развитие детекторов, не использующих модели.</p> <p>Разработка/модификация моделей детектирования, использующих меньшие выборки для обучения.</p> <p>Создание пользовательского интерфейса для детальной настройки детекторов в процессе их непосредственного использования.</p>

В процессе проведения SWOT-анализа было выявлено, что основные слабые стороны проекта напрямую связаны с его производительностью, а потенциальные угрозы в большей степени зависят от поведения рынка. Однако гибкость проекта, достигаемая за счёт его модульности и открытого кода используемых библиотек,

позволяет без лишних потерь модифицировать функционал разработанной системы на протяжении всего её жизненного цикла.

## 6.2 Планирование научно-исследовательских работ

### 6.2.1 Структура работ в рамках научного исследования

Трудоемкость выпускной квалификационной работы определяется опытным путем в человеко-днях и имеет вероятностный характер, так как зависит от трудно учитываемых факторов. Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

В качестве исполнителей в проекте представлены руководитель и инженер.

Перечень этапов, работ и исполнителей приведен в таблице 6.2.1.1.

Таблица 6.2.1.1 Основные этапы выполнения ВКР

№	Описание этапа	Исполнитель и	Загруженность исполнителей
1	Составление и утверждение задания	НР	НР – 100%
2	Анализ предметной области	С, НР	С – 50%, НР – 50%
3	Разработка календарного плана	С, НР	С – 10%, НР – 90%
4	Обзор литературы и интернет - источников	С, НР	С – 50%, НР – 50%
5	Обзор задач и методов реализации алгоритма классификации объектов на видеопоследовательностях	С	С – 100%
6	Постановка задачи алгоритмизации	С, НР	С – 10%, НР – 90%
7	Выбор метода решения поставленной задачи	С, НР	С – 10%, НР – 90%

Таблица 6.2.1.1 (продолжение) Основные этапы выполнения ВКР

8	Разработка алгоритмов решения задачи	С	С – 100%
9	Программная реализация компьютерной модели	С	С – 100%
10	Проведение исследований на модельных данных	С, НР	С – 80% НР – 20%
11	Проведение исследований на реальных данных	С, НР	С – 60% НР – 40%
12	Расчет экономической эффективности научно-технической продукции	С	С – 100%
13	Оценка социальной ответственности проекта	С	С – 100%
14	Написание пояснительной записки	С, НР	С – 90% НР – 10%

### 6.2.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости используется следующая формула:

$$t_{\text{ож}} = \frac{3 \cdot t_{\text{min}} + 2 \cdot t_{\text{max}}}{5}, \quad (6.2.2.1)$$

где  $t_{\text{min}}$  – предположительно минимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки;

$t_{\text{max}}$  – предположительно максимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях  $T_p$ , учитывающая параллельность выполнения работ несколькими исполнителями.

$$T_{p_i} = \frac{t_{\text{ож}_i}}{ч_i}, \quad (6.2.2.2)$$

где  $T_{p_i}$  – продолжительность одной работы, раб. дн.;

$t_{\text{ож}_i}$  – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

$Ч_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Продолжительность каждого этапа рассчитывается по формуле:

$$t_{\text{раб}} = t_{\text{ож}} \cdot k_{\text{д}}, \quad (6.2.2.3)$$

где  $t_{\text{раб}}$  – длительность этапов в рабочих днях;

$k_{\text{д}}$  – коэффициент, учитывающий дополнительное время на консультации и согласование работ,  $k_{\text{д}} = 1.2$ .

### 6.2.3 Разработка графика проведения научного исследования

В данном пункте необходимо построить диаграмму Ганта – горизонтально-ленточный график, на котором работы представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения приведенных работ.

Линейный график строится на основании полученных значений  $t_{\text{раб}}$ , предварительно переведенных в календарные дни по формуле:

$$t_{\text{к}} = t_{\text{раб}} \cdot K_{\text{н}}, \quad (6.2.3.1)$$

где  $t_{\text{к}}$  – длительность этапов работ в календарных днях;

$K_{\text{н}}$  – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$K_{\text{н}} = \frac{T_{\text{к}}}{T_{\text{к}} - T_{\text{вд}} - T_{\text{пд}}}, \quad (6.2.3.2)$$

где  $T_{\text{к}}$  – количество календарных дней,  $T_{\text{к}} = 365$ ;

$T_{\text{вд}}$  – количество выходных дней,  $T_{\text{вд}} = 108$ ;

$T_{\text{пд}}$  – количество праздничных дней,  $T_{\text{пд}} = 10$ ;

$$K_{\text{н}} = \frac{365}{365 - 108 - 10} = 1.478 \quad (6.2.3.3)$$

Все расчеты сведены в таблицу 9.2.3.1.

Таблица 6.2.3.1 Временные показатели проведенных работ

№	Исполнители	Продолжительность работ в днях			Трудоемкость	
		$t_{min}$	$t_{max}$	$t_{ож}$	$t_{раб}$	$t_k$
1	НР	1	2	1,4	1,68	2,48
2	С, НР	2	3	2,4	2,88	4,26
3	С, НР	1	2	1,4	1,68	2,48
4	С, НР	3	5	3,8	4,56	6,74
5	С	2	5	3,2	3,84	5,68
6	С, НР	3	6	4,2	5,04	7,45
7	С, НР	5	8	6,2	7,44	11,00
8	С	3	6	4,2	5,04	7,45
9	С	4	6	4,8	5,76	8,51
10	С, НР	2	5	3,2	3,84	5,68
11	С, НР	2	4	2,8	3,36	4,97
12	С	5	7	5,8	6,96	10,29
13	С	4	5	4,4	5,28	7,80
14	С, НР	9	12	10,2	12,24	18,09
Итого				58	69,6	102,87

На основании таблицы 6 построим диаграмму Ганта (рисунок 6.2.3.1).



■ – инженер      ■ – руководитель

Рисунок 6.2.3.1 – Диаграмма Ганта

### **6.3 Бюджет научно-технического исследования (НТИ)**

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

- материальные затраты НТИ;
- затраты на специальное оборудование для научных (экспериментальных) работ;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- затраты научные и производственные командировки;
- контрагентные расходы;
- накладные расходы.

В нашем случае расчет основных затрат производится только для научного руководителя, так как исполнитель является студентом, и, следовательно, со стороны ТПУ него нет затрат.

#### **6.3.1 Расчет затрат на специальное оборудование для научных (экспериментальных) работ**

В Данная статья отражает стоимость всех материалов, используемых при разработке проекта, включая расходы на их приобретение и доставку. Транспортные расходы принимаются в пределах 3-5% от стоимости материалов. В материальные затраты, помимо вышеуказанных, включаются дополнительно затраты на канцелярские принадлежности, диски, картриджи и т.п. Однако их учет ведется в данной статье только в том случае, если в научной организации их не включают в расходы на использование оборудования или накладные расходы.

Расчет затрат на материалы производится по форме, приведенной в таблице 6.3.1.1.

Таблица 6.3.1.1. Материальные затраты

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, руб.
Ноутбук Dexp Atlas H115	Шт	1	34000	34000
Microsoft Windows 10 Professional RU x32/x64	Шт	1	9000	9000
Пакет Microsoft Office 2019 Home and Student RU x32/x64	Шт	1	4600	4600
Электрoэнергия	кВт	102,3	6,59	674,157
Канцелярские принадлежности	Шт	1	300	300
Итого				48574,157

### 6.3.2 Основная заработная плата исполнителей темы

Основная заработная плата руководителя рассчитывается на основании отраслевой оплаты труда (оклад, стимулирующие выплаты, районный коэффициент). Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20 –30 % от тарифа или оклада. Для студента-дипломника основную заработную плату составляет государственная стипендия с учетом районного коэффициента.

Для расчета основной заработной платы (таблица 6.4.2.2) необходимо привести действительный годовой фонд рабочего времени руководителя и студента (табл. 6.3.2.1).

Таблица 6.3.2.1 Баланс времени

Показатели рабочего времени	Руководитель	Студент-дипломник
Календарное число дней	365	365
Количество нерабочих дней (выходные дни и праздничные дни)	52 и 10	108 и 10
Потери рабочего времени (отпуск, больничные), дни	56	24
Действительный годовой фонд рабочего времени, дни	247	223

Статья включает основную заработную плату работников, непосредственно занятых выполнением НИИ, и дополнительную заработную плату:

$$Z_{зп} = Z_{осн} + Z_{доп}, \quad (6.3.2.1)$$

где  $Z_{осн}$  – основная заработная плата;

$Z_{доп}$  – дополнительная заработная плата (10% от основной).

Основная заработная плата руководителя (лаборанта, инженера) от предприятия рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p \quad (6.3.2.2)$$

где  $Z_{осн}$  – основная заработная плата одного работника;

$Z_{дн}$  – среднедневная заработная плата работника, руб.;

$T_p$  – продолжительность работ, выполняемых работником, раб.дн.

Среднедневная зарплата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m \cdot M}{F_d}, \quad (6.3.2.3)$$

где  $Z_m$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года:

при отпуске в 56 раб. дней  $M = 10,4$  месяца, 6-дневная неделя;

$F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн., равный 251.

Месячный должностной оклад работника:

$$Z_m = Z_{тс} \cdot k_p, \quad (6.3.2.4)$$

где  $Z_{тс}$  – заработная плата по тарифной ставке, руб.;

$k_p$  – районный коэффициент, равный 1,3 для Томска.

Расчет основной платы представлен в таблице 6.3.2.2.

Таблица 6.3.2.2 Расчет основной заработной платы

Исполнитель	Тарифная заработная плата, руб.	Районный коэффициент	Месячный должностной оклад работника, руб.	Среднедневная заработная плата, руб.	Продолжительность работ, дни	Заработная плата основная, руб.
Руководитель	35111,50	1,3	45644,95	1923,48	14	26934,18
Инженер	22695,68	1,3	29504,50	1495,24	52	77752,73
<b>Итого:</b>						104686,91

### 6.3.3 Дополнительная заработная плата

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}} = 0,15 \cdot 26934,18 = 4040,13 \quad (6.3.3.1)$$

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}} = 0,15 \cdot 77752,73 = 11662,91 \quad (6.3.3.2)$$

где  $Z_{\text{осн}}$  – основная заработная плата;

$k_{\text{доп}}$  – коэффициент дополнительной заработной платы на стадии проектирования, принимается равным 0,15.

### 6.3.4 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органами

государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}), \quad (6.3.4.1)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Для учреждений, осуществляющих образовательную и научную деятельность в 2019г., водится пониженная ставка 27,1% (п. 6 ч. 1 ст. 58 Закона 212-ФЗ).

Расчет отчислений во внебюджетные фонды приведен в таблице 6.3.4.1.

Таблица 6.3.4.1 Отчисления во внебюджетные фонды

Исполнитель	Руководитель	Инженер
Основная заработная плата, руб.	26934,18	77752,73
Дополнительная заработная плата, руб.	4040,13	11662,91
Коэффициент отчислений во внебюджетные фонды	27,1	
Сумма отчислений	8394,04	24231,64
Итого	32625,68	

### 6.3.5 Прочие прямые затраты

К данному виду затрат относятся затраты на электроэнергию. Для юридических лиц стоимость 1 кВт·ч составляет 5,8 рублей. При умеренном пользовании ноутбук средней мощности потребляет 100 Вт в час в среднем. В день на работу затрачивается 6 часов, всего на работу с компьютером и оборудованием затрачивается 34 дня у инженера и 9 дней у руководителя. Тогда затраты на электроэнергию составят:

$$Z_{\text{эн}} = 100 \cdot \frac{5,8}{1000} \cdot 6 \cdot 43 = 149,64 \text{ руб.} \quad (6.3.5.1)$$

### 6.3.6 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергия, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = k_{\text{накл}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}), \quad (6.3.6.1)$$

где  $k_{\text{накл}}$  – коэффициент накладных расходов, 16 %.

Получим:

$$Z_{\text{накл}} = 0,16 \cdot (26934,18 + 4040,13) = 8723,81 \text{ руб.} \quad (6.3.6.2)$$

$$Z_{\text{накл}} = 0,16 \cdot (77752,73 + 11662,91) = 14306,50 \text{ руб.} \quad (6.3.6.3)$$

### 6.3.7 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект приведено в таблице 6.4.7.1.

Таблица 6.3.7.1 Расчет бюджета затрат

Наименование статьи	Сумма, руб.
1. Затраты на специальное оборудование	16 000
2. Затраты по основной заработной плате исполнителей проекта	26934,18
3. Затраты по дополнительной заработной плате исполнителей проекта	7541,57
4. Отчисления во внебюджетные фонды	16864
5. Накладные расходы	14 895
<b>Бюджет затрат НИИ</b>	<b>107 989</b>

#### 6.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп}i} = \frac{\Phi_{p_i}}{\Phi_{\text{max}}}, \quad (6.4.1)$$

где  $I_{\text{финр}}^{\text{исп}i}$  – интегральный финансовый показатель разработки;

$\Phi_{p_i}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта.

$\Phi_{\text{max}}$  зависит от сложности проекта для которого разрабатывается ПО.

Т.к. стоимость всех вариантов исполнения одинакова, интегральные финансовые показатели также будут одинаковы и равны 1.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{p_i} = \sum a_i \cdot b_i, \quad (6.4.2)$$

где  $I_{p_i}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта исполнения разработки;

$a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки;

$b_i^a, b_i^p$  – балльная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания.

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формулам:

$$I_{\text{исп}_1} = \frac{I_{p-\text{исп}_1}}{I_{\text{фин}p}^{\text{исп}_1}}$$

$$I_{\text{исп}_2} = \frac{I_{p-\text{исп}_2}}{I_{\text{фин}p}^{\text{исп}_2}}$$
(6.4.3)

Так как интегральные финансовые показатели одинаковы и равны 1, то интегральные показатели эффективности вариантов исполнения разработки равны соответствующим интегральным показателям ресурсоэффективности.

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп}_i}}{I_{\text{исп}_1}}$$
(6.4.4)

В пункте 6.2 было рассмотрено два варианта исполнения алгоритма. На основании этого необходимо провести сравнительную характеристику вариантов исполнения (таблица 6.5.1).

Таблица 6.5.1 Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент	Исп.1	Исп.2
1. Повышение производительности труда	0,3	5	5
2. Удобство в эксплуатации	0,2	4	3
3. Удобство в считывании исходных данных	0,2	5	2
4. Скорость работы	0,1	4	3
5. Простота эксплуатации	0,1	4	3
6. Техническая поддержка платформы	0,1	5	3
$I_{p_i}$		4,6	3,4

На основании полученных показателей выполним сравнение интегрального показателя эффективности вариантов исполнения разработки (табл. 6.4.2).

Таблица 6.4.2 Сравнительная эффективность разработки

Показатели	Исп.1	Исп.2
------------	-------	-------

Интегральный финансовый показатель разработки	1	1
Интегральный показатель ресурсоэффективности разработки	4,6	3,4
Интегральный показатель эффективности	4,6	3,4
Сравнительная эффективность вариантов исполнения	1	0,74

С позиции финансовой и ресурсной эффективности на основании таблицы, первый вариант исполнения системы наиболее выгодный. Данный вариант исполнения и используется в выпускной квалификационной работе.

## ЗАКЛЮЧЕНИЕ

В рамках выполнения задания выпускной квалификационной работы были получены следующие результаты:

- проведён обзор литературы в области систем распознавания объектов и компьютерного зрения;
- сформулирована актуальность решаемой задачи;
- сформулирована техническая постановка задачи;
- произведена алгоритмизация блоков для реализации системы распознавания объектов дорожного движения;
- произведён анализ технических средств разработки, выбрана наиболее оптимальная среда для реализации системы;
- разработан модуль детектирования дорожной разметки;
- разработан модуль детектирования и классификации светофоров;
- разработан модуль классификации знаков дорожного движения;
- разработан модуль детектирования и классификации участников дорожного движения.
- проведено тестирование корректности работы программы на тестовых данных.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. How OpenCV-Python Bindings Works? / Home – OpenCV – Электронный ресурс. – Режим доступа [https://docs.opencv.org/master/da/d49/tutorial\\_py\\_bindings\\_basics.html](https://docs.opencv.org/master/da/d49/tutorial_py_bindings_basics.html).
2. Чистый Python против NumPy и TensorFlow. Сравнение производительности / DIGITAL EXPERIENCE WORKSHOP – Электронный ресурс. – Режим доступа <https://chel-center.ru/python-yfc/2020/12/21/chistyj-python-protiv-numpy-i-tensorflow-sravnenie-proizvoditelnosti/>.
3. Распознавание изображений в Python с помощью TensorFlow и Keras / pythobyte.com – Электронный ресурс. – Режим доступа <https://pythobyte.com/image-recognition-in-python-with-tensorflow-and-keras-20d8f085/>.
4. М. В. Гордиенко Исследование эффективности свёрточных нейронных сетей в задачах классификации визуальных // Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева. – 2020. – С. 31.
5. Hough Line Transform / Home – OpenCV – Электронный ресурс. – Режим доступа [https://docs.opencv.org/3.4/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html).
6. Berthold К.Р. Horn. Robot Vision — MIT Press, 1986. (Б.К.П. Хорн, Зрение роботов: перевод с англ. — М.: Мир, 1989).
7. Л. Шапиро, Дж. Стокман. Компьютерное зрение — М.: Бином. Лаборатория знаний, 2006. — 752 с.
8. Желтов С.Ю. и др. Обработка и анализ изображений в задачах машинного зрения. — М.: Физматкнига, 2010. — 672 с.
9. А.А. Лукьяница ,А.Г. Шишкин. Цифровая обработка видеоизображений. — М.: «Ай-Эс-Эс Пресс», 2009. — 518 с.
10. Нильсон Н. Искусственный интеллект. — М.: Мир, 1973. — 273 с.
11. Николенко С., Кадуриин А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2018. — 480 с.

12. Каллан Р. Основные концепции нейронных сетей. — М.: Вильямс, 2001. — 288 с.
13. Ясницкий Л. Н. Введение в искусственный интеллект. — М.: Издат. центр «Академия», 2005. — 176 с.
14. Беркинблит М. Б. Нейронные сети. — М.: МИРОС и ВЗМШ РАО, 1993. — 96 с.
15. А.А. Лукьяница ,А.Г. Шишкин. Цифровая обработка видеоизображений. — М.: «Ай-Эс-Эс Пресс», 2009. — 518 с.
16. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.
17. СанПиН 2.2.548-96. Гигиенические требования к микроклимату производственных помещений.
18. СанПиН 2.2.3670-20 Санитарно-эпидемиологические требования к условиям труда.
19. Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы. Выписки из СанПиН 2.2.2.542-96.
20. СНиП 23-05-95\* Естественное и искусственное освещение.
21. ГОСТ 12.1.019 (с изм. №1) ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты.
22. Федеральный закон Российской Федерации от 22 июля 2008 г. N 123-ФЗ "Технический регламент о требованиях пожарной безопасности".
23. Федеральный классификационный каталог отходов [Электронный ресурс] – 2013. – Режим доступа: <http://www.ecoguild.ru/>, свободный. – Загл. с экрана.
24. Об утверждении правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде: Постановление Правительства Российской Федерации от 3 сентября 2010 года № 681.

25. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования
26. ГОСТ 12.2.061-81 ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам
27. СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

## СПИСОК ПУБЛИКАЦИЙ

1. Кружков Д. С. Разработка программного обеспечения распознавания дорожной полосы на видеопоследовательностях / Д. С. Кружков // XVIII Международная научно-практическая конференция студентов, аспирантов и молодых ученых МСИТ-2021.

## ПРИЛОЖЕНИЕ А

(обязательное)

Раздел ВКР, выполненный на иностранном языке

### Раздел 3 Software engineering for traffic object classification system

Студент:

Группа	ФИО	Подпись	Дата
8ВМ91	Кружков Денис Сергеевич		

Руководитель ВКР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Друки Алексей Алексеевич	К.Т.Н.		

Консультант – лингвист отделения (НОЦ) школы ОИЯ, ШБИП:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель	Ажель Юлия Петровна	-		

### **3 SOFTWARE ENGINEERING FOR TRAFFIC OBJECT CLASSIFICATION SYSTEM**

The interpreted high-level general-purpose programming language Python was chosen as a programming language for engineering the system for traffic object classification.

The advantages of this programming language are:

- a high level of code readability provides a comfortable level of concentration on syntax;
- Python is free, this is the reason for the rapid modernization of the language, and support from the community;
- compatibility with a large number of platforms let us avoid the cross-platform issues common to most programming languages;
- support for object-oriented programming allows using objects based on data and functionality;
- a huge base of libraries created or converted for Python make it easier to work with any type of data.

Table 3.1 Comparative characteristics of C++/Java/Python

Topic	C++11	Java 8	Python 3.5
Memory management	Manual allocation Manual deallocation	Manual allocation Automatic garbage collection	Automatic allocation Automatic deallocation
Code length	Longer lines of code as compared to Python	About 40% more code in comparison to C++	3-5 times shorter than Java programs
Typing system	Statically typed, both code and compiler	Statically typed, both code and compiler	Dynamically typed, Interpreter is strongly typed, language is typed-free
Compiling	Compiled to machine code	Compiles to byte code, then interpreted by the JVM	Interpreted by the Python interpreter
Execution speed (loops, dynamic arrays with numbers, basic math operations)	100%	1176%	1800%
Portability	Platform dependent, must be recompiled for different platform	Platform independent, byte code generated works on every OS	Platform independent, code is interpreted upon execution
Thread support	Built-in support since C++11	It has built-in thread support	No multithreading
Syntax complexity	Defining blocks using {, ending statements using }	Defining blocks using {, ending statements using }	Using indentation only

### 3.1 Road marking detector

Road marking is one of the most easily detectable objects due to its uniform shape and low degree of variability in various images. The detection of this object can be programmed using standard image analysis functions, without an unique detection model.

The road marking detector was created using the basic image processing methods of the «OpenCV» computer vision library.

#### 3.1.1 Canny Boundary Detector

Before proceeding to the detection of borders, we need to make a preliminary preparation of the original image.

The first stage of preprocessing is the transition from the BGR color space to the Grayscale. We can do this by using the «cvtColor» function, with the input parameter "COLOR\_BGR2GRAY" (figure 3.1.1.2).

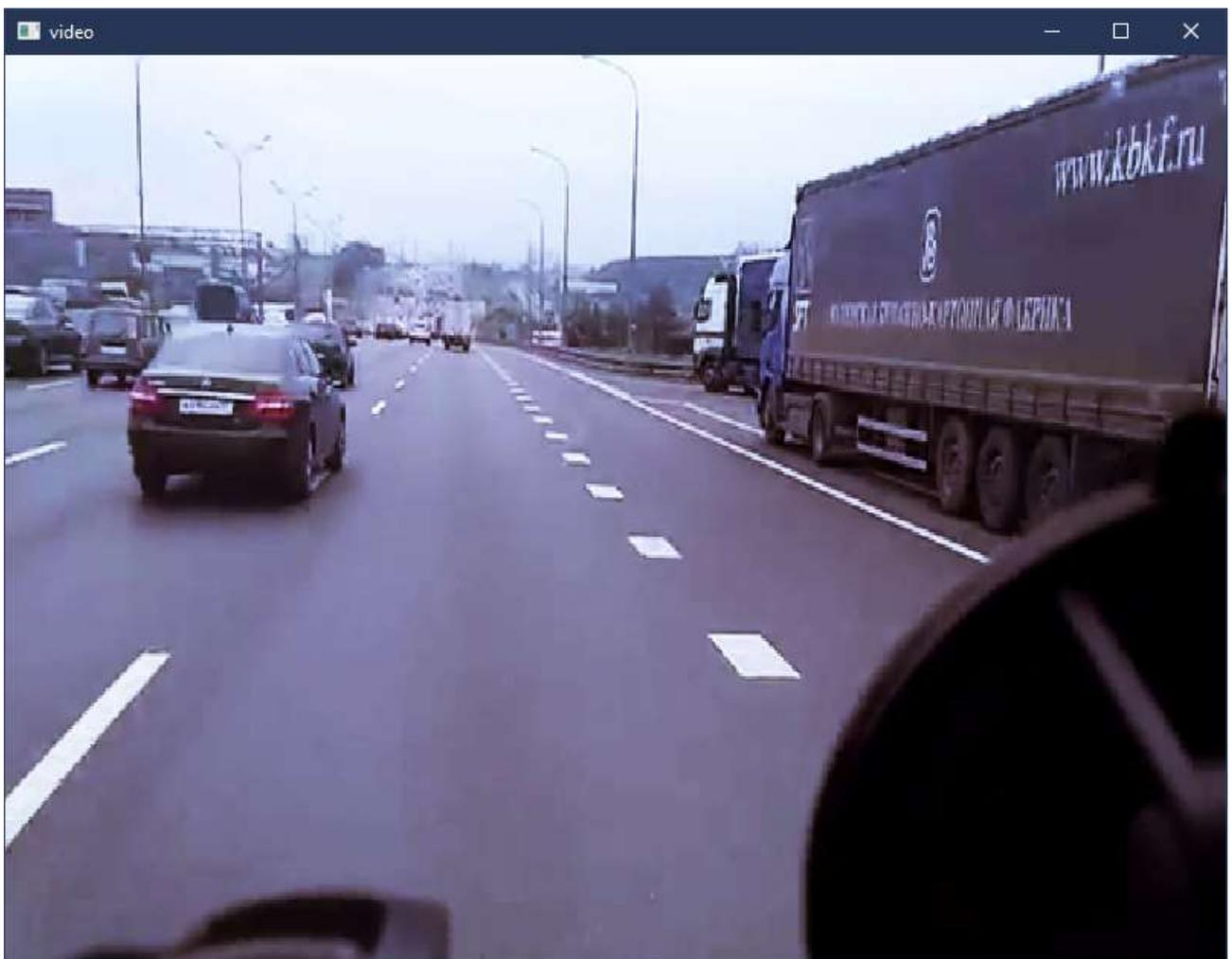


Figure 3.1.1.1 Frame of the original video



Figure 3.1.1.2 Result of transition to the Grayscale color space

To apply a Gaussian blur filter, use the «GaussianBlur» function (figure 3.1.1.3).



Figure 3.1.1.3 Applying a «GaussianBlur» filter

The prepared image is sent to the input of the «Canny» function, which is a direct implementation of the boundary detector for the «OpenCV» library.

The process of the "Canny" boundary detector includes five stages of image processing:

- smoothing the image with a Gaussian filter to suppress noise in the input image;
- search for image intensity gradients;
- use of the "Non-maximum suppression" technique to avoid a false response to edge detection;
  - determination of the potential edge using a double threshold;
  - completion of border detection, suppression of sharp edges that are weak and not related to strong edges (hysteresis).

Using the "Canny" boundary detector, we get the original image, represented by a set of contours (figure 3.1.1.4).



Figure 3.1.1.4 Result of using the boundary detector «Canny»

### 3.1.2 Search for the region of interest

In the case of image analysis, when the search object is a road marking line, the area of interest is an elongated trapezoid along the direction of the road.

The first step in forming a mask for the region of interest is to create an array of coordinates of the vertices of the trapezoid.

After correctly forming the points that comprise the trapezoid, you need to create a shape. To do this, the «zeros\_like» function of the «numpy» library creates an array of zeros with a shape and data type similar to the original image, and uses «fillPoly» to fill the area formed by the array of vertex coordinates. The resulting shape is superimposed as a mask on the original image using the bitwise function "And" for each pixel, thus cutting off those parts of the image that are outside the area of interest represented by the trapezoid (figure 3.1.2.1).



Figure 3.1.2.1 Result of applying a mask to cut off the region of interest

### 3.1.3 Calculating and drawing guides

We use the «HoughLinesP» function of the «OpenCV» library to calculate the road marking guides. This is an implementation of the algorithm for finding lines in the image.

The algorithm of this function is based on the Hough line transformation. Line in the image space can be expressed with two variables (for example  $m$  and  $b$  – in the Cartesian coordinate system,  $r$  and  $\theta$  – in the Polar coordinate system, figure. 3.1.3.1).

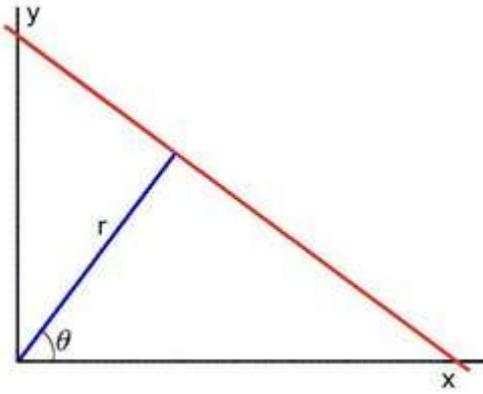


Figure 3.1.3.1 Lines in Cartesian and Polar coordinate systems

For Hough Transforms, we will express lines in the Polar system. Hence, a line equation can be written as:

$$y = \left(-\frac{\cos \theta}{\sin \theta}\right) x + \left(\frac{r}{\sin \theta}\right), \quad (3.1.3.1)$$

arranging the terms:

$$r = x \cos \theta + y \sin \theta. \quad (3.1.3.2)$$

In general, for each point  $(x_0, y_0)$ , we can define the family of lines that goes through that point as:

$$r_\theta = x_0 \cos \theta + y_0 \sin \theta. \quad (3.1.3.3)$$

Meaning that each pair  $(r_\theta, \theta)$  represents each line that passes by  $(x_0, y_0)$ .

If for a given  $(x_0, y_0)$  we plot the family of lines that goes through it, we get a sinusoid (figure 3.1.3.2).

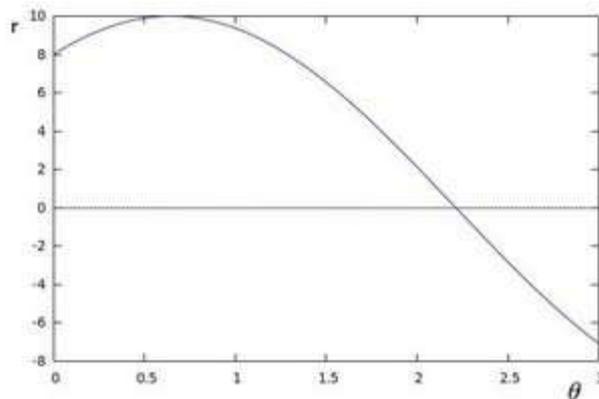


Figure 3.1.3.2 Sinusoid in space  $\theta - r$

We consider only such points as  $r > 0$  and  $0 < \theta < 2\pi$ .

We can do the same operation above for all the points in an image. If the curves of two different points intersect in the plane  $\theta - r$ , that means that both points belong to a same line (figure 3.1.3.3).

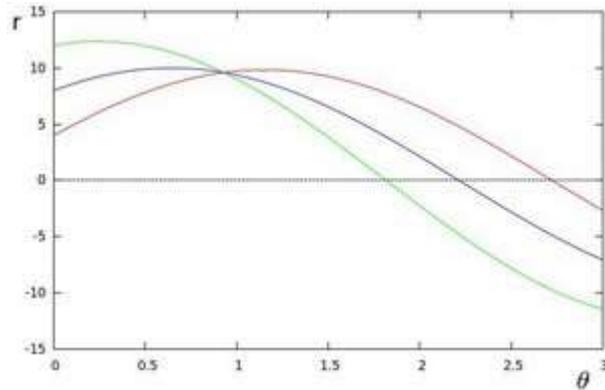


Figure 3.1.3.3 Intersection of two sinusoids on the plane  $\theta - r$

It means that in general, a line can be detected by finding the number of intersections between curves. The more curves intersecting means that the line represented by that intersection have more points. Taken as a whole, we can define a threshold of the minimum number of intersections needed to detect a line.

This is what the Hough Line Transform does. It keeps track of the intersection between curves of every point in the image. If the number of intersections is above some threshold, then it declares it as a line with the parameters  $(\theta, r_\theta)$  of the intersection point [5].

After drawing the lines based on the calculated coordinates and filling in the space between them, we get an image with the detected lane marking of traffic (figure 3.1.3.4)

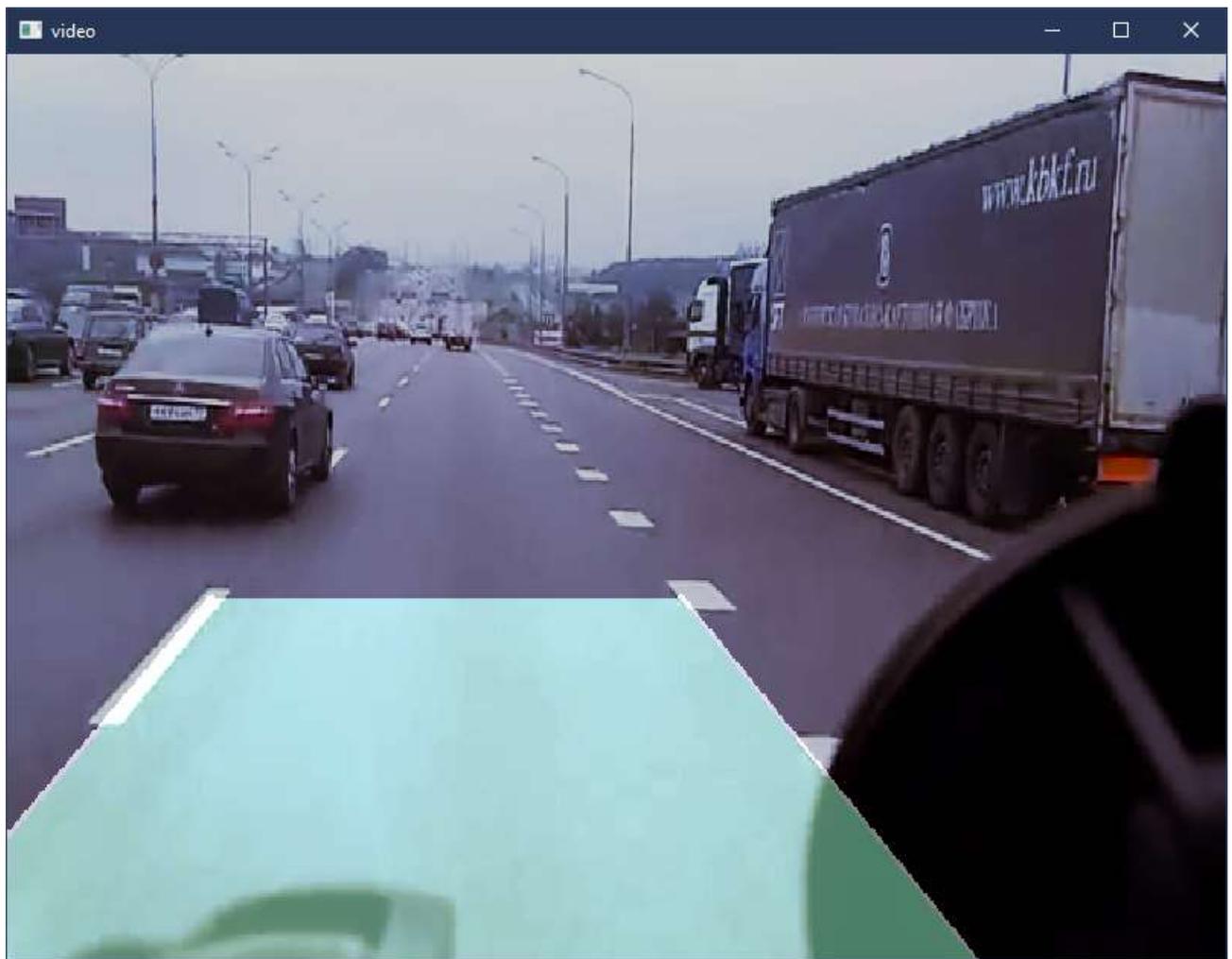


Figure 3.1.3.4 Traffic lane detector performance

## **3.2 Traffic light signal detection and recognition module**

To create and train the detector model, we used the methods of the «Dlib» library, which allow us to get a correctly working detector for the subsequent implementation of the detected objects classification functionality.

### **3.2.1 Training the detector model**

To train the model for the detector to recognize and classify objects, we need to create a training sample with images of high resolution.

On the images included in the selection, we need to mark up the detected objects. This work was carried out manually, using open access software «labelImg» (figure 3.2.1.1).

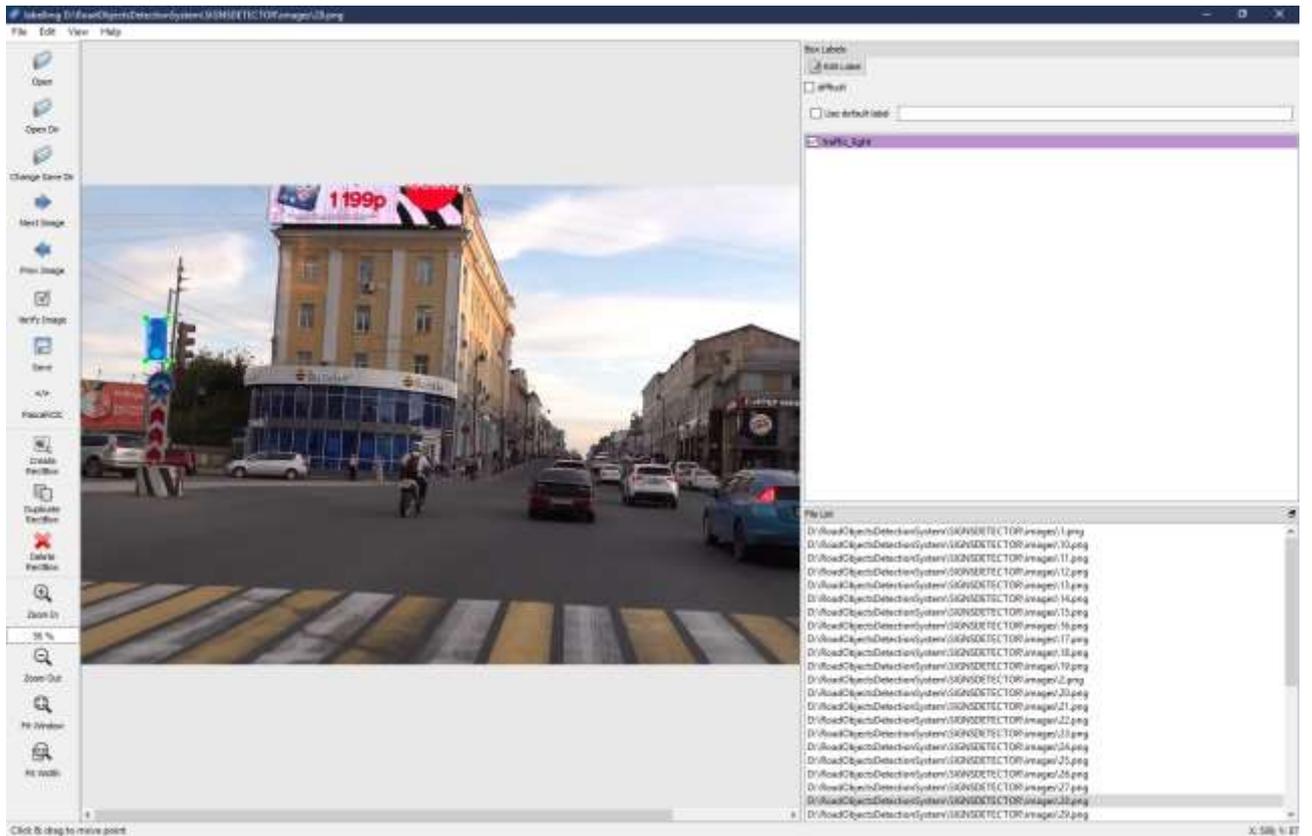


Figure 3.2.1.1 Interface of the program for forming the markup of the training sample

With this software, we can visualize images and mark the area where the detected object is located on it. When you save the markup, the image dimensions, the coordinates of the marked area, and the name of the class of the detected image that belongs to the selected area are written to the «.xml» resolution text markup file (figure 3.2.1.2).

```

<?xml version="1.0"?>
- <annotation>
  <folder>images</folder>
  <filename>1.PNG</filename>
  <path>D:\RoadObjectsDetectionSystem\SIGNSDETECTOR\images\1.PNG</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>traffic_light</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>524</xmin>
      <ymin>423</ymin>
      <xmax>551</xmax>
      <ymax>485</ymax>
    </bndbox>
  </object>
  - <object>
    <name>traffic_light</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>556</xmin>
      <ymin>426</ymin>
      <xmax>579</xmax>
      <ymax>485</ymax>
    </bndbox>
  </object>
</annotation>

```

Figure 3.2.1.2 Example of an annotation to a specific image from a training sample

During the training of the detection model, the HOG-SVM detector analyzes the marked areas in the images and writes the analyzed data to the model.

In order to train such a model, a function of the «Dlib» library called "train\_simple\_object\_detector" was used with the location of the images from the generated sample and its corresponding «.xml» files-annotations containing markup data.

### 3.2.1 Creating a classifier block

With an array of coordinates  $(x_1, y_1, x_2, y_2)$  object's areas detected by the HOG-SVM detector, adjust the image inside the area belonging to the rectangle from the coordinate array to the constant-set resolution-50px:100px using the «resize» method of the OpenCV library. Also, we need to cut off the extra edges (10px on each side), which may contain background particles with highlights (Figure 3.2.2.1).

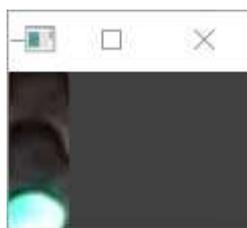


Figure 3.2.2.1 The result of the detector operation, converted to constant resolution

Using the «cvtColor» function, with the «COLOR\_BGR2HSV» argument, convert the source frame to the HSV color space.

That's array of pixel brightness values for the detected image.

To determine the signal lamp zone of the traffic light, let us calculate the sum of the pixels brightness for each zone.

The final stage of the classification is to compare the calculated values and display the corresponding information about the result on the image (figure 3.2.2.2).

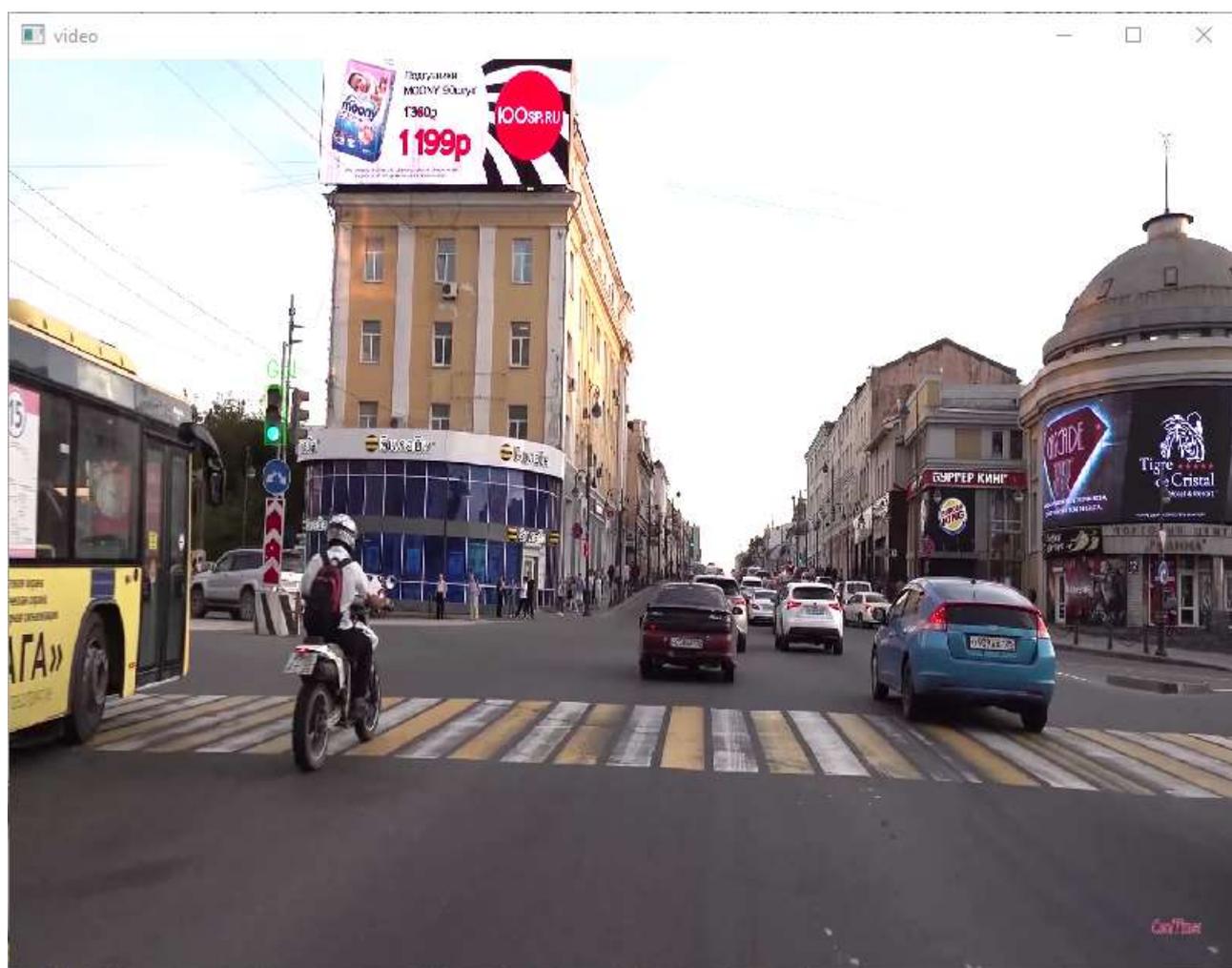


Figure 3.2.2.2 Result of the detector and classifier of traffic light signals

### **3.3 Road sign detection module**

To detect and classify the road signs, standard «OpenCV» color conversion methods were used.

#### **3.3.1 Image binarization**

To prepare the image for contour detection, we need to binarize it using a color filter for exclude a certain range of colors.

To binarize the image, the «inRange» method was used, with the initial and final values of the range of the allowed color palette.

To increase the smoothness of the contour lines, we used the «erode» function, with the number of iterations equal to two, which allows you to blur the boundaries of foreground objects. Also, to increase the borders of the foreground images, the «dilate» function was applied to the resulting image, with the number of iterations equal to two.

This approach makes it possible to increase the density and thickness of the boundaries, which significantly affects the quality of subsequent detection of contours (Figure 3.3.1.1).

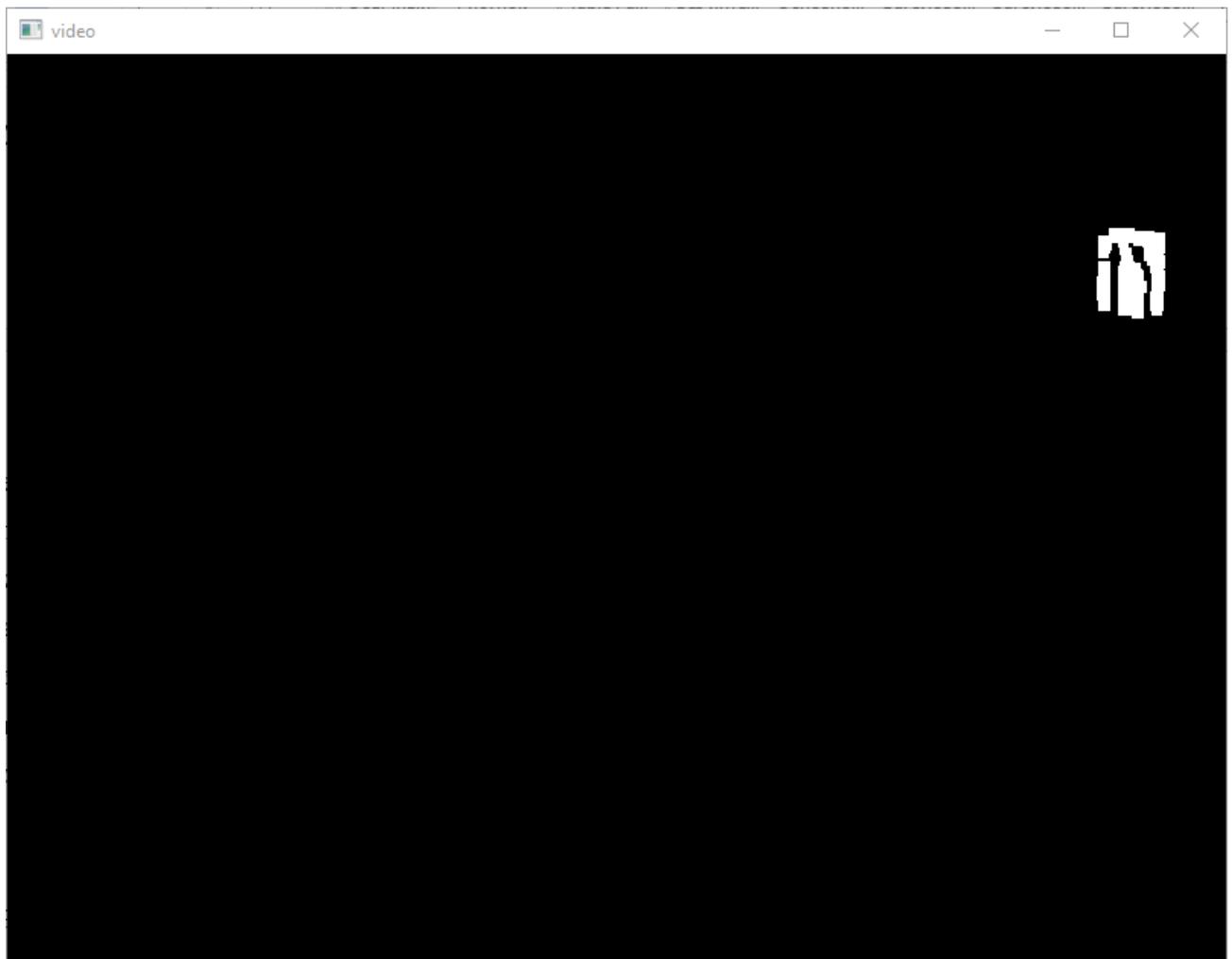


Figure 3.3.1.1 Result of image binarization

### 3.3.2 Contour detection

The next step in developing the detector is to search for contours, which uses the «findContours» function. This function returns an array of properties of found contours in the input image. From these contours, we are only interested in its coordinates, which are located in the first cell of the resulting array. We can cut off all unnecessary information.

To get an array of coordinates of the largest contour, we need to sort by the area of the image area covered by each contour. We can search for such an area using the «Contour» function. The element with the largest area, cut off by a closed contour, is located in the first element of the sorted array.

For classification, we will cut off the area formed by the obtained coordinates and reduce it to a fixed size-64px:64px (Figures 3.3.2.1a and 3.3.2.1b).

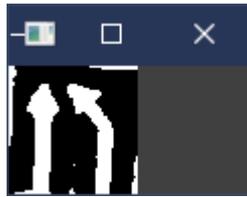


Figure 3.3.2.1a Area #1 defined by the detector

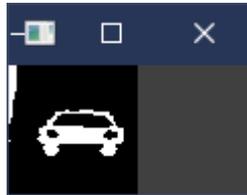


Figure 3.3.2.1b Area #2 defined by the detector

### 3.3.3 Reference classification

To classify the detected road signs, we need to develop an algorithm for pixel-by-pixel comparison of the resulting image with the template image.

In order to make a comparison, we need to convert the resulting image and the reference image to a single resolution. In this case, the most optimal resolution was chosen, which provides sufficient comparison efficiency, with an acceptable error of 64px:64px. The conversion is performed using the «resize» function (figures 3.3.3.1a and 3.3.3.1b).

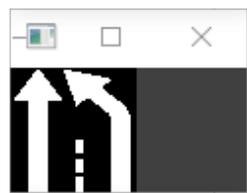


Figure 3.3.3.1a Reference image #1

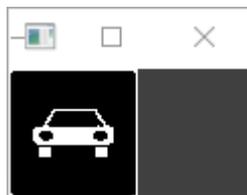


Figure 3.3.3.1b Reference image #2

The final stage of classification of the detected traffic signs is the development of a cycle for comparing each pixel of the resulting image and filling in the match counters for each key. After exiting the loop, based on the data recorded in the counters, we make a conclusion about the classification of the detected image and display information about the results of the analysis on the image in the form of a rectangular area with a caption, if the object belongs to any class (figures 3.3.3.2a and 3.3.3.2b).



Figure 3.3.3.2a Result #1 of the classifier operation

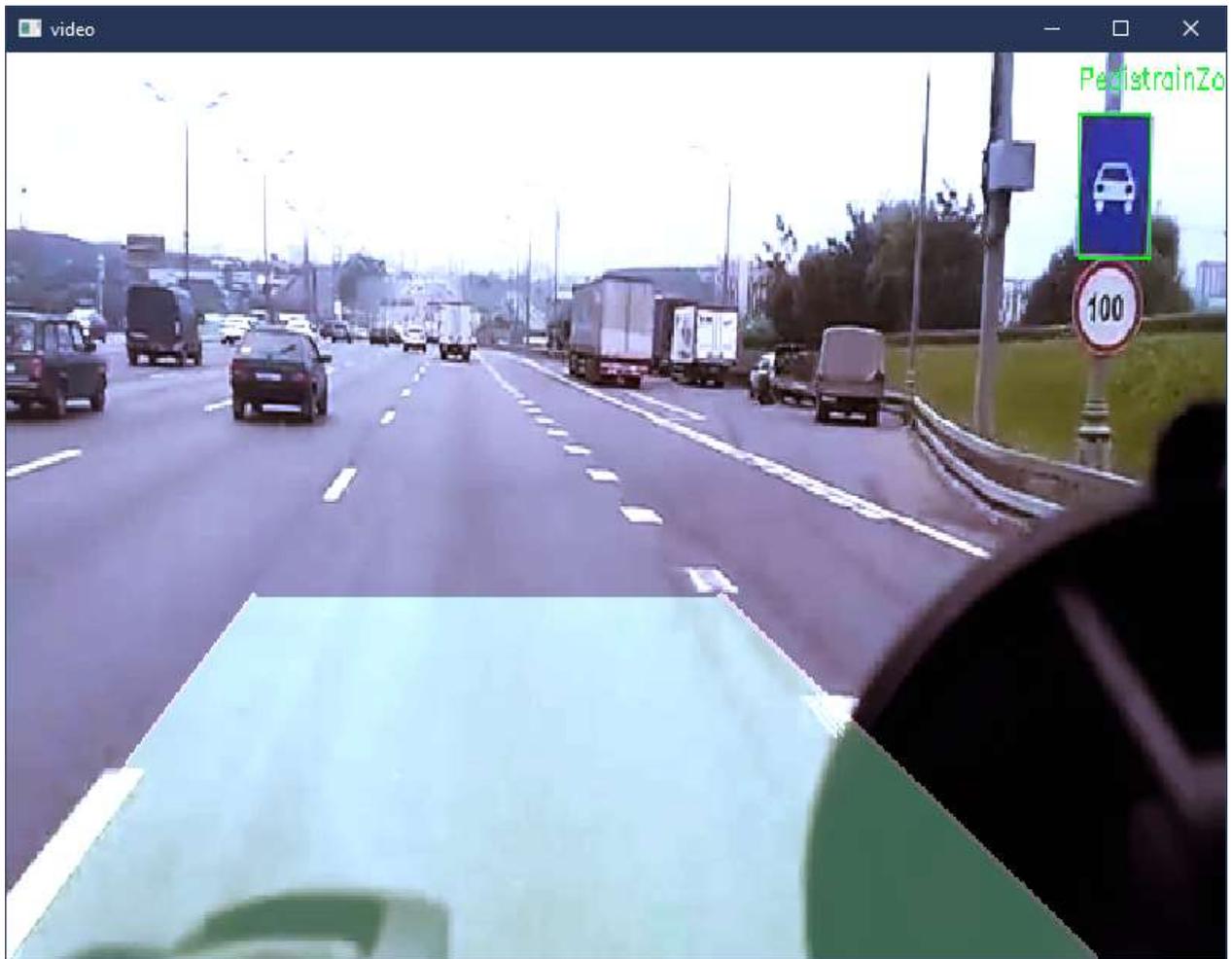


Figure 3.3.3.2b Result #2 of the classifier operation

### **3.4 Detection of road participants**

The task of detecting road participants, such as various types of transport, cyclists and motorcyclists, pedestrians, etc., is too complex to implement its functionality through simple detectors. Detecting and classifying such a large number of objects that differ in their properties and characteristics is a fairly large-scale task. As a software component for implementing the classification algorithm, it was decided to use an add – on for TensorFlow and Keras, which simplifies the implementation of object recognition algorithms-ImageAI.

#### **3.4.1 A neural network architecture**

The video processing speed was estimated using 10 different flower videos with an average duration of 20 seconds.

Table 3.4.1.1 shows the values of average precision for each class for each SNA architecture, mAP, video processing speed in frames per second (FPS), and the volume of weight coefficients in MB [6].

Table 3.4.1.1 Comparison of convolutional neural network architectures

Architecture	Video processing speed, FPS	mAP, %	Volume of coefficients, MB
YOLOv3	35	0.6757	241
RetinaNet	29	0.6809	426
YOLOv3-tiny	65	0.3529	33

As a conclusion to the tests conducted, we can say that the YOLOv3 architecture shows a less accurate result, compared to RetinaNet, but spends less time on video processing. YOLOv3-tiny is much faster and more productive than its full version, but to the increase in processing speed, it loses accuracy.

### 3.4.2 Configuring the Detector model

To create the correct processing of streaming video, you must first configure the detector specifying the following characteristics as input:

- detection speed (there is a choice between "normal", "fast", "faster" and "flash") determines the relationship between the image processing speed of the convolutional neural network and the efficiency of detection and classification;
- a set of object classes, which is a list of object types that will be detected by the convolutional neural network during image processing;
- specify an array of points as input and output data;
- specify the analyzed frame extracted from the video sequence as the input image.

#### 3.4.1 Detector in the system

To optimize performance and reduce the load on the system, it was decided to process the neural network not every frame of the video stream, but to delimit the intervals with the time of its playback.

The optimal time to start image processing by the convolutional neural network was selected – one second of the playback time. This approach to implementation allows you to get rid of unnecessary iterations of recalculating the detected images coordinates and reduce the load on the system several times.

After entering the start time interval, it is necessary to display the results of detection and classification on the images after the selected time interval.

System gets a list of coordinates of the detection area, then enters each of these areas in a rectangle of the appropriate size, and also displays information about the class that the initialized object belongs to above the found area (figure 3.4.3.1).

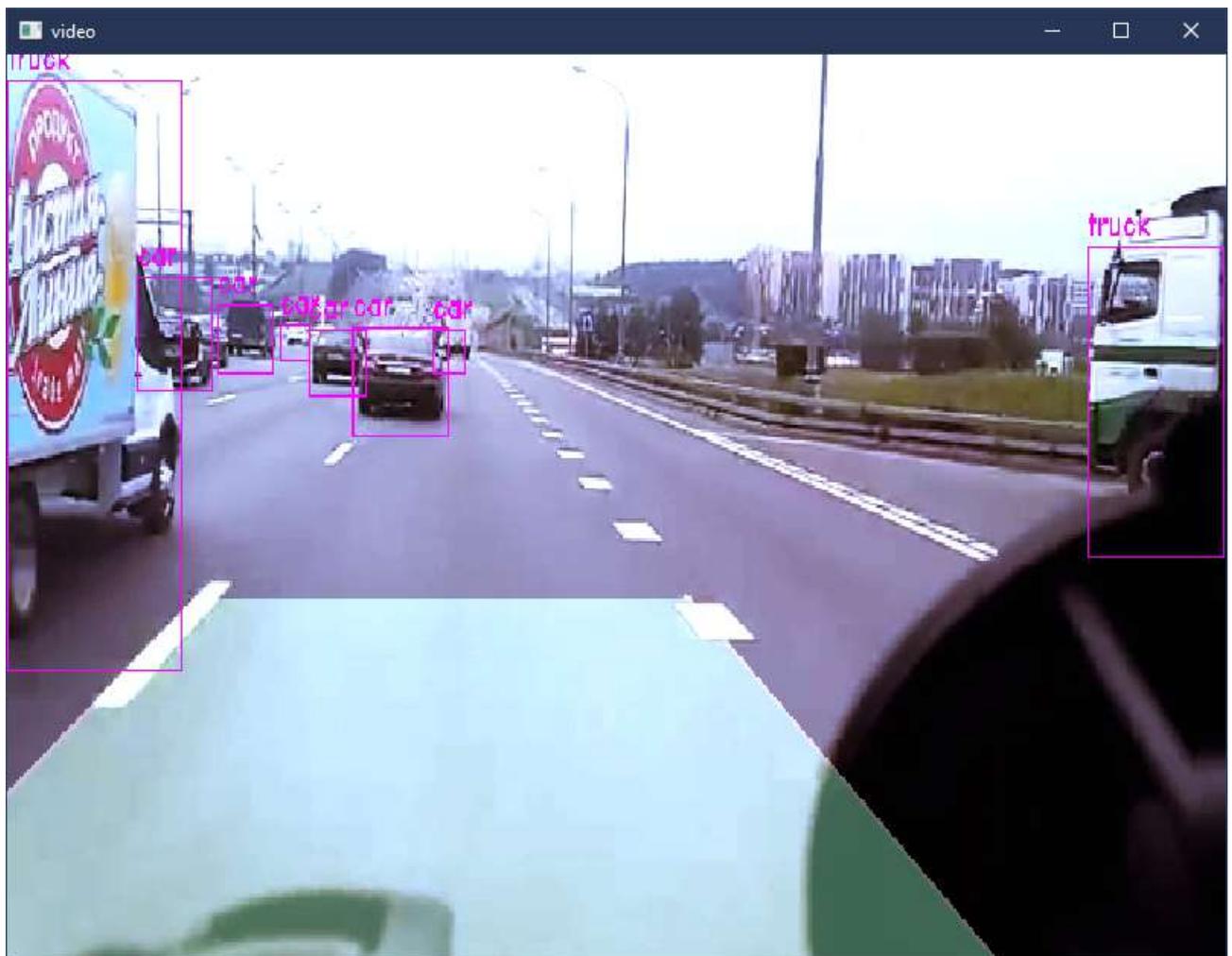


Figure 3.4.3.1 Result of the classifier of road participants

## Приложение Б– Детектор дорожной разметки

```
frame = lanes.canny(frame)
frame = lanes.region_of_interest(frame)
lines = cv.HoughLinesP(frame,2,np.pi/180, 44, np.array([()]),
minLineLength=20, maxLineGap=100)
averaged_lines = lanes.average_slope_intercept(frame,lines)
line_image = lanes.display_lines(copy_img, averaged_lines)

combo = cv.addWeighted(copy_img, 0.8, line_image, 0.5, 1)
```

```

import cv2 as cv
import numpy as np

def canny(img):
    img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
    blur = cv.GaussianBlur(img, (5, 5), 0)
    return cv.Canny(img, 50, 150)

def make_coordinates(image, line_parameters):
    slope, intercept = line_parameters
    y1 = image.shape[0]
    y2 = int(y1*(3/5))
    x1 = int((y1-intercept)/slope)
    x2 = int((y2-intercept)/slope)
    return np.array([x1, y1, x2, y2])

def average_slope_intercept(image, lines):
    left_fit = []
    right_fit = []

    while lines is not None:
        for line in lines:
            x1, y1, x2, y2 = line.reshape(4)
            parameters = np.polyfit((x1, x2), (y1, y2), 1)
            slope = parameters[0]
            intercept = parameters[1]
            if slope < 0:
                left_fit.append((slope, intercept))
            else:
                right_fit.append((slope, intercept))

        left_fit_average = np.average(left_fit, axis=0)
        left_line = make_coordinates(image, left_fit_average)
        right_fit_average = np.average(right_fit, axis=0)
        right_line = make_coordinates(image, right_fit_average)
        return np.array([left_line, right_line])

def display_lines(image, lines):
    line_image = np.zeros_like(image)
    if lines is not None:
        i = 1
        for x1, y1, x2, y2 in lines:
            if i == 1:
                cv.line(line_image, (x1, y1), (x2, y2), (255, 255, 255), 4)
                pl = [x1, y1, x2, y2]
                i+=1
            else:
                cv.line(line_image, (x1, y1), (x2, y2), (255, 255, 255), 4)
                pts = np.array([[pl[0], pl[1]], [pl[2], pl[3]], [x2, y2], [x1, y1]])
                cv.fillPoly(line_image, pts, (202, 255, 150), lineType=8, shift=0,
                dtype=np.int32,
                offset=None)
        return line_image

def region_of_interest(image):
    height = image.shape[0]
    #polygons = np.array([(230, height//3), (250, height//3), (400, 250),
    (0, 250)])
    polygons = np.array([(225, height//3), (250, height//3), (400, 250),

```

```
(0,250)])  
    mask = np.zeros_like(image)  
    cv.fillPoly(mask, np.array([polygons], dtype=np.int64), 1024)  
    masked_image = cv.bitwise_and(image, mask)  
    return masked_image
```

## Приложение Г Обучение HOG-SVM детектора

```
import cv2
import os
import dlib
import xml.etree.ElementTree as pars

dir=r"D:\RoadObjectsDetectionSystem\SIGNSDETECTOR"
images=[]
annots=[]

ImgNameList = os.listdir(dir+"\images")
print (ImgNameList)

for FileName in ImgNameList:
    image=cv2.imread(dir+"/images/"+FileName)
    image=cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    OnlyFileName = FileName.split(".")[0]
    print (OnlyFileName)
    e = pars.parse(dir+"/annots/xmls/"+OnlyFileName+".xml")
    root = e.getroot()

    #for object in root.findall("object"):
    object = root.find("object")
    object = object.find("bndbox")

    x = int(object.find("xmin").text)
    y = int(object.find("ymin").text)
    x2 = int(object.find("xmax").text)
    y2 = int(object.find("ymax").text)

    #if (x2-x)/(y2-y) < 0.7:
    images.append(image)
    annots.append([dlib.rectangle(left=x, top=y, right=x2, bottom=y2)])

options = dlib.simple_object_detector_training_options()
options.be_verbose = True

detector = dlib.train_simple_object_detector(images, annots, options)

detector.save("traffic_light.svm")
print ("Detector saved")
```

## Приложение Д Модуль детектирования и классификации сигналов

### светофора

```
boxes = model_detector(copy_imglight, 1)
# print(boxes)
for box in boxes:
    (x, y, xb, yb) = [box.left(), box.top(), box.right(),
box.bottom()]
    cut_imglight = cv.resize(clear[y:yb, x:xb], (50,100))
    cut_imglight = cut_imglight[10:90, 10:40]
    cv.imshow("lighttraf", cut_imglight)
    hsv = cv.cvtColor(cut_imglight, cv.COLOR_BGR2HSV)
    v = hsv[:, :, 2]
    cv.imshow("lighttraf", cut_imglight)
    red_sum = np.sum(v[0:25, 0:30])
    yellow_sum = np.sum(v[26:50, 0:30])
    green_sum = np.sum(v[51:80, 0:30])
    if green_sum > yellow_sum and green_sum > red_sum:
        cv.rectangle(copy_imglight, (x, y), (xb, yb), (0, 255, 0),
2)
        cv.putText(combo, 'Go!', (x, y - 10),
cv.FONT_HERSHEY_DUPLEX, 1, (0, 255, 0), 1)
        elif yellow_sum > green_sum and yellow_sum > red_sum:
            cv.rectangle(copy_imglight, (x, y), (xb, yb), (0, 255, 255),
2)
            cv.putText(combo, 'Wait!', (x, y - 10),
cv.FONT_HERSHEY_DUPLEX, 1, (0, 255, 255), 1)
            elif red_sum > green_sum and red_sum > yellow_sum:
                cv.rectangle(copy_imglight, (x, y), (xb, yb), (0, 0, 255),
2)
                cv.putText(combo, 'Stop!', (x, y - 10),
cv.FONT_HERSHEY_DUPLEX, 1, (0, 0, 255), 1)

combo = cv.addWeighted(combo, 0.8, copy_imglight, 0.5, 1)
```

## Приложение Е Модуль детектирования и классификации знаков дорожного движения

```
#copy_imgsign = cv.blur(copy_imgsign, (5,5))
mask = cv.inRange(copy_imgsign, (89,130,73), (255,255,255))
mask = cv.erode(mask, None, iterations=2)
mask = cv.dilate(mask, None, iterations=2)
#result = cv.bitwise_and(copy_imgsign, copy_imgsign, maskDi)
contours = cv.findContours(mask, cv.RETR_TREE, cv.CHAIN_APPROX_NONE)
contours = contours[0]
if contours:
    contours = sorted(contours, key=cv.contourArea, reverse=True)
    #cv.drawContours(combo, contours[0], -1, (255,0,255), 3)
    (x,y,w,h) = cv.boundingRect(contours[0])

    hol=frameCopy[y:y+h, x:x+w]
    hol=cv.resize(hol, (64,64))
    hol=cv.inRange(hol, (89,108,73), (255,255,255))
    cv.imshow("Sign",hol)

    EndOfLine_val= 0
    PedistrainZone_val = 0

    for i in range(64):
        for j in range(64):
            if hol[i][j]==EndOfLine[i][j]:
                EndOfLine_val+=1
            if hol[i][j]==PedistrainZone[i][j]:
                PedistrainZone_val+=1

    print("Pedistrain = ", PedistrainZone_val, "EndLine = ",
EndOfLine_val)

    if EndOfLine_val>3000:
        cv.rectangle(combo, (x, y), (x + w, y + h), (0, 255, 0), 1)
        cv.putText(combo, 'EndOfLine', (x, y-10),
cv.FONT_HERSHEY_SIMPLEX, 0.4, (36,255,12), 1)
    else:
        if PedistrainZone_val>3500:
            cv.rectangle(combo, (x, y), (x + w, y + h), (0, 255, 0),
1)
            cv.putText(combo, 'PedistrainZone', (x, y-10),
cv.FONT_HERSHEY_SIMPLEX, 0.4, (36,255,12), 1)
```

## Приложение Ж Модуль распознавания участников дорожного

### ДВИЖЕНИЯ

```
start_imageai = time.time()
    if start_imageai-finish_imageai>0.5:
        _, array_detect = detector.detectObjectsFromImage(
            custom_objects=custom_objects,
            input_image=combo,
            input_type="array",
            output_type="array")
        finish_imageai = time.time()

    for obj in array_detect:
        coord = obj['box_points']
        cv.rectangle(combo, (coord[0], coord[1]), (coord[2], coord[3]),
(255,0,255), 4)
        cv.putText(combo, obj['name'], (coord[0], coord[1]-5),
cv.FONT_HERSHEY_DUPLEX, 1, (255,0,255), 1)
```

## Приложение 3 Подготовка детектора для модуля распознавания участников дорожного движения

```
detector = ObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath(os.path.join(execution_path, "yolo.h5"))
detector.loadModel("fast")
custom_objects = detector.CustomObjects(car=True, motorcycle=True, truck=True,
                                         person=True, bus=True, bicycle=True,
                                         traffic_light=False, train=True,
                                         dog=True, cat=True)

finish_imageai = 0
finish_svm_hog = 0
array_detect = []
```

## Приложение И Полный код разработанной системы распознавания объектов дорожного движения

```
from cv2 import cv2 as cv
from imageai.Detection import ObjectDetection
import numpy as np
import lanes
import os
import dlib
import time

execution_path = os.getcwd()

video = cv.VideoCapture("video_full.mp4")
model_detector = dlib.simple_object_detector("traffic_light.svm")

detector = ObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath(os.path.join(execution_path, "yolo.h5"))
detector.loadModel("fast")
custom_objects = detector.CustomObjects(car=True, motorcycle=True, truck=True,
                                         person=True, bus=True, bicycle=True,
                                         traffic_light=False, train=True,
                                         dog=True, cat=True)

finish_imageai = 0
finish_svm_hog = 0
array_detect = []
# istraflight = False

Rdmrk = False
Trffclght = True
SgnDtct = True
ImgAIDtct = True

if not video.isOpened():
    print('error while opening video')

cv.waitKey(30)

EndOfLine=cv.imread("1_1.JPG")
EndOfLine_nice=cv.imread("1.JPG")
PedistrainZone=cv.imread("2.PNG")
PedistrainZone_nice=cv.imread("2.PNG")
EndOfLine=cv.resize(EndOfLine, (64,64))
PedistrainZone=cv.resize(PedistrainZone, (64,64))
EndOfLine=cv.inRange(EndOfLine, (89,108,73), (255,255,255))
PedistrainZone=cv.inRange(PedistrainZone, (89,108,73), (255,255,255))
cv.imshow("EndOfLine",EndOfLine)
cv.imshow("PedistrainZone",PedistrainZone)

while video.isOpened():
    ret, frame = video.read()

    cv.resizeWindow("video", 800, 600)
    cv.resizeWindow("lighttraf", 40, 100)
    cv.namedWindow("video", cv.WINDOW_NORMAL)

    clear = frame.copy()
    copy_img = np.copy(frame)
    #copy_img = frame.copy()
    copy_imglight = frame.copy()
    copy_imgsign = cv.cvtColor(frame.copy(), cv.COLOR_BGR2HSV)
    frameCopy = frame.copy()
```

```

try:
    combo = frame
    if Rdmrk:
        frame = lanes.canny(frame)
        frame = lanes.region_of_interest(frame)
        lines = cv.HoughLinesP(frame,2,np.pi/180, 44, np.array([()]),
minLineLength=20, maxLineGap=100)
        averaged_lines = lanes.average_slope_intercept(frame,lines)
        line_image = lanes.display_lines(copy_img, averaged_lines)

        combo = cv.addWeighted(copy_img, 0.8, line_image, 0.5, 1)

    if Trffcflght:
        boxes = model_detector(copy_imglight, 1)
        # print(boxes)
        for box in boxes:
            (x, y, xb, yb) = [box.left(), box.top(), box.right(),
box.bottom()]
            cut_imglight = cv.resize(clear[y:yb, x:xb], (50,100))
            cut_imglight = cut_imglight[10:90, 10:40]
            cv.imshow("lighttraf", cut_imglight)
            hsv = cv.cvtColor(cut_imglight, cv.COLOR_BGR2HSV)
            v = hsv[:, :, 2]
            cv.imshow("lighttraf", cut_imglight)
            red_sum = np.sum(v[0:25, 0:30])
            yellow_sum = np.sum(v[26:50, 0:30])
            green_sum = np.sum(v[51:80, 0:30])
            if green_sum > yellow_sum and green_sum > red_sum:
                cv.rectangle(copy_imglight, (x, y), (xb, yb), (0, 255, 0),
2)
                cv.putText(combo, 'Go!', (x, y - 10),
cv.FONT_HERSHEY_DUPLEX, 1, (0, 255, 0), 1)
                elif yellow_sum > green_sum and yellow_sum > red_sum:
                    cv.rectangle(copy_imglight, (x, y), (xb, yb), (0, 255, 255),
2)
                    cv.putText(combo, 'Wait!', (x, y - 10),
cv.FONT_HERSHEY_DUPLEX, 1, (0, 255, 255), 1)
                    elif red_sum > green_sum and red_sum > yellow_sum:
                        cv.rectangle(copy_imglight, (x, y), (xb, yb), (0, 0, 255),
2)
                        cv.putText(combo, 'Stop!', (x, y - 10),
cv.FONT_HERSHEY_DUPLEX, 1, (0, 0, 255), 1)

            combo = cv.addWeighted(combo, 0.8, copy_imglight, 0.5, 1)

    if SgnDtct:
        #copy_imgsign = cv.blur(copy_imgsign, (5,5))
        mask = cv.inRange(copy_imgsign, (89,130,73), (255,255,255))
        mask = cv.erode(mask, None, iterations=2)
        mask = cv.dilate(mask, None, iterations=2)
        #result = cv.bitwise_and(copy_imgsign, copy_imgsign, maskDi)
        contours = cv.findContours(mask, cv.RETR_TREE, cv.CHAIN_APPROX_NONE)
        contours = contours[0]
        if contours:
            contours = sorted(contours, key=cv.contourArea, reverse=True)
            #cv.drawContours(combo, contours[0], -1, (255,0,255), 3)
            (x,y,w,h) = cv.boundingRect(contours[0])

            hol=frameCopy[y:y+h, x:x+w]
            hol=cv.resize(hol, (64,64))
            hol=cv.inRange(hol, (89,108,73), (255,255,255))
            cv.imshow("Sign",hol)

```

```

EndOfLine_val= 0
PedistrainZone_val = 0

for i in range(64):
    for j in range(64):
        if hol[i][j]==EndOfLine[i][j]:
            EndOfLine_val+=1
        if hol[i][j]==PedistrainZone[i][j]:
            PedistrainZone_val+=1

print("Pedistrain = ", PedistrainZone_val, "EndLine = ",
EndOfLine_val)

if EndOfLine_val>3000:
    cv.rectangle(combo, (x, y), (x + w, y + h), (0, 255, 0), 1)
    cv.putText(combo, 'EndOfLine', (x, y-10),
cv.FONT_HERSHEY_SIMPLEX, 0.4, (36,255,12), 1)
else:
    if PedistrainZone_val>3500:
        cv.rectangle(combo, (x, y), (x + w, y + h), (0, 255, 0),
1)
        cv.putText(combo, 'PedistrainZone', (x, y-10),
cv.FONT_HERSHEY_SIMPLEX, 0.4, (36,255,12), 1)

if ImgAIDtct:
    start_imageai = time.time()
    if start_imageai-finish_imageai>0.5:
        _, array_detect = detector.detectObjectsFromImage(
            custom_objects=custom_objects,
            input_image=combo,
            input_type="array",
            output_type="array")
        finish_imageai = time.time()

    for obj in array_detect:
        coord = obj['box_points']
        cv.rectangle(combo, (coord[0], coord[1]), (coord[2], coord[3]),
(255,0,255), 4)
        cv.putText(combo, obj['name'], (coord[0], coord[1]-5),
cv.FONT_HERSHEY_DUPLEX, 1, (255,0,255), 1)

    cv.imshow("video", combo)
except:
    pass

if cv.waitKey(60) & 0xFF == ord('q'):
    video.release()
    cv.destroyAllWindows()

video.release()
cv.destroyAllWindows()

```