



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

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.01 Информатика и вычислительная техника
ООП/ОПОП – Data Science in Software Engineering / Анализ данных в разработке программного обеспечения
Отделение школы (НОЦ) – Отделение информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРАНТА

Тема работы
Детектирование летающих объектов с помощью свёрточных нейронных сетей класса YOLO

УДК 004.032.26:004.93`1

Обучающийся

Группа	ФИО	Подпись	Дата
8ВМ1И	Мунько Александр Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор (ОИТ, ИШИТР)	Марков Н. Г.	Д.Т.Н., профессор		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент (ОСГН, ШБИП)	Спицына Л. Ю.	к.э.н.		

По разделу «Социальная ответственность»

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

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент (ОИТ, ИШИТР)	Зарубин А. Г.	к.х.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП
по направлению 09.04.01 Информатика и вычислительная техника

Код компетенции	Наименование компетенции
Универсальные компетенции	
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
Общепрофессиональные компетенции	
ОПК(У)-1	Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, 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
Профессиональные компетенции	

Код компетенции	Наименование компетенции
ПК(У)-1	Способен разрабатывать и администрировать системы управления базами данных
ПК(У)-2	Способен к созданию программного обеспечения для анализа, распознавания и обработки информации, систем цифровой обработки сигналов
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



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

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.01 Информатика и вычислительная техника
ООП/ОПОП – Data Science in Software Engineering / Анализ данных в разработке программного обеспечения
Отделение школы (НОЦ) – Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

_____ А.Г. Зарубин
(Подпись) _____ (Дата) _____ (Ф.И.О.)

**ЗАДАНИЕ
на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации

Студенту:

Группа	ФИО
8ВМ1И	Мунько Александру Сергеевичу

Тема работы:

Детектирование летающих объектов с помощью свёрточных нейронных сетей класса YOLO	
Утверждена приказом директора (дата, номер)	№ 138-44/с от 18.05.2023

Срок сдачи студентом выполненной работы:

	29.05.2023
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<ol style="list-style-type: none"> 1. Набор RGB-изображений летающих объектов четырёх классов («Птица», «БПЛА вертолётного типа», «БПЛА самолётного типа», «Неизвестный объект») для формирования датасетов. 2. Использование свёрточных нейронных сетей (СНС) класса YOLO. 3. Точность детектирования по метрике $AP_{0,5}$ для отдельных классов объектов и усреднённой метрики $mAP_{0,5}$ больше или равна 0,9 для
---	---

	объектов разных масштабов на изображениях; больше или равна 0,7 для объектов малых размеров на изображениях.
<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</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"> 1. Анализ моделей СНС, с целью создания мобильных систем компьютерного зрения. 2. Формирование четырёх датасетов. 3. Разработка, обучение, валидация и исследование четырёх компактных моделей СНС. 4. Проведение комплексных исследований четырёх моделей СНС, анализ полученных результатов. 5. Финансовый менеджмент. 6. Социальная ответственность. 7. Раздел на английском языке.
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ol style="list-style-type: none"> 1. Схемы моделей СНС. 2. Таблицы результатов обучения, валидации и исследования моделей СНС. 3. Диаграмма Ганта.
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Спицына Л. Ю., Доцент (ОСГН, ШБИП), к.э.н.
Социальная ответственность	Антоневич О. А., Доцент (ООД, ШБИП), к.б.н.
Английский язык	Ануфриева Т. Н., Старший преподаватель (ОИЯ, ШБИП)
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
Chapter 1. Convolutional neural network models for mobile computer vision systems	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	30.01.2023
--	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор (ОИТ, ИШИТР)	Марков Н. Г.	д.т.н, профессор		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Мунько Александр Сергеевич		



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

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.01 Информатика и вычислительная техника
ООП/ОПОП – Data Science in Software Engineering / Анализ данных в разработке программного обеспечения
Отделение школы (НОЦ) – Отделение информационных технологий
Период выполнения – Весенний семестр 2022 / 2023 учебного года

Форма представления работы:

магистерская диссертация

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

Срок сдачи студентом выполненной работы:

29.05.2023

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
25.05.2023	Основная часть ВКР, включая раздел на английском языке	70
16.05.2023	Раздел «Социальная ответственность»	15
22.05.2023	Раздел «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор (ОИТ, ИШИТР)	Марков Н. Г.	д.т.н, профессор		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент (ОИТ, ИШИТР)	Зарубин А. Г.	к.х.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ВМ1И	Мунько Александру Сергеевичу

Школа	Инженерная школа информационных технологий и робототехники	Отделение школы (НОЦ)	ОИТ
Уровень образования	магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Бюджет проекта – не более 979 812,47 руб., в т.ч. затраты по оплате труда – не более 605 750,06 руб.
2. Нормы и нормативы расходования ресурсов	Значение показателя интегральной ресурсоэффективности – не менее 4,75 баллов из 5
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды 30%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	Провести предпроектный анализ
2. Разработка устава научно-технического проекта	Представить Устав научного проекта магистерской работы
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Разработать план управления НТИ
4. Определение ресурсной, финансовой, экономической эффективности	Рассчитать сравнительную эффективность исследования

Перечень графического материала (с точным указанием обязательных чертежей):

<ol style="list-style-type: none"> 1. Сегментирование рынка 2. Оценка конкурентоспособности технических решений 3. Матрица SWOT 4. Иерархическая структура работ 5. График проведения и бюджет НТИ 6. Оценка сравнительной эффективности НТИ 	
--	--

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент (ОСГН, ШБИП)	Спицына Любовь Юрьевна	К.Э.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Мунько Александр Сергеевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8ВМ1И		Мунько Александру Сергеевичу	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	ОИТ
Уровень образования	магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

Тема ВКР:

Детектирование летающих объектов с помощью сверточных нейронных сетей класса YOLO	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение</p> <ul style="list-style-type: none"> – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации 	<p><i>Объект исследования:</i> сверточные нейронные сети семейства YOLO, которые используются для детектирования (классификации) объектов на изображениях;</p> <p><i>Область применения:</i> системы компьютерного зрения, машинное обучение;</p> <p><i>Рабочая зона:</i> офис;</p> <p><i>Размеры помещения:</i> 2,4×4 м.;</p> <p><i>Количество и наименование оборудования рабочей зоны:</i> персональный компьютер, ЖК монитор, офисное кресло, центральное отопление; естественная вентиляция; искусственное и естественное освещение;</p> <p><i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> сбор и обработка данных, разработка архитектур сверточных нейронных сетей, обучение и исследование сверточных нейронных сетей;</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<p>ГОСТ 12.2.032-78 Рабочее место при выполнении работ сидя;</p> <p>ГОСТ 21889-76 Система "Человек-машина". Кресло человека-оператора;</p> <p>ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места;</p> <p>ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения;</p> <p>СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.</p>
<p>2. Производственная безопасность при разработке проектного решения:</p> <ul style="list-style-type: none"> – Анализ выявленных вредных и опасных производственных факторов – Расчет уровня опасного или вредного производственного фактора 	<p>Опасные факторы:</p> <p>1. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий.</p> <p>Вредные факторы:</p> <p>1. Отсутствие или недостатки необходимого искусственного освещения.</p> <p>2. Рабочая поза;</p> <p>3. Умственное перенапряжение, в том числе вызванное информационной нагрузкой.</p> <p>4. Перенапряжение анализаторов, в том числе вызванное информационной нагрузкой.</p>

	<p>5. Монотонность труда, вызывающая монотонию;</p> <p>6. Отклонение показателей микроклимата;</p> <p>7. Повышенный уровень шума.</p> <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: использование устройств защитного заземления, увеличение источников искусственного освещения, использование очков при работе с ЭВМ.</p> <p>Расчет: расчет системы искусственного освещения.</p>
3. Экологическая безопасность при разработке проектного решения	<p>Воздействие на селитебную зону: отсутствует.</p> <p>Воздействие на литосферу: неправильная утилизации комплектующих ЭВМ; неправильная утилизации люминесцентных ламп.</p> <p>Воздействие на гидросферу: отсутствует.</p> <p>Воздействие на атмосферу: выброс вредных веществ в результате выработки электроэнергии, необходимой для работы ЭВМ; неправильная утилизации комплектующих ЭВМ.</p>
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	<p>Возможные ЧС: отказ систем безопасности, пожар, выход из строя электрической системы, землетрясение, пожар.</p> <p>Наиболее типичная ЧС: пожар.</p>

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент (ООД, ШБИП)	Антоневич Ольга Алексеевна	к.б.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Мунько Александр Сергеевич		

РЕФЕРАТ

Выпускная квалификационная работа 177 с., 40 рис., 42 табл., 77 источников, 14 прил.

Ключевые слова: детектирование летающих объектов на изображениях, свёрточная нейронная сеть класса YOLO, датасет, мобильная система компьютерного зрения, машинное обучение.

Объектом исследования являются изображения летающих объектов четырёх классов: «Птица», «БПЛА самолётного типа», «БПЛА вертолётного типа» и «Неизвестный объект».

Цель работы — разработка и исследование моделей свёрточных нейронных сетей (СНС) класса YOLO для детектирования (обнаружения, локализации и классификации) летающих объектов различных размеров на изображениях и выбор в результате исследований, наиболее эффективных моделей СНС для создания мобильных систем компьютерного зрения (СКЗ) реального времени.

В результате исследования разработаны алгоритмы, в том числе оригинальный алгоритм определения на изображении ограничивающих объект прямоугольников, и программное обеспечение для создания датасетов. С их помощью сформированы четыре датасета на основе изображений летающих объектов четырёх классов. Разработаны, программно-реализованы и обучены три компактные модели СНС и предложена новая модель СНС на основе модели YOLOv5 Small, входящие в класс моделей YOLO. Проведены комплексные исследования этих четырёх моделей СНС, получены новые результаты, позволившие оценить эффективность таких моделей СНС. Разработаны рекомендации по использованию наиболее эффективных моделей СНС при создании мобильных СКЗ реального времени.

Основные технико-эксплуатационные характеристики: наиболее эффективные модели СНС удовлетворяют требованиям со стороны проектируемых мобильных СКЗ реального времени.

Степень внедрения: основные полученные результаты внедрены в ООО «ЛЭМЗ–Т».

ОГЛАВЛЕНИЕ

ОСНОВНЫЕ СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ	15
ВВЕДЕНИЕ.....	16
ГЛАВА 1. МОДЕЛИ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ СОЗДАНИЯ МОБИЛЬНЫХ СИСТЕМ КОМПЬЮТЕРНОГО ЗРЕНИЯ	19
1.1 Основные классы задач компьютерного зрения.....	19
1.2 Мобильные системы компьютерного зрения.....	20
1.3 Требования к мобильной СКЗ для детектирования летающих объектов на изображениях.....	22
1.4 Анализ моделей СНС для детектирования объектов	26
1.4.1 Первые модели СНС для детектирования объектов	26
1.4.2 Модели СНС класса YOLO.....	27
1.4.3 Компактные модели СНС из класса YOLO.....	33
1.5 Цель и задачи исследований	34
ГЛАВА 2. СОЗДАНИЕ ДАТАСЕТОВ ДЛЯ ОБУЧЕНИЯ И ИССЛЕДОВАНИЯ МОДЕЛЕЙ СНС	37
2.1 Описание используемых форматов датасетов	37
2.2 Преобразование аннотаций из формата TensorFlow Object Detection CSV в формат YOLO PyTorch TXT	39
2.3 Преобразование аннотаций из формата YOLO Keras TXT в формат YOLO PyTorch TXT	40
2.4 Определение ограничивающих прямоугольников	41
2.5 Формирование основного датасета	46
2.6 Формирование трёх дополнительных датасетов	49
ГЛАВА 3. РАЗРАБОТКА, ОБУЧЕНИЕ И ВАЛИДАЦИЯ КОМПАКТНЫХ МОДЕЛЕЙ СНС	54
3.1 Общие сведения.....	54
3.2 Разработка и обучение модели СНС YOLOv4 Tiny	54

3.3	Разработка и обучение модели СНС YOLOv5 Small	56
3.4	Разработка и обучение модели СНС YOLOv7 Tiny	58
3.5	Обучение моделей СНС YOLOv5 Small на изображениях с объектами трёх категорий размеров.....	60
ГЛАВА 4. КОМПЛЕКСНЫЕ ИССЛЕДОВАНИЯ ЭФФЕКТИВНОСТИ КОМПАКТНЫХ МОДЕЛЕЙ СНС.....		63
4.1	Результаты исследования компактных моделей СНС с использованием основного датасета.....	63
4.2	Исследование моделей YOLOv5 Small с использованием дополнительных датасетов	67
4.3	Новая модель СНС и её обучение	69
4.4	Результаты исследования новой модели СНС	71
4.5	Анализ результатов исследований и рекомендации по использованию моделей СНС	73
ГЛАВА 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....		77
5.1	Предпроектный анализ	77
5.1.1	Потенциальные потребители результатов исследования	77
5.1.2	Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения.....	78
5.1.3	SWOT-анализ.....	80
5.1.4	Оценка готовности проекта к коммерциализации.....	83
5.1.5	Методы коммерциализации результатов научно-технического исследования.....	84
5.2	Инициация проекта	86
5.2.1	Цели и результат проекта.....	86
5.2.2	Организационная структура проекта	88
5.2.3	Ограничения и допущения проекта	89

5.3	Планирование управления научно-техническим проектом.....	90
5.3.1	Иерархическая структура работ проекта	90
5.3.2	План проекта.....	90
5.4	Бюджет научного исследования	94
5.4.1	Сырье, материалы, покупные изделия и полуфабрикаты.....	94
5.4.2	Затраты на электроэнергию	94
5.4.3	Специальное оборудование для научных (экспериментальных) работ	95
5.4.4	Основная заработная плата	95
5.4.5	Дополнительная заработная плата	98
5.4.6	Отчисления на социальные нужды	99
5.4.7	Накладные расходы.....	100
5.4.8	Формирование бюджетных расходов	101
5.5	Оценка сравнительной эффективности исследования	101
5.6	Выводы по разделу.....	104
ГЛАВА 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ		106
6.1	Правовые и организационные вопросы обеспечения безопасности	106
6.1.1	Специальные правовые нормы трудового законодательства.....	106
6.1.2	Организационные мероприятия при компоновке рабочей зоны	107
6.2	Производственная безопасность	109
6.2.1	Опасные производственные факторы	111
6.2.2	Вредные производственные факторы	112
6.3	Экологическая безопасность.....	118
6.4	Безопасность в чрезвычайных ситуациях.....	120
6.5	Выводы по разделу.....	121
ЗАКЛЮЧЕНИЕ		123
СПИСОК ПУБЛИКАЦИЙ		125
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ		126

ПРИЛОЖЕНИЕ А Chapter 1. Convolutional neural network models for mobile computer vision systems	133
ПРИЛОЖЕНИЕ Б Схема алгоритма нахождения ограничивающих прямоугольников.....	150
ПРИЛОЖЕНИЕ В Программа для нахождения ограничивающих прямоугольников на основе библиотеки OpenCV	153
ПРИЛОЖЕНИЕ Г Программа обучения логистической регрессии для формирования размера структурирующего элемента морфологических преобразований.....	157
ПРИЛОЖЕНИЕ Д Программа для формирования основного датасета.....	158
ПРИЛОЖЕНИЕ Е Программа для формирования датасетов по размерам объектов.....	160
ПРИЛОЖЕНИЕ Ж Архитектура модели СНС YOLOv4 Tiny.....	164
ПРИЛОЖЕНИЕ И Параметры модели СНС YOLOv4 Tiny	165
ПРИЛОЖЕНИЕ К Архитектура модели СНС YOLOv5 Small.....	170
ПРИЛОЖЕНИЕ Л Параметры модели СНС YOLOv5 Small.....	171
ПРИЛОЖЕНИЕ М Архитектура модели СНС YOLOv7 Tiny.....	172
ПРИЛОЖЕНИЕ Н Параметры модели СНС YOLOv7 Tiny	175
ПРИЛОЖЕНИЕ П Акт о внедрении	176
ПРИЛОЖЕНИЕ Р Диплом III степени за доклад	177

ОСНОВНЫЕ СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

БПЛА — беспилотный летательный аппарат;

ИСР — иерархическая структура работ

ПЛИС — программируемая логическая интегральная схема.

ПЭВМ — персональная электронно-вычислительная машина.

СКЗ — система компьютерного зрения;

СНС — свёрточная нейронная сеть.

ССБТ — система стандартов безопасности труда.

AP — average precision; средневзвешенное значение точности детектирования объектов на изображении.

CNN — convolutional neural network; свёрточная нейронная сеть.

mAP — mean average precision; среднее значение метрики AP, рассчитанное для всех классов объектов.

NMS — non-maximum suppression; это метод, используемый при обнаружении объектов, который направлен на выбор наилучшей ограничивающего прямоугольника из набора перекрывающихся прямоугольников.

SGD — stochastic gradient descent; стохастический градиентный спуск.

SWOT — метод стратегического планирования, заключающийся в выявлении факторов внутренней и внешней среды организации и разделении их на четыре категории: Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы).

ВВЕДЕНИЕ

В настоящее время во всем мире довольно интенсивно решается актуальная проблема создания современных систем компьютерного зрения (СКЗ) различного назначения. Такие системы сегодня востребованы для решения многих прикладных задач компьютерного зрения, в том числе в различных отраслях промышленности. Особое практическое значение имеют мобильные (возимые или носимые, включая встраиваемые) СКЗ на основе современных моделей свёрточных нейронных сетей (СНС). Вычислительные возможности мобильных СКЗ обычно весьма ограничены, что накладывает жёсткие требования на скорость вычисления моделей СНС. Это указывает на актуальность разработки и исследования новых моделей СНС и выбора из них тех, что отвечают требованиям по скорости вычислений и точности распознавания объектов на изображениях, выдвигаемым при проектировании конкретных мобильных СКЗ. Этим взаимоисключающим требованиям особенно непросто удовлетворить в случаях создания мобильных СКЗ, функционирующих в режиме реального времени. Именно таким требованиям должна удовлетворять мобильная СКЗ, предназначенная для детектирования (обнаружения, локализации и классификации) летающих объектов в воздушном пространстве. Более того, в практически важных случаях такая СКЗ должна одновременно распознавать объекты разных размеров (масштабов) на изображении. Особую сложность представляют летающие объекты малых размеров. Например, птицы и беспилотные летательные аппараты (БПЛА) имеют малые размеры и часто их контуры весьма близки, поэтому их трудно отличить. Все это ещё раз подчёркивает актуальность создания новых высокоточных моделей СНС как основы мобильных СКЗ для детектирования летающих объектов в воздушном пространстве в реальном времени.

Целью данной ВКР является разработка и исследование моделей СНС класса (часто говорят, семейства) YOLO для детектирования летающих объектов различных размеров (масштабов) на изображениях и выбор в результате исследований, в том числе с учётом категории объектов по размерам, наиболее

эффективных моделей СНС из этого класса для создания мобильных СКЗ реального времени.

В первой главе формулируются основные задачи компьютерного зрения и вводится понятие мобильных СКЗ. По ряду критериев анализируются различные современные модели СНС, в том числе из класса YOLO, которые, в принципе, можно использовать для создания мобильных СКЗ. В итоге выбираются три компактные модели СНС из этого класса. Формулируются цель работы и решаемые для её реализации задачи.

Во второй главе описано решение задачи создания четырёх датасетов (основного и трёх дополнительных) для обучения, валидации и исследования моделей СНС. В качестве исходных изображений при формировании каждого датасета используются изображения летающих объектов четырёх классов.

Третья глава посвящена особенностям разработки, обучения и валидации трёх компактных моделей СНС, выбранных в результате анализа моделей СНС класса YOLO в первой главе. С учётом результатов валидации выявляется наиболее эффективная из этих моделей. Затем она обучается на изображениях с объектами трёх категорий размеров, входящих в обучающие выборки дополнительных датасетов.

В четвертой главе приводятся и анализируются результаты исследований эффективности трёх обученных компактных моделей и новой модели СНС, созданной на основе наиболее эффективной из них. Приводятся результаты исследований моделей при детектировании объектов различных размеров на одном изображении и объектов с размерами одной из трёх категорий на одном изображении. Выбираются наиболее эффективные модели СНС в качестве основы для создания мобильной СКЗ реального времени для детектирования летающих объектов.

В пятой главе изложены вопросы, связанные с финансовым менеджментом, ресурсоэффективностью и ресурсосбережением разработки.

В шестой главе рассмотрены вопросы социальной ответственности: правовые и организационные вопросы, техника безопасности работников на рабо-

чем месте, экологическая безопасность и безопасность в чрезвычайных ситуациях.

Основные результаты работы внедрены в ООО «ЛЭМЗ–Т», являющемся Центром исследований и разработок ПАО «НПО «Алмаз», при выполнении НИР по созданию мобильных СКЗ различного назначения. Получен соответствующий акт.

Результаты работы докладывались на XX Международной научно–практической конференции студентов и молодых учёных «Молодёжь и современные информационные технологии», проходившей с 20 по 22 марта 2023г. в г. Томске. Получен диплом III степени за лучший доклад. Статья опубликована в сборнике трудов конференции.

Акселерация работы проводилась в рамках технологической практики в период с 30.01.2023г. по 20.04.2023г. в ООО «ЛЭМЗ–Т» под руководством наставника, начальника сектора 147 Григорьева Е. В. Автор выражает ему благодарность за ценные советы и помощь при прохождении технологической практики.

Автор выражает благодарность руководителю ВКР профессору Маркову Н. Г. за помощь и советы при выполнении работы.

ГЛАВА 1. МОДЕЛИ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ СОЗДАНИЯ МОБИЛЬНЫХ СИСТЕМ КОМПЬЮТЕРНОГО ЗРЕНИЯ

1.1 Основные классы задач компьютерного зрения

Развитие теории компьютерного зрения уже сегодня позволило решить с разной степенью точности большое число прикладных задач в робототехнике, создании беспилотного транспорта (автомобили, поезда, воздушные и подводные суда), при анализе аэрокосмических снимков и т.д. Современные достижения в глубоком обучении [1, 2, 3] позволили достичь значительных успехов в решении многих задач компьютерного зрения. В частности, применение моделей свёрточных нейронных сетей (СНС) привело к прорывным результатам в решении ряда задач детектирования объектов на изображениях [4].

Современные СКЗ на основе моделей СНС позволяют решать четыре класса задач распознавания объектов на изображениях. К ним относятся задачи следующих классов:

1. Семантической сегментации (англ. Semantic Segmentation) изображений: модель СНС на выходе для каждого пикселя изображения выдаёт вероятность принадлежности его к каждому классу; часто такие задачи называют задачами попиксельной классификации.
2. Распознавание объекта (англ. Object Recognition): модель СНС получает на вход изображение с ровно одним объектом, а на её выходе появляются координаты содержащего объект прямоугольника и его класс; таким образом, решается задача локализации и классификации одного объекта на изображении.
3. Задача объектного детектирования (англ. Object Detection), часто говорят детектирования объектов на изображении: в случае нескольких объектов на изображении модель СНС должна обвести прямоугольником и предсказать класс каждого из объектов; эта задача является одной из фундаментальных задач компьютерного зрения и заключается в одновременном обнаружении, локализации (определении местоположения) и классификации всех объектов на изображении. Ме-

стоположение объекта определяется координатами ограничивающего его прямоугольника (англ. bounding box).

4. Распознавание экземпляров объекта (англ. Instance Segmentation). Это самые сложные задачи для моделей СНС, когда на изображении несколько объектов разных классов и надо не только обнаружить и классифицировать объекты, но и отличить друг от друга разные объекты (экземпляры объекта) с совпадающим классом.

В последние годы значительное внимание уделяется решению задач второго и третьего классов. При этом исследовательский интерес большинства авторов ранних работ [5, 6], а также авторов многих современных работ [7, 8] в области распознавания / детектирования объектов на изображениях сосредоточен на достижении высокой точности распознавания / детектирования этих объектов с помощью моделей СНС, при этом эффективности алгоритмов вычисления СНС не уделяется должного внимания. Погоня за точностью распознавания / детектирования объектов приводит к разработке моделей СНС с высокой вычислительной сложностью, ведущей, в итоге, к низкой алгоритмической эффективности этих моделей. Здесь и далее под алгоритмической эффективностью будем понимать количество ресурсов вычислительного устройства, необходимое для достижения определённых показателей в решении поставленной задачи. Оценка такой эффективности — основной путь измерения алгоритмического прогресса в классических задачах информатики. Это означает, что при создании СКЗ на основе моделей СНС, позволяющих решать задачи компьютерного зрения второго и третьего классов, необходимы комплексные исследования эффективности предлагаемых моделей СНС.

1.2 Мобильные системы компьютерного зрения

Современным направлением в развитии СКЗ является создание мобильных (возимых и носимых, включая встраиваемые) вариантов СКЗ. Мобильные СКЗ востребованы также при решении задач обнаружения и классификации летающих объектов в воздушном пространстве. На изображениях и видео, получаемых при мониторинге воздушного пространства с помощью, соответствующей

щей фото или видеоаппаратуры в составе таких СКЗ, могут появляться один и более летающих объектов, причём в случае нескольких объектов они могут принадлежать к разным классам. Мобильные СКЗ на основе моделей СНС должны обнаруживать, локализовать, сопровождать (осуществлять трекинг при перемещении объектов) и классифицировать летающие объекты. По сути, каждая такая СКЗ с помощью той или иной модели СНС должна решать задачу объектного детектирования. При создании мобильных СКЗ, в принципе, можно использовать модели СНС, позволяющие распознавать один летающий объект на изображении, то есть решать задачи по распознаванию объекта. Однако СКЗ с такой моделью СНС может показать высокую точность распознавания только в случае одиночного объекта на анализируемом изображении. Поскольку модели СНС, решающие задачи детектирования нескольких летающих объектов на изображении, могут использоваться и в случае одиночного летающего объекта, тогда имеет смысл применять только их при создании мобильных СКЗ. Далее рассматриваются только модели СНС, позволяющие решать задачи из третьего класса: детектировать объекты на изображениях.

Применение упомянутых выше моделей СНС с высокой вычислительной сложностью в мобильных СКЗ при решении многих реальных задач компьютерного зрения практически невозможно. Дело в том, что во многих подвижных робототехнических комплексах, на беспилотных автомобилях и на БПЛА, и тому подобных системах задачи распознавания / детектирования объектов на изображениях с помощью входящих в их состав мобильных СКЗ необходимо решать в режиме реального времени. В случае летающих объектов при мониторинге воздушного пространства масштаб реального времени определяется скоростью перемещения таких объектов. При этом задачи распознавания / детектирования объектов на изображениях обычно приходится решать в условиях ограниченных ресурсов памяти и производительности вычислительных устройств мобильных СКЗ. Особенно эти ограничения приходится учитывать при разработке СКЗ для БПЛА и в случае носимых СКЗ, где существуют ещё и дополнительные жёсткие ограничения по энергопотреблению. Все это

позволяет считать, что применение сложных моделей СНС во многих реальных задачах создания мобильных СКЗ весьма затруднительно. В этой связи при разработке мобильных СКЗ реального времени растёт интерес к эффективным и компактным моделям СНС. Примерами работ, нацеленных на создание более эффективных моделей СНС для детектирования объектов на изображениях, могут служить исследования одноступенчатых моделей СНС [9, 10, 11, 12, 13, 14], безъякорных архитектур [15, 16, 17] и сжатия существующих моделей СНС [18, 19, 20]. По сути, для решения реальных задач компьютерного зрения с помощью мобильных СКЗ необходимы модели СНС, позволяющие соблюдать баланс между точностью детектирования и их алгоритмической эффективностью, и в итоге позволяющие мобильным СКЗ функционировать в масштабе реального времени и детектировать объекты на изображениях с приемлемой точностью.

Другой важной проблемой при создании многих СКЗ, включая мобильные системы, является необходимость детектирования с помощью таких СКЗ объектов разных масштабов (линейных размеров, площади). В первую очередь, детектирование объектов малого масштаба представляет особую сложность для современных моделей СНС. Сегодня, не смотря на общие успехи в достижении высокой точности детектирования с помощью таких моделей, все ещё сохраняется значительный разрыв между точностью детектирования объектов малого и большого масштабов. Все это указывает на актуальность создания высокоточных моделей СНС для решения задач детектирования объектов разных масштабов с помощью мобильных СКЗ.

1.3 Требования к мобильной СКЗ для детектирования летающих объектов на изображениях

Как показано выше, сегодня весьма востребованы для решения многих прикладных задач компьютерного зрения мобильные (возимые или носимые) СКЗ на основе современных моделей СНС. Вычислительные возможности мобильных СКЗ обычно весьма ограничены, что накладывает жёсткие требования на скорость вычисления моделей СНС. Это указывает на актуальность разра-

ботки и исследования новых моделей СНС и выбора из них тех, что отвечают требованиям по скорости вычислений и точности распознавания объектов на изображениях, выдвигаемым при проектировании конкретных мобильных СКЗ. Этим взаимоисключающим требованиям особенно непросто удовлетворить в случаях создания мобильных СКЗ, функционирующих в режиме реального времени.

Таким взаимоисключающим требованиям должна также удовлетворять мобильная СКЗ, предназначенная для детектирования (обнаружения, локализации и классификации) летающих объектов в воздушном пространстве. На изображениях и видео, получаемых при мониторинге воздушного пространства с помощью, соответствующей фото или видеоаппаратуры в составе такой СКЗ, могут появляться один и более летающих объектов, причём в случае нескольких объектов они могут принадлежать к разным классам. СКЗ должна не только обнаруживать и классифицировать их, но и сопровождать перемещающиеся в пространстве объекты (трекинг объектов). При этом масштаб реального времени при функционировании мобильной СКЗ определяется скоростью перемещения таких объектов в воздушном пространстве.

Практика мониторинга воздушного пространства показывает, что часто СКЗ должна одновременно распознавать объекты разных размеров (масштабов). Особую сложность для обнаружения и классификации представляют летающие объекты малых размеров. Например, птицы и БПЛА имеют малые размеры и часто их контуры весьма близки, поэтому их трудно отличить. Сегодня, несмотря на общие успехи в достижении высокой точности детектирования объектов с помощью моделей СНС, все ещё сохраняется значительный разрыв между точностью детектирования объектов малого и большого масштабов.

Пусть на основе модели СНС из класса YOLO проектируется мобильная СКЗ, которая должна позволять детектировать на изображениях летающие объекты, следующих четырёх классов:

- Птица (птицы);
- БПЛА самолётного типа;

- БПЛА вертолётного типа;
- Неизвестные объекты.

При этом предлагается по размеру (по площади) эти летающие объекты на изображениях разделить на три категории:

- Малые – до 32×32 пикселей;
- Средние – от 33×33 до 96×96 пикселей;
- Большие – от 97×97 пикселей и более.

Рассмотрим количественные характеристики всех указанных выше требований к мобильной СКЗ и, соответственно, к лежащей в её основе той или иной модели СНС. Точность детектирования (классификации) объектов на изображении вычисляется по общепринятым метрикам. В наших исследованиях будем использовать метрики Precision — выражение (1), Recall — выражение (2), AP — площадь под кривой Precision-Recall (AUC-PR), оценённая с использованием порогового значения IoU [21], mAP — среднее значение AP по всем классам объектов (выражение 3) и метрика F1 — выражение (4), которая является гармоническим средним Precision и Recall [22].

$$Precision = \frac{TP}{TP + FP}, \quad (1)$$

где TP — истинно-положительные значения;
 FP — ложно-положительные значения.

$$Recall = \frac{TP}{TP + FN}, \quad (2)$$

где TP — истинно-положительные значения;
 FN — ложно-отрицательные значения.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (3)$$

где N — количество классов объектов.

Для $AP_{0,5}$ и $mAP_{0,5}$ значение IoU равно 0,5, в то время как для расчёта $AP_{0,5:0,95}$ и $mAP_{0,5:0,95}$ используется интервал значений IoU $\in [0,5; 0,95]$ с шагом

0,05. Именно эти метрики наиболее часто используются в работах других авторов, посвящённых моделям СНС класса YOLO.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Будем считать, что точность детектирования (классификации) летающих объектов на изображениях с помощью модели СНС высока (требование к точности детектирования выполняется), если значение метрики $AP_{0,5}$ для отдельных классов объектов и усреднённой метрики $mAP_{0,5}$ больше или равно 0,9 в случае объектов на изображениях разных масштабов или если значение метрики $AP_{0,5}$ для отдельных классов объектов и усреднённой метрики $mAP_{0,5}$ больше или равно 0,7 в случае наличия на изображениях объектов только малых размеров.

Следующей количественной характеристикой мобильной СКЗ является скорость детектирования объектов на изображении (скорость вычислений модели СНС), измеряемая как число анализируемых изображений в секунду в FPS (англ. Frames Per Second). Значение FPS, позволяющее считать, что мобильная СКЗ удовлетворяет требованию по скорости детектирования объектов с помощью модели СНС, должно быть не менее 25. Отметим, что пороговое значение FPS определяется масштабом реального времени, в котором работает мобильная СКЗ. В свою очередь, масштаб реального времени задаётся исходя из скорости перемещения летающих объектов. У многих мобильных СКЗ пороговые значения FPS зависят от особенностей решаемых прикладных задач и обычно находятся в интервале 25-30.

Последняя количественная характеристика — это алгоритмическая эффективность модели СНС. Учитывая, что она определяется как количество ресурсов вычислительного устройства, необходимое для достижения определённых показателей модели СНС в решении поставленной задачи, следует при оценке алгоритмической эффективности модели СНС рассчитывать две характеристики: вычислительная сложность и размер модели СНС. Вычислительная сложность модели СНС, в GFLOPs — это количество операций умножения,

сложения и сравнения над 16-битными числами с плавающей точкой. Размер модели (компактность архитектуры) СНС, в МБ — это объем памяти вычислительного устройства СКЗ, необходимый для хранения весовых коэффициентов свёрточных фильтров и промежуточных буферов при вычислении модели СНС. Пороговые значения вычислительной сложности и размера модели СНС, позволяющие считать, что мобильная СКЗ (и, соответственно, модель СНС в её основе) удовлетворяет требованиям разработчиков её вычислительного устройства, задаются в техническом задании на создание СКЗ. При проведении исследований моделей СНС будем оценивать значения этих двух характеристик, ориентируясь на некоторые усреднённые значения параметров современных и широко распространённых графических процессоров (например, компании Nvidia), систем на кристалле с программируемыми логическими интегральными схемами (ПЛИС) (например, компании Xilinx) и подобных им вычислительных устройств, если иное не оговорено в техническом задании на СКЗ.

1.4 Анализ моделей СНС для детектирования объектов

Ниже приведены результаты анализа моделей СНС с целью выбора моделей, отвечающих хотя бы некоторым требованиям со стороны мобильных СКЗ для детектирования летающих объектов.

1.4.1 Первые модели СНС для детектирования объектов

Ранние алгоритмы детектирования объектов на изображении на основе глубокого обучения адаптировали классификаторы для решения данной задачи. СНС для классификации применялась последовательно к различным областям изображения по методу скользящего окна [23], либо, в более поздних работах, к определённым образом сгенерированным областям интереса на изображении [5]. Высокая оценка классификатора в некоторой области изображения расценивалась как детекция объекта в данной области. Такие двухступенчатые архитектуры СНС для детектирования объектов являются сложными и медленными — для обработки одного изображения требуется от нескольких сотен до не-

скольких тысяч проходов сети по различным областям изображения, в связи чем такие архитектуры не пригодны к использованию в мобильных СКЗ.

Одним из наиболее широко используемых подходов к детектированию объектов на изображениях является использование модели R-CNN (англ. Region-based Convolutional Neural Network), которая была предложена в 2014 году. Модель R-CNN состоит из двух основных частей: первой — извлечение предложений (англ. Region Proposal Network, RPN), чтобы выделить области на изображении, которые могут содержать объекты; и второй — классификация и локализация объектов. Эта модель показала высокую точность детектирования, но имела низкую скорость обработки изображения [24].

Модель Fast R-CNN была разработана для решения проблемы низкой скорости модели R-CNN. Она использует единую свёрточную сеть для извлечения признаков и классификации объектов, а также объединяет RPN и детектор объектов в единую архитектуру. Модель Fast R-CNN показала более высокую скорость в сравнении с R-CNN при сохранении высокого уровня точности в детектировании объектов [24].

Следующая модель Mask R-CNN является расширением Faster R-CNN, которая позволяет не только детектировать объекты, но и производить сегментацию изображений. Таким образом, она позволяет определить не только местоположение объекта, но и его форму.

Несмотря на то, что перечисленные модели дают весьма высокую точность детектирования объектов, скорость их вычислений, даже у улучшенных моделей Fast R-CNN и Mask R-CNN, весьма низка. Кроме того, их алгоритмическая эффективность весьма невысока. Все это не позволяет рассматривать такие модели в качестве основы при создании мобильных СКЗ, а тем более функционирующих в масштабе реального времени.

1.4.2 Модели СНС класса YOLO

Модели класса (часто говорят, семейства) YOLO (англ. You Only Look Once) — это ещё одни популярные модели для детектирования объектов, которые также способны решать задачи сегментации изображений. Исследователи,

предложившие модель YOLO [10], одними из первых применили другой, не классический подход к решению задачи детектирования: они рассматривали детектирование как задачу регрессии рамок (прямоугольников, которые ограничивают объекты) и классов объектов. Модели СНС класса YOLO, как и другие модели СНС с одноступенчатой архитектурой, обрабатывают изображение в исходном масштабе за один проход. Преимущества таких одноступенчатых моделей СНС в сравнении с двухступенчатыми моделями в следующем:

- предсказания основаны на глобальном контексте, т. к. модель обрабатывает сразу всё изображение;
- предсказание рамок и классов всех объектов на изображении за один проход по сети значительно ускоряет процесс детектирования.

Модели СНС для предсказания рамок и классов делятся на одноступенчатые и двухступенчатые, якорные и безъякорные (таблица 1).

Таблица 1 — Модели СНС для предсказания ограничивающих прямоугольников и классов

	Якорные	Безъякорные
Одноступенчатые	SSD [9], YOLO [10], RetinaNet [25]	CornerNet [15, 26], CenterNet [27], MatrixNet [28], FCOS [17]
Двухступенчатые	R-CNN [5], Fast R-CNN [29], Faster R-CNN [30], Mask R-CNN [6], Libra R-CNN [31], R-FCN [32]	RepPoints [33]

Модель YOLO — современный алгоритм глубокого обучения, который широко используется для обнаружения объектов. Он был разработан Джозефом Редмоном и Али Фархади в 2016 году [34]. Именно эта модель положила начало классу моделей YOLO, созданных в последующие годы.

Основное отличие модели YOLO от других моделей СНС, используемых для обнаружения объектов, заключается в том, что она очень быстро опознает объекты в режиме реального времени. Принцип работы модели YOLO подразумевает ввод сразу всего изображения, которое проходит через СНС только один раз, в то время как в других алгоритмах этот процесс происходит много-

кратно. Модель YOLO обладает преимуществом высокоскоростного обнаружения объектов, чем не могут похвастать другие алгоритмы [34].

Модели СНС класса YOLO могут находить несколько ограничивающих прямоугольников для одного и того же объекта, поэтому для выбора лишь того, который максимально точно описывает объект, используется формула расчёта IoU и алгоритм не максимального подавления (англ. Non-Maximum Suppression, NMS) [21].

Модель YOLOv2 была выпущена в 2017 году и получила почётное упоминание на CVPR 2017 [11]. В её архитектуру был внесён ряд итерационных улучшений поверх исходной модели YOLO, включая BatchNorm, более высокое разрешение и блоки привязки.

Модель YOLOv3 была разработана в 2018 году и основывалась на предыдущих моделях (Рис. 1), добавляя оценку объектности к прогнозированию ограничивающего прямоугольника, добавляя подключения к слоям магистральной сети и делая прогнозы на трёх отдельных уровнях детализации для повышения производительности модели на более мелких объектах [12].

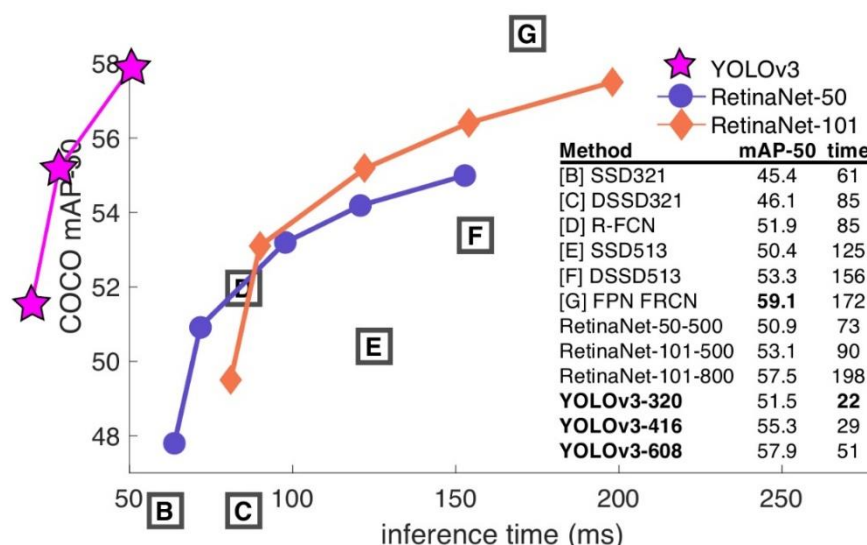


Рисунок 1 — Сравнение на датасете MS COCO метрики mAP и скорости обработки изображений с помощью модели YOLOv3 с другими моделями СНС

После выпуска СНС YOLOv3 Джозеф Редмон отошёл от исследований в области компьютерного зрения. Однако такие исследователи, как Алексей Бочковский, и такие новаторы, как Гленн Джохер, начали открывать исходный

программный код о своих достижениях в исследованиях компьютерного зрения. Такие группы, как Baidu, также выпустили свои собственные реализации модели YOLO (например, PP-YOLO), демонстрирующие улучшение значений метрики mAP и снижение времени задержки на вычисление модели [35].

Модель СНС YOLOv4 — это одноступенчатая (часто говорят, одноэтапная) модель обнаружения и классификации объектов, созданная на основе предыдущих оригинальных моделей YOLO. Современные детекторы объектов на изображениях обычно состоят из двух компонентов: основы (backbone) и головы (head) (Рис. 2), что имеет место и для этой модели.

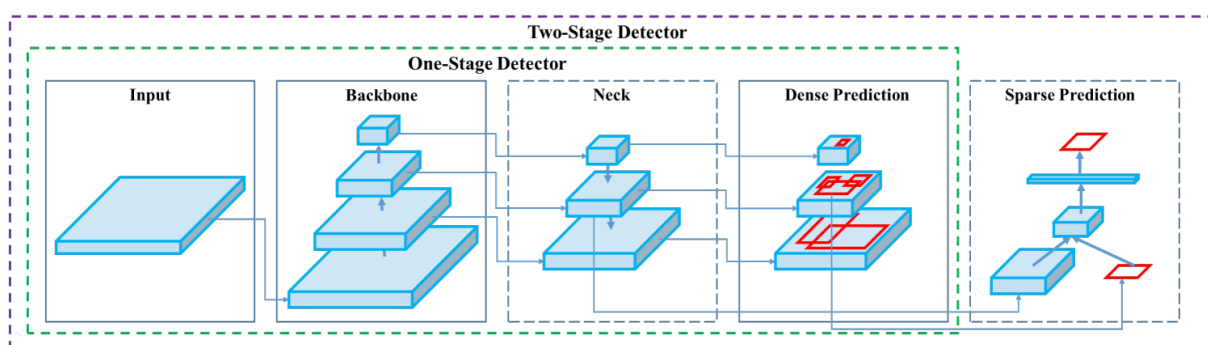


Рисунок 2 — Архитектура моделей СНС в общем виде

Основа обычно предварительно обучена на большом наборе данных для классификации изображений, например, на ImageNet, и служит для кодирования соответствующей информации о входных данных. Голова предсказывает классы объектов и информацию об ограничивающем прямоугольнике. Также, в модели YOLOv4 выделяют особый слой — шею (англ. neck), который является связующим между слоями основы и головы. Он служит для сбора карт признаков с разных этапов сети [36] и повышения скорости обработки изображений (Рис. 3). Модель YOLOv4 использует новые функции: WRC, CSP, CmBN, SAT, Mish-активация, увеличение данных Mosaic, CmBN, регуляризация DropBlock и потери CIoU [13]. Модель YOLOv4, как и модель YOLOv3, реализована с помощью фреймворка Darknet [37].

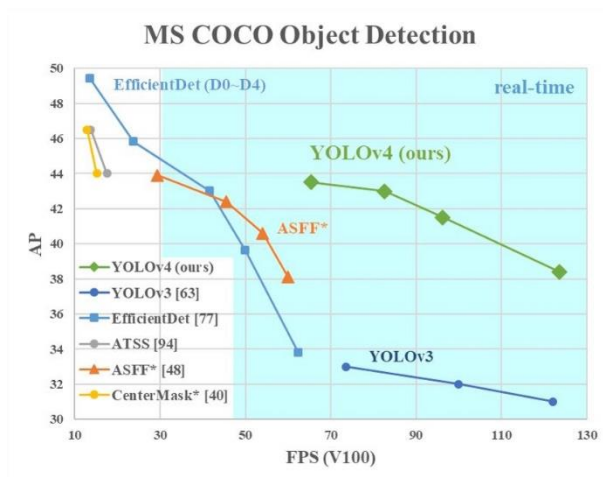


Рисунок 3 — Сравнение на датасете MS COCO метрики mAP и скорости обработки изображений с помощью модели YOLOv4 с другими моделями СНС

Модель СНС YOLOv5 имеет семейство архитектур для обнаружения и классификации объектов, предварительно обученных на наборе данных MS COCO (Рис. 4). Это единственный детектор объектов из класса YOLO, не имеющий научной статьи, в которой обосновывается улучшение архитектуры по сравнению с предыдущими версиями YOLO. Отсутствие научной статьи обусловлено тем, что данная архитектура разрабатывается сообществом. Модель СНС YOLOv5 реализована средствами фреймворка PyTorch, устраняя ограничения фреймворка Darknet (основанного на языке программирования С и не созданного с учётом выполнения в производственных средах) [38, 39].

Модель YOLOv5 является одной из официальных современных моделей, которая предоставляет возможность модификации архитектуры, тонкую настройку, тестирование и развёртывание на различных целевых платформах. Как и модель YOLOv4, модель YOLOv5 использует кросс-этапные частичные соединения с Darknet-53 в качестве основы и Path Aggregation Network в качестве шеи. Основные усовершенствования этой модели включают в себя новое увеличение мозаичных данных и автообучаемые привязки к ограничивающим прямоугольникам [38].

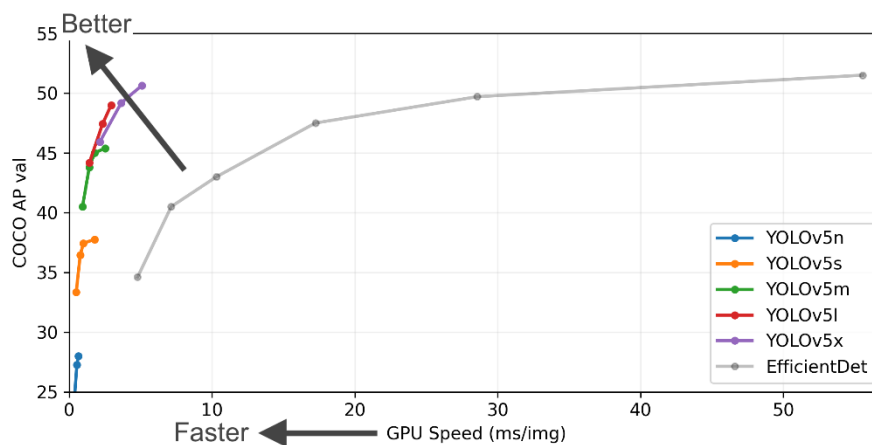


Рисунок 4 — Сравнение на датасете MS COCO метрики mAP и скорости обработки изображений с помощью модели YOLOv5 с другими моделями СНС

Модель СНС YOLOv7 является идейным наследником модели YOLOv4 и вносит в её архитектуру новые возможности: вычислительный блок магистрали E-ELAN и тренируемый блок BoF (англ. Bag of Freebies) [40]. В отличие от модели YOLOv4, модель YOLOv7 реализована с помощью фреймворка PyTorch [41]. Возможности её показаны на рисунке 5.

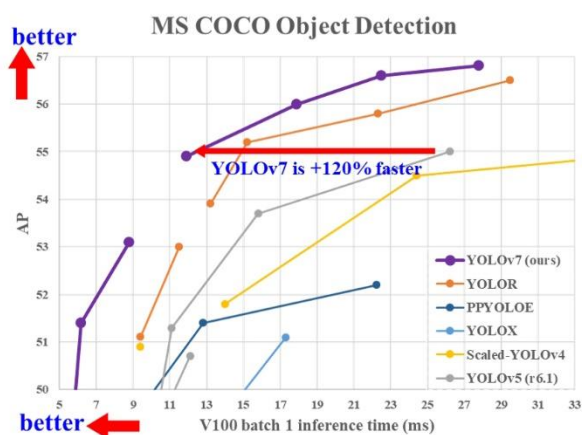


Рисунок 5 — Сравнение на датасете MS COCO метрики mAP и скорости обработки изображений с помощью модели YOLOv7 с другими моделями СНС

Обычно модели СНС с одноступенчатой архитектурой при детектировании объектов вычисляются со скоростью, достаточной для функционирования мобильных СКЗ, имеющих в своём составе вычислительное устройство в виде современного графического процессора. Однако большинство таких моделей, включая многие модели класса YOLO, хоть и значительно быстрее и эффективнее двухступенчатых, все же не подходят в качестве основы при создании мно-

гих мобильных СКЗ из-за жёстких ограничений ресурсов памяти и производительности вычислительных устройств таких СКЗ.

В связи с этим были предложены на основе рассмотренных моделей СНС класса YOLO модели с уменьшенными наборами свёрточных слоёв, получившие название СНС Tiny YOLO [10]. При их вычислениях предъявляются значительно меньшие требования к памяти и производительности вычислительного устройства СКЗ. Однако, как показали исследования, точность таких моделей ниже полноразмерных моделей YOLO [10, 11]. Отметим, что кроме моделей Tiny YOLO одной из компактных архитектур СНС для детектирования объектов является модель СНС YOLO Nano [20]. К сожалению, она показывает ещё меньшую точность детектирования по сравнению с моделями СНС Tiny YOLO.

1.4.3 Компактные модели СНС из класса YOLO

Проведённый анализ современных моделей класса YOLO [41, 35] показал, что наиболее перспективными для достижения компромисса при удовлетворении требований к СКЗ из подраздела 1.3 является подмножество моделей YOLOv4, YOLOv5 и YOLOv7 из этого класса.

Более того, на наш взгляд, исследованию и выявлению наиболее эффективных из этого подмножества подлежат компактные (малое число свёрточных слоёв) модели СНС: YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny. Во-первых, компактные модели СНС действительно имеют меньшее число свёрточных слоёв по сравнению с полноразмерными моделями из этого подмножества (таблица 2 и таблица 3), что напрямую влияет на скорость вычисления моделей СНС и на их алгоритмическую эффективность. Из таблицы 3 следует, что эти компактные модели СНС имеют весьма высокую алгоритмическую эффективность по сравнению с другими моделями из анализируемого подмножества моделей класса YOLO.

Таблица 2 — Основные характеристики компактных моделей СНС класса YOLO

Характеристики	YOLOv4 Tiny	YOLOv5 Small	YOLOv7 Tiny
Количество всех слоёв	37	49	78

Характеристики	YOLOv4 Tiny	YOLOv5 Small	YOLOv7 Tiny
Количество свёрточных слоёв	21	35	55
Функция активации	Mish	SiLU	LeakyReLU
Функция оптимизации	SGD	SGD	SGD
Функция потерь	CIoU	BCEWithLogitsLoss	BCEWithLogitsLoss

Во-вторых, выбранные для исследования модели СНС имеют довольно хорошие показатели по метрикам AP/mAP, например, для датасета MS COCO [43] имеем:

- СНС YOLOv4 Tiny — $AP_{0,5} \approx 42,1\%$ [13, 37, 14];
- СНС YOLOv5 Small — $mAP_{0,5} \approx 56,8$ [39];
- СНС YOLOv7 Tiny — $AP_{0,5} \approx 52,8\%$ [41, 14].

Таблица 3 — Размеры и вычислительная сложность (количество операций) моделей СНС класса YOLO [41, 35]

Модель СНС	Размер модели, МБ	Количество операций, GFLOPs
YOLOv4 Tiny	23,1	6,9
YOLOv4	245,0	60,1
YOLOv5 Nano	3,9	4,5
YOLOv5 Small	14,1	16,5
YOLOv5 Medium	40,8	49,0
YOLOv7 Tiny	11,6	5,8
YOLOv7	72,0	105,2

1.5 Цель и задачи исследований

Как сказано выше, сегодня весьма востребованы мобильные (возимые или носимые, включая встраиваемые системы) СКЗ на основе современных моделей СНС. В качестве моделей СНС для мобильной СКЗ, предназначенной для детектирования летающих объектов на изображениях, будем использовать выбранные в результате анализа в разделе 1.4.3 три компактные модели СНС из класса YOLO: YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny. Именно они подлежат исследованию на соответствие требованиям к мобильной СКЗ, изложен-

ным в подразделе 1.3. В итоге можно сформулировать следующие цель и задачи исследований по ВКР.

Целью данной ВКР является разработка и исследование моделей СНС класса (часто говорят, семейства) YOLO для детектирования летающих объектов различных размеров (масштабов) на изображениях и выбор в результате исследований, в том числе с учётом категории объектов по размерам, наиболее эффективных моделей СНС из этого класса для создания мобильных СКЗ реального времени.

Для реализации поставленной цели необходимо решить следующие задачи.

1. Сформировать два набора RGB-изображений летающих объектов четырёх классов: «Птица (стая птиц)», «БПЛА самолётного типа», «БПЛА вертолётного типа» и «Неизвестный объект». На каждом изображении первого набора содержится один и более объектов перечисленных классов, причём объекты будут иметь разные размеры (разномасштабные объекты). Часть изображений второго набора содержит только малоразмерные объекты, но объекты относятся к любому из четырёх классов, вторая, равная первой, часть изображений этого набора содержит изображения со среднемасштабными объектами из четырёх классов, а третья равная часть из этого набора — изображения с крупномасштабными объектами любого из четырёх классов. При формировании второго набора следует руководствоваться размерами каждой категории объектов, изложенными в подразделе 1.3. Объем изображений во втором наборе должен в три раза превышать объем первого набора.

2. Разметить и аннотировать объекты указанных классов на RGB-изображениях наборов и сформировать в итоге четыре датасета (первый, названный основным, датасет формируется из изображений первого набора и три дополнительных датасета — из изображений второго набора: по одному датасету из изображений, содержащих объекты каждой категории по размерам) для обучения, валидации и исследования разработанных моделей СНС класса YOLO.

3. Провести разработку и обучение выбранных в результате анализа трёх компактных моделей СНС с использованием обучающих и валидационных выборок из подготовленных датасетов.

4. Исследовать эффективность обученных компактных моделей СНС с использованием тестовых выборок каждого из четырёх датасетов. По результатам исследований выявить наиболее эффективную компактную модель СНС.

5. Разработать на основе наиболее эффективной модели новую модель СНС класса YOLO с использованием Inception – ResNet-модулей и механизма пробросов признаков.

6. Провести обучение новой модели СНС с использованием подготовленных обучающих и валидационных выборок каждого из трёх датасетов, сформированных из изображений второго набора.

7. Исследовать эффективность каждого варианта обученной новой модели СНС с использованием тестовой выборки в каждом из трёх датасетов.

8. Провести анализ всех результатов комплексных исследований трёх компактных и новой моделей СНС и с учётом ранее сформированных требований к мобильной СКЗ реального времени выбрать наиболее подходящую модель СНС для реализации в составе такой СКЗ для детектирования летающих объектов разных масштабов на изображениях.

9. Сформировать рекомендации по использованию разработанных моделей СНС в мобильных системах компьютерного зрения.

ГЛАВА 2. СОЗДАНИЕ ДАТАСЕТОВ ДЛЯ ОБУЧЕНИЯ И ИССЛЕДОВАНИЯ МОДЕЛЕЙ СНС

2.1 Описание используемых форматов датасетов

Датасеты для обучения, валидации и тестирования выбранных компактных моделей СНС YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny формировались на основе большого количества RGB-изображений летающих объектов четырёх указанных выше классов. Размер изображений 416×416 пикселей на входе моделей СНС.

Формирование каждого из четырёх датасетов происходило за счёт объединения размеченных другими авторами наборов изображений (сторонних датасетов) и ручного сбора и разметки изображений, удовлетворяющих требованиям, изложенным в подразделе 1.5. Создаваемые датасеты предназначены для обучения и исследования трёх компактных моделей СНС, разработанных с использованием двух разных фреймворков Darknet и PyTorch. В связи с этим файлы аннотаций, которые описывают размеченные изображения, будут представлены в различных форматах, либо вовсе отсутствуют.

В подготовленных другими авторами датасетах, являющихся фрагментами создаваемых нами датасетов, чаще всего используются следующие форматы аннотаций:

- TensorFlow Object Detection CSV;
- YOLO Keras TXT.

Исследуемые модели СНС класса YOLO используют схожие форматы файлов аннотаций — YOLO Darknet TXT [44] и YOLO PyTorch TXT [45]. Отличия между форматами YOLO Darknet TXT и YOLO PyTorch TXT состоит в том, что для YOLO Darknet TXT необходим дополнительный файл, описывающий все используемые классы объектов из датасета, в то время как YOLO PyTorch TXT описывает классы в файле конфигурации датасета.

В связи с этим необходимо унифицировать все изображения в датасете, причём как размеченные другими авторами наборы изображений, так и собранные вручную и размеченные изображения. В рамках унификации изображений,

которая является по сути предварительной обработкой данных, необходимо решить следующие задачи:

1. Преобразовать аннотации к изображениям из формата TensorFlow Object Detection CSV в формат YOLO PyTorch TXT.
2. Преобразовать аннотации к изображениям из формата YOLO Keras TXT в формат YOLO PyTorch TXT.
3. Определить ограничивающие прямоугольники объектов для тех наборов изображений, которые не имеют аннотаций, и сохранить их в формате YOLO PyTorch TXT.

Изображения, собранные вручную, необходимо разметить в формате YOLO PyTorch TXT или другом, который можно преобразовать в данный формат.

Формат YOLO PyTorch TXT представляет собой текстовый файл, в котором на каждой строке описаны все объекты определённого изображения в следующем порядке:

- ID класса объекта;
- Нормализованный центр объекта по оси X;
- Нормализованный центр объекта по оси Y;
- Нормализованная ширина объекта;
- Нормализованная высота объекта.

На рисунке 6 показано визуальное представление того, что описывает каждая строка данного формата аннотаций.

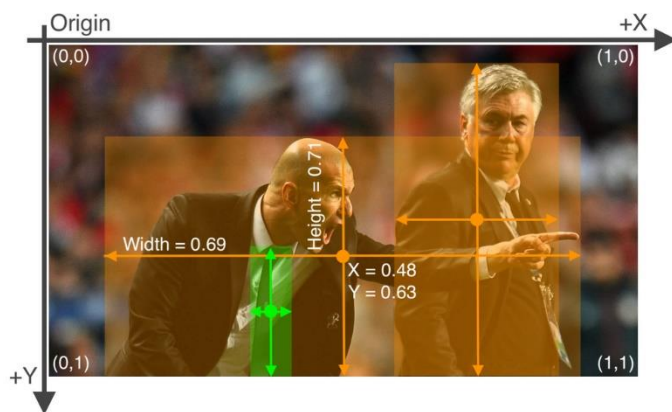


Рисунок 6 — Пример размеченных объектов на изображении

Для проведения описанной выше предварительной подготовки изображений разрабатывалось программное обеспечение на языке программирования Python 3 в среде Anaconda с использованием следующих библиотек: OpenCV, Numpy, Pandas, Shapely, Scikit Learn, YAML и стандартных библиотек Python.

2.2 Преобразование аннотаций из формата TensorFlow Object Detection CSV в формат YOLO PyTorch TXT

Формат аннотаций TensorFlow Object Detection CSV [46] представлен следующим образом:

```
filename,width,height,class,xmin,ymin,xmax,ymax
10.png,2500,1667,bird,1173,1210,1262,1247
10.png,2500,1667,bird,561,1116,684,1164
10.png,2500,1667,bird,774,1107,872,1167
10.png,2500,1667,bird,1048,1087,1156,1142
10.png,2500,1667,bird,1355,946,1470,1022
```

Каждая строка описывает один объект, его класс, координаты начала и конца по осям X и Y. Для преобразования такой строки в необходимый формат была разработана функция на языке Python 3 с использованием библиотеки Pandas:

```
def convert_tf_record(series):
    center_x = (series['xmin'] + series['xmax']) / 2
    center_y = (series['ymin'] + series['ymax']) / 2
    size_x = series['xmax'] - series['xmin']
    size_y = series['ymax'] - series['ymin']

    center_x /= series['width']
    center_y /= series['height']
    size_x /= series['width']
    size_y /= series['height']

    return (
        classes.index(entry['class']),
        clamp(0.0, 1.0, center_x),
        clamp(0.0, 1.0, center_y),
        clamp(0.0, 1.0, size_x),
        clamp(0.0, 1.0, size_y)
    )
```

Функция `clamp` ограничивает значения параметра в интервале $V \in [F; T]$ и рассчитывается с использованием выражения:

$$\text{Clamp}(F, T, V) = \text{Max}(F, \text{Min}(T, V)), \quad (5)$$

где F — минимальное допустимое значение;
 T — максимальное допустимое значение;
 V — текущее ограничиваемое значение.

Собранные аннотации не содержат данные о размерах изображений и о названии файла, т.к. данные о размере изображения вычисляются при помощи фреймворка PyTorch, а название файла изображения переносится в название файла аннотаций.

2.3 Преобразование аннотаций из формата YOLO Keras TXT в формат YOLO PyTorch TXT

Формат аннотаций YOLO Keras TXT [47] представлен следующим образом:

```
01.png 120,22,742,758,4
02.jpg 440,53,732,435,4
03.jpg 457,264,661,574,4
04.jpg 202,437,410,525,4 822,275,1212,810,4
05.jpg 160,207,795,635,4
```

Каждая строка описывает один файл изображения и объекты на изображении. Все параметры в такой строке разделены символом пробела. Каждый набор параметров разделён символом запятой, и параметры расположены в следующем порядке:

- Начало ограничивающего прямоугольника по оси X;
- Начало ограничивающего прямоугольника по оси Y;
- Конец ограничивающего прямоугольника по оси X;
- Конец ограничивающего прямоугольника по оси Y;
- Класс объекта.

Для преобразования данных каждой строки в необходимый формат была разработана функция на языке Python 3 с использованием библиотеки OpenCV:

```
def convert_keras_record(string):
    filename, other = string.strip().split(' ', 1)
    params = [list(map(int, entry.split(','))) for entry in other.split('
')]
    ')]
```



```

    image = cv2.imdecode(np.fromfile(join(dirname, filename), np.uint8),
cv2.IMREAD_UNCHANGED)
    height, width, _ = image.shape
    result = []

    for (xmin, ymin, xmax, ymax, entity_class) in params:
        center_x = (xmin + xmax) / 2
        center_y = (ymin + ymax) / 2
        size_x = xmax - xmin
        size_y = ymax - ymin

        center_x /= width
        center_y /= height
        size_x /= width
        size_y /= height

        result.append((
            entity_class,
            clamp(0.0, 1.0, center_x),
            clamp(0.0, 1.0, center_y),
            clamp(0.0, 1.0, size_x),
            clamp(0.0, 1.0, size_y)
        ))

    return result

```

В связи с тем, что строка аннотации в формате YOLO Keras TXT может содержать больше одного объекта изображения, необходимо возвращать массив аннотаций из функции.

2.4 Определение ограничивающих прямоугольников

Наборы изображений, не имеющих файлов аннотаций, представлены двумя подмножествами RGB-изображений: исходные изображения и изображения масок объектов (Рис. 7). В данной задаче необходимо разработать алгоритм, позволяющий вычислить ограничивающий прямоугольник объекта с использованием маски объекта.

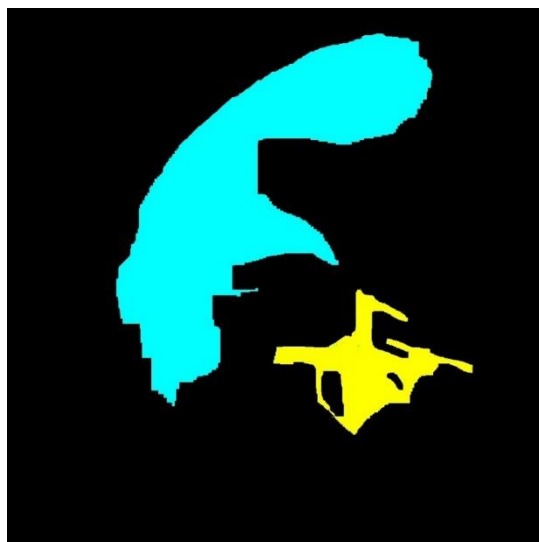


Рисунок 7 — Пример исходного изображения и соответствующей ему маски

Маски, описывающие контуры объектов, представлены разными цветами таким образом, что каждый определённый цвет характеризует собой определённый класс объекта:

- Птица — голубой;
- БПЛА самолётного типа — синий;
- БПЛА вертолётного типа — жёлтый;
- Неизвестные объекты — фиолетовый.

Для решения данной задачи разработан оригинальный алгоритм, описываемый пошагово в виде укрупнённых блоков:

Начало

Шаг 1. Определить цветовые интервалы масок на изображении;

Шаг 2. Определить размер морфологических преобразований;

Шаг 3. Для каждого цветового интервала маски:

Шаг 3.1. Вычислить изображение, соответствующее цветовому интервалу маски;

Шаг 3.2. Конвертировать изображение в градации серого;

Шаг 3.3. Конвертировать изображение в черно-белое;

Шаг 3.4. Применить размытие к изображению;

Шаг 3.5. Применить морфологические преобразования к изображению;

Шаг 3.6. Вычислить контуры объектов на изображении;

Шаг 3.7. Для всех пар контуров, найти те контуры, чьи ограничивающие прямоугольники не являются полным пересечением других ограничивающих прямоугольников;

Шаг 3.8. Для всех пар контуров, найти те контуры, чьи ограничивающие прямоугольники не пересекаются с другими ограничивающими прямоугольниками;

Шаг 3.9. В том случае, если не было найдено контуров в шагах 3.7 и 3.8, то необходимо учитывать все контуры, найденные на шаге 3.6;

Шаг 3.10. Вычислить ограничивающие прямоугольники для всех оставшихся контуров;

Шаг 3.11. Сохранить вычисленные ограничивающие прямоугольники в файл.

Шаг 4. Если не все цветовые интервалы маски обработаны, то на Шаг 3, иначе на Конец.

Конец

Детальная схема алгоритма представлена в Приложении Б.

Для разработки программы, реализующей этот алгоритм, использован язык программирования Python 3 в среде Anaconda со следующими библиотеками: OpenCV, Shapely, Numpy, Pandas, Scikit Learn.

Применение морфологического преобразования требует определения размера структурирующего элемента. При нахождении ограничивающих прямоугольников объекта было выяснено, что для изображений, имеющих объекты больших размеров, нужны меньшие размеры структурирующего элемента, в то время как для изображений с объектами, имеющими малые размеры, необходимы более крупные размеры структурирующего элемента. Для определения морфологических преобразований была построена модель логистической регрессии. Данные для обучения модели были подобраны вручную (Рис. 8) на основе результатов работы алгоритма по определению ограничивающих прямоугольников.

	width	height	area	width_height_ratio	height_width_ratio	target
0	4096.0	4096.0	16777216.0	1.000000	1.000000	3
1	2560.0	1622.0	4152320.0	1.578298	0.633594	3
2	2048.0	1266.0	2592768.0	1.617694	0.618164	6
3	2048.0	2048.0	4194304.0	1.000000	1.000000	6
4	775.0	518.0	401450.0	1.496139	0.668387	12
5	512.0	512.0	262144.0	1.000000	1.000000	12

Рисунок 8 — Пример данных для обучения логистической регрессии

На рисунке 9 в качестве примера представлено изображение маски, на котором необходимо найти границы двух объектов.

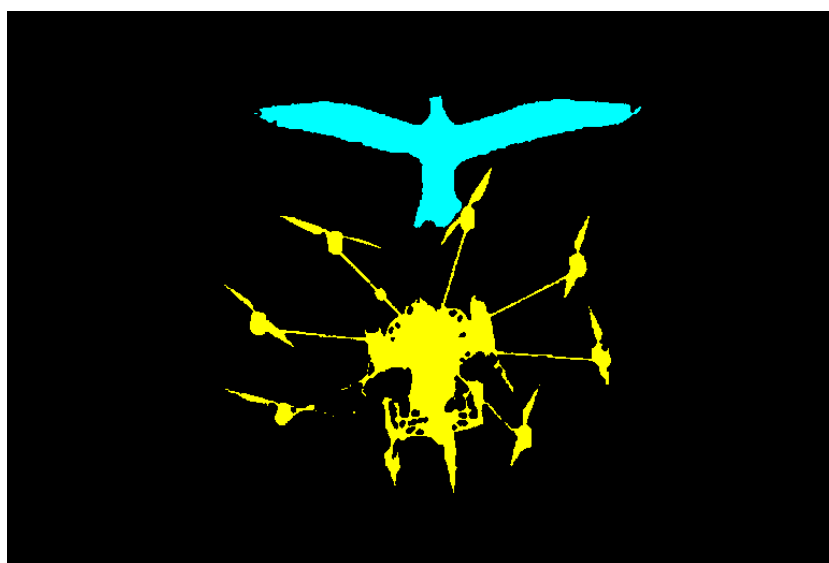


Рисунок 9 — Изображение маски для двух объектов

На рисунках 10, 11, 12, 13 и 14 представлено пошаговое преобразование изображения для маски объекта класса «БПЛА вертолётного типа».

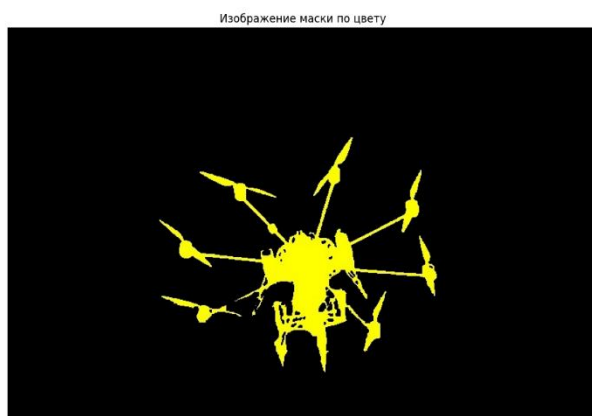


Рисунок 10 — Вычисленная маска класса



Рисунок 11 — Вычисленная маска в градациях серого



Рисунок 12 — Черно-белая вычисленная маска



Рисунок 13 — Размытая вычисленная маска



Рисунок 14 — Вычисленная маска с морфологическими преобразованиями

Изображение с применёнными к нему морфологическими преобразованиями лучше подходит для нахождения контуров объекта, т.к. большинство мелких деталей после преобразования были соединены между собой.

На рисунках 15 и 16 представлены примеры визуализации найденных ограничивающих прямоугольников на изображениях маски.

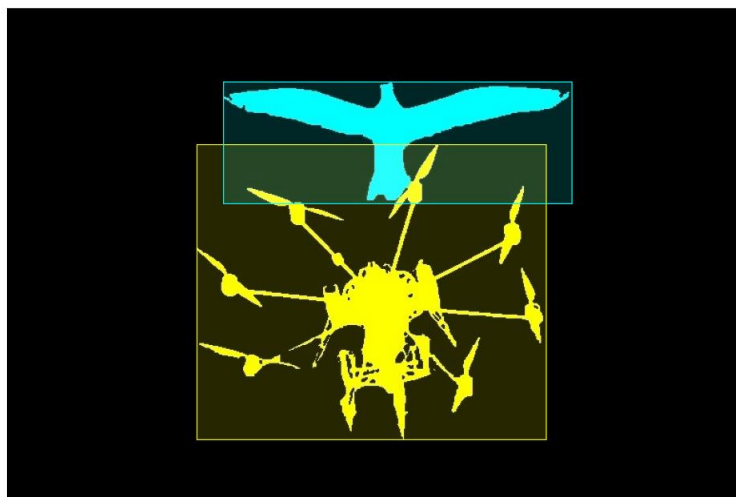


Рисунок 15 — Пример найденных границ объектов разных классов на изображении маски

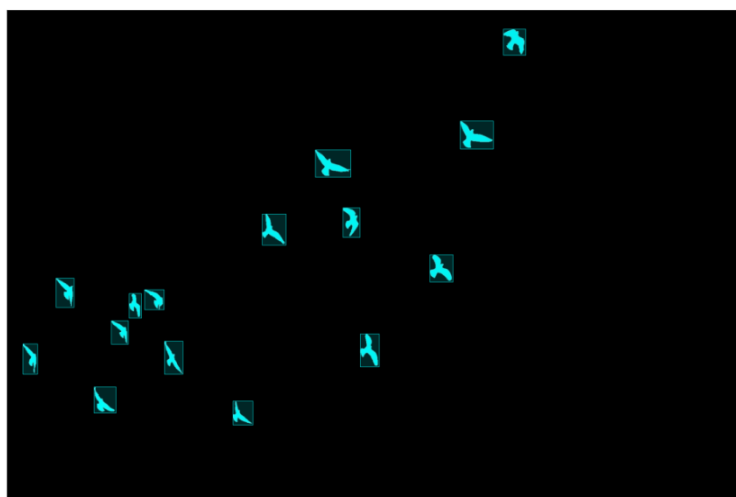


Рисунок 16 — Пример найденных границ объектов одного класса на изображении маски

Исходный код программы нахождения ограничивающих прямоугольников представлен в Приложении В. Исходный код программы обучения логической регрессии представлен в Приложении Г.

2.5 Формирование основного датасета

После того, как все аннотации изображений других авторов были унифицированы и размечены дополнительные изображения, собранные вручную, необходимо было удалить все возможные дубликаты.

Для нахождения дубликатов использовалась программа AllDup [48], которая позволяет сравнить все файлы изображений между собой для последующей группировки и удаления найденных дубликатов (Рис. 17).

Для поиска дубликатов изображений используются следующие исходные критерии:

- Совпадение: 99%;
- Область сравнения: 16×16 пикселей.

При поиске дубликатов изображений используется следующий алгоритм и форматы:

- Алгоритм: dHash;
- Форматы файлов: bmp, gif, jpeg, jpg, png.

Необходимые требования при поиске дубликатов изображений:

- Учёт ориентации;
- Учёт соотношения сторон.

Группа - Похожие изображения	Путь	Дубликатов	Размер (байт)	Совпадение
▲ Результаты поиска		63	545 837 973	
▲ 101.png		2	32 702 146	
101.png	F:\Project\fqw\bulk\images		16 311 675	100%
104.png	F:\Project\fqw\bulk\images		16 390 471	99%
▷ 1036.png		5	6 977 438	
▷ 1040.png		2	2 968 123	
▷ 10882.jpg		3	44 930	
▷ 1193.png		2	1 352 060	
▷ 1196.png		2	1 354 746	
▷ 1206.png		3	1 969 232	
▷ 1214.png		2	1 321 444	
▷ 1233.png		5	9 603 482	
▷ 1239.png		2	3 895 504	
▷ 1267.png		2	2 046 179	
▷ 2154.jpg		2	33 251	
▷ 284.png		3	45 606 230	
▷ 288.png		2	30 680 351	
▷ 291.png		2	30 891 278	
▷ 292.png		2	30 738 849	
▷ 299.png		2	30 678 368	
▷ 300.png		2	30 782 456	
▷ 304.png		2	30 803 924	
▷ 53.png		4	62 041 101	
▷ 59.png		2	31 064 061	
▷ 62.png		3	46 430 928	
▷ 75.png		3	46 944 562	
▷ 82.png		2	32 344 096	
▷ 89.png		2	32 563 234	

Рисунок 17 — Результаты поиска дубликатов изображений

Оставшиеся после удаления дубликатов изображения и файлы аннотаций входят в датасет, который ещё не разделён на обучающий, валидационный и тестовый наборы данных (выборки).

Для формирования основного датасета из нормализованных файлов изображений и аннотаций разработана программа на языке Python 3 с использованием библиотеки Scikit Learn (приведена в Приложении Д). Программа также позволяет разделять в датасете изображения и аннотации каждого класса объектов на три набора данных (обучающий, валидационный и тестовый), соответственно, в пропорции 80%, 10% и 10% от объёма датасета. В связи с тем, что в каждом наборе должен быть одинаковый процент объектов каждого класса, в случаях, когда на изображениях представлены объекты разных классов, такие изображения могут повторяться в наборах данных.

Полученный основной датасет включает в себя 15 235 исходных изображений, содержащих 38 553 объектов, и столько же файлов аннотаций. Каждое изображение датасета содержит один и более объектов разных классов и разных размеров. На рисунках 18 и 19 представлены количественные характеристики изображений по классам объектов и по числу объектов на изображениях в этом датасете.

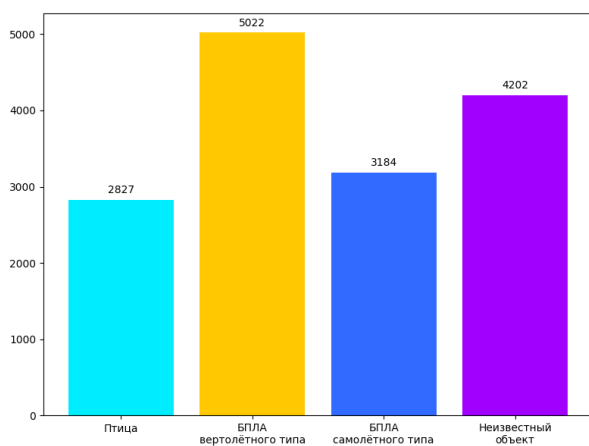


Рисунок 18 — Количество изображений в датасете по классам объектов

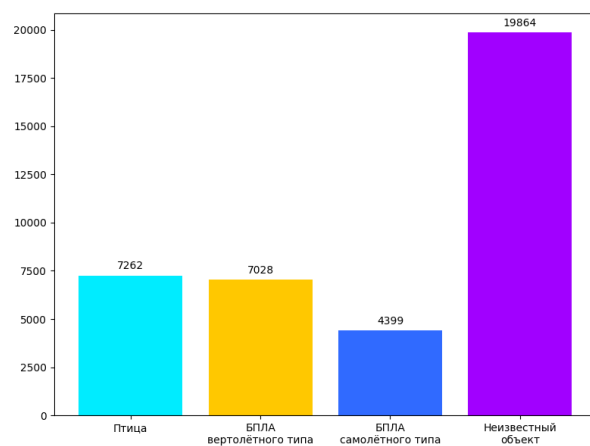


Рисунок 19 — Количество экземпляров объектов в датасете каждого класса

Количество экземпляров объектов в датасете больше в связи с тем, что на одном изображении может быть несколько объектов одного или нескольких классов. На рисунке 20 представлено распределение числа изображений по классам в зависимости от количества объектов на одном изображении. По оси

У приведено количество изображений каждого класса в датасете с соответствующим числом объектов, а по оси X — число объектов на отдельном изображении.

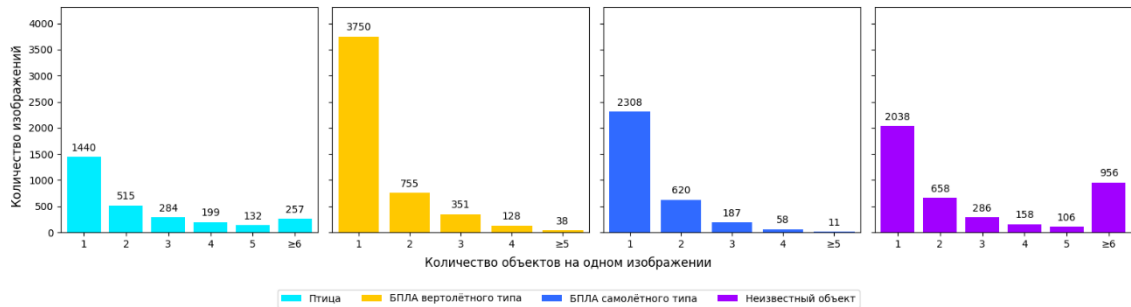


Рисунок 20 — Распределение числа изображений в датасете по классам в зависимости от количества объектов на отдельном изображении

2.6 Формирование трёх дополнительных датасетов

Для исследования точности детектирования объектов на изображении и скорости вычисления моделей СНС в зависимости от размеров объектов создавались ещё три датасета, названных дополнительными датасетами. Для этого использовались два подхода. В соответствии с первым подходом дополнительные датасеты формировались только на основе размеченных изображений основного датасета. При этом в первый дополнительный датасет попадали изображения с объектами только малых размеров (первая категория объектов, см. подраздел 1.3), но объекты могли быть разных классов. Во второй датасет вошли размеченные изображения из основного датасета, имеющие объекты только средних размеров (объекты второй категории). В третий датасет попали изображения только с большемерными объектами (третья категория объектов). В итоге файлы аннотаций изображений, входящих в датасет с объектами определённой категории размеров, содержат данные о размеченных объектах с размерами, соответствующими только этой заданной категории. В том случае, если изображение содержит объекты нескольких категорий размеров, то в аннотациях не указываются объекты, не входящие в формируемую категорию.

Поскольку объем каждого из полученных первым способом дополнительных датасетов невелик и не удовлетворяет требованию из подраздела 1.5 о

близости к объёму основного датасета, то использовался второй подход. Суть его в том, что кроме части размеченных изображений из основного датасета, включаемых в дополнительные датасеты в соответствии с первым подходом, необходима аугментация таких изображений и аугментация вновь полученных и размеченных изображений. При реализации второго подхода сначала осуществлялись поиск и разметка новых изображений из сторонних источников с объектами трёх категорий по размерам и интересующих нас четырёх классов. Затем к размеченным таким изображениям и к выбранным в соответствии с первым подходом изображениям из основного датасета применялись различные известные методы аугментации [7, 49]. Для изображений при аугментации использовались процессы, реализованные в системе Roboflow [50]:

- Отражение по горизонтали;
- Повороты $\pm 20^\circ$;
- Сдвиги (срезы) по вертикали и горизонтали $\pm 20^\circ$.

Для автоматизации процессов формирования дополнительных датасетов в соответствии с предложенными подходами разработаны соответствующие алгоритмы и программа. Код её на языке Python 3.9 представлен в Приложении Е.

Созданные датасеты также разделялись на обучающую, валидационную и тестовую выборки. Пропорции разделения каждого датасета на эти выборки аналогичны пропорциям деления основного датасета, то есть 80%, 10% и 10% от количества изображений для каждого класса, входящих в датасет.

Полученный при реализации первого подхода дополнительный датасет с объектами малых размеров на изображениях состоит из 3 296 изображений. Размер этого датасета при применении второго подхода после аугментации был увеличен в более чем три раза и составил 10 674 изображения, содержащих 34 191 объектов, и столько же файлов аннотаций. На рисунках 22 и 23 представлены количественные характеристики изображений по классам объектов и по числу объектов на изображениях в этом датасете.

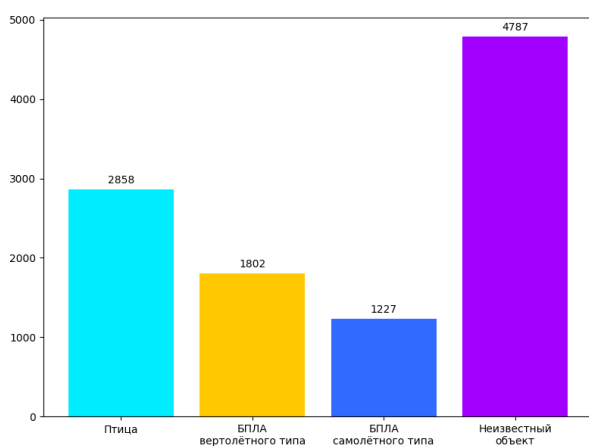


Рисунок 21 — Количество изображений в датасете по классам малоразмерных объектов

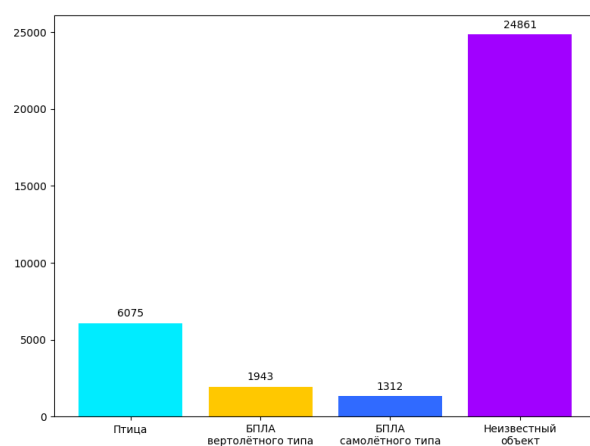


Рисунок 22 — Количество экземпляров малоразмерных объектов в датасете каждого класса

На рисунке 23 представлено распределение числа изображений в датасете малоразмерных объектов. По оси Y приведено количество изображений каждого класса в датасете с соответствующим числом объектов, а по оси X — число объектов на отдельном изображении.

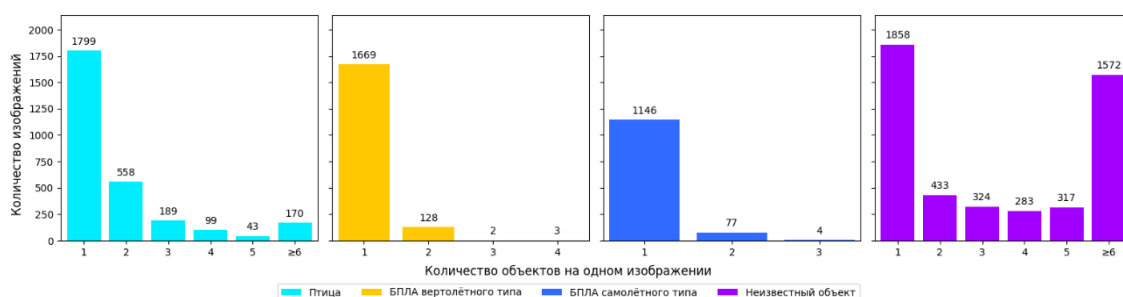


Рисунок 23 — Распределение числа изображений в датасете по классам в зависимости от количества малоразмерных объектов на отдельном изображении

Полученный в соответствии с первым подходом дополнительный датасет с изображениями объектов средних размеров состоит из 4 334 изображений. Размер его после аугментации был увеличен в более чем три раза и составил 14 740 изображений с 24 366 объектами и столько же файлов аннотаций. На рисунках 24 и 25 представлены количественные характеристики изображений по классам объектов и по числу объектов на изображениях в этом датасете.

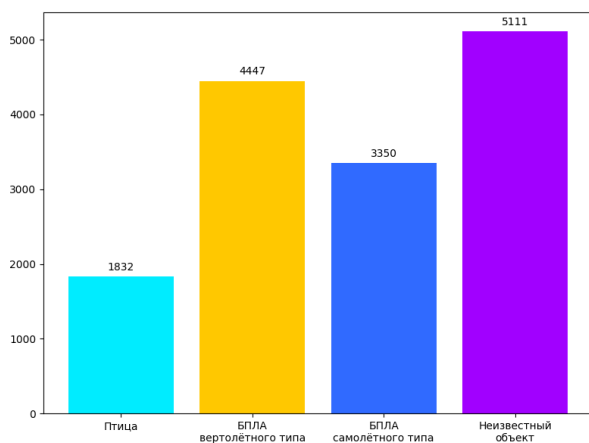


Рисунок 24 — Количество изображений в датасете по классам объектов средних размеров

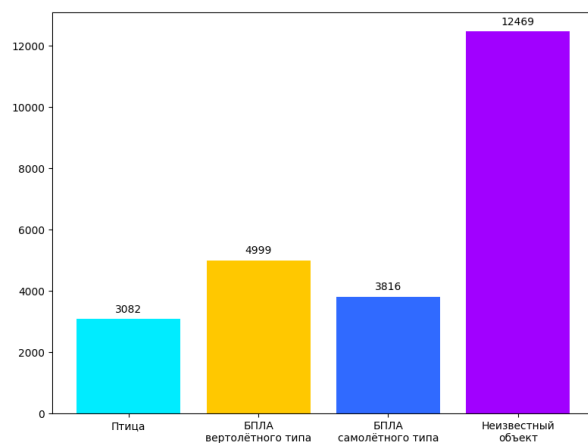


Рисунок 25 — Количество экземпляров объектов средних размеров в датасете каждого класса

На рисунке 26 представлено распределение числа изображений в датасете в случае объектов средних размеров на них. По оси Y приведено количество изображений каждого класса в датасете с соответствующим числом объектов, а по оси X — число объектов на отдельном изображении.

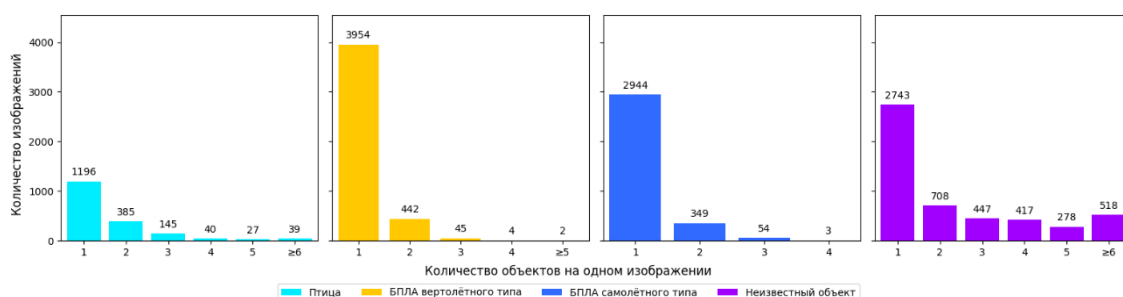


Рисунок 26 — Распределение числа изображений в датасете по классам в зависимости от количества объектов средних размеров на отдельном изображении

Полученный при использовании первого подхода дополнительный датасет с изображениями объектов больших размеров состоит из 4 688 изображений. Размер дополнительного датасета после аугментации увеличен в более чем три раза и составляет 14 926 изображений и столько же файлов аннотаций. На его изображениях содержатся 17 497 объектов. На рисунках 27 и 28 представлены количественные характеристики изображений по классам объектов и по числу объектов на изображениях в этом датасете.

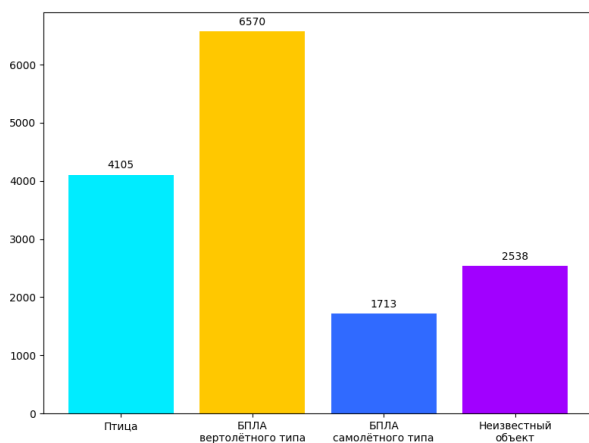


Рисунок 27 — Количество изображений в датасете по классам объектов больших размеров

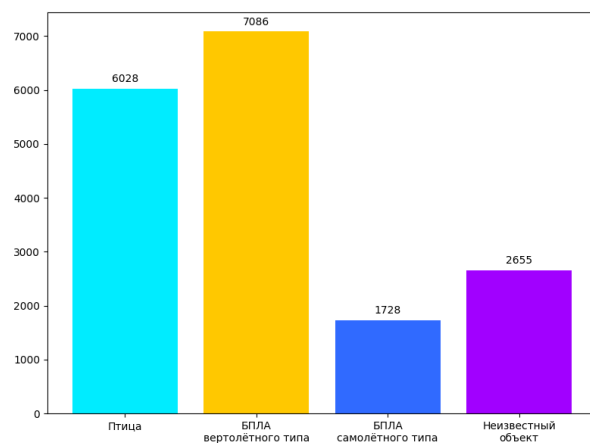


Рисунок 28 — Количество экземпляров объектов больших размеров в датасете каждого класса

На рисунке 29 представлено распределение числа изображений в датасете, в случае больших размеров объектов на них. По оси Y приведено количество изображений каждого класса в датасете с соответствующим числом объектов, а по оси X — число объектов на отдельном изображении.

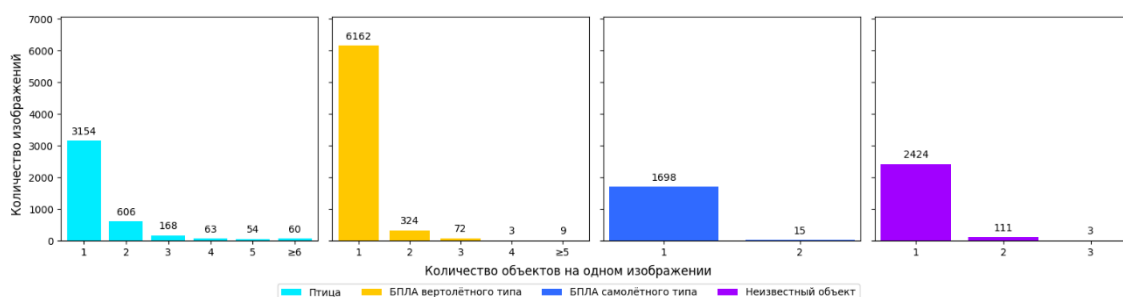


Рисунок 29 — Распределение числа изображений в датасете по классам в зависимости от количества объектов больших размеров на отдельном изображении

Все четыре разработанных датасета далее были использованы при обучении, валидации и исследованиях (тестировании) выбранных компактных моделей СНС и вновь разработанной модели СНС, входящих в класс YOLO.

ГЛАВА 3. РАЗРАБОТКА, ОБУЧЕНИЕ И ВАЛИДАЦИЯ КОМПАКТНЫХ МОДЕЛЕЙ СНС

3.1 Общие сведения

Ниже рассматриваются особенности разработки, обучения и валидации на обучающей и валидационной выборках (наборах данных) основного датасета трёх потенциально перспективных для создания мобильных СКЗ компактных моделей СНС класса YOLO: YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny.

Для того, чтобы в исследованиях этих моделей получить сопоставимые результаты и правильно выбрать по этим результатам наиболее эффективную модель СНС необходимо при обучении всех моделей использовать схожие параметры. Размер каждого входного изображения из основного датасета для всех исследуемых моделей составляет 416×416 пикселей.

В обучаемых моделях классы объектов представлены текстовыми идентификаторами:

- `bird` — Птица;
- `helicopter-type_uav` — БПЛА вертолётного типа;
- `aircraft-type_uav` — БПЛА самолётного типа;
- `unknown` — Неизвестный объект.

3.2 Разработка и обучение модели СНС YOLOv4 Tiny

Разработка модели СНС YOLOv4 Tiny осуществлялась с использованием фреймворка Darknet. Это фреймворк с открытым исходным кодом для создания и обучения моделей СНС, которые используются в таких областях, как обработка изображений, распознавание объектов различной физической природы, детектирование лиц и т.д. Darknet предоставляет простой и практичный API для создания и обучения моделей СНС.

В качестве основы для разработки и программной реализации модели СНС YOLOv4 Tiny использовалась архитектура модели, представленная в Приложении Ж. Проведены соответствующие тонкие настройки модели.

При обучении модели СНС на обучающей выборке из основного датасета применялся фреймворк Darknet и использовались следующие основные параметры:

- Скорость обучения (lr): 0,001;
- Импульс SGD ($momentum$): 0,949;
- Затухание веса оптимизатора ($weight_decay$): 0,0005;
- Количество итераций: 16 000;
- Размер выборки: 64;
- Функция активации: Mish;
- Функция оптимизации: SGD;
- Функция потерь: CIoU.

Итерация — некоторая абстракция, которую можно перевести в эпохи.

Эпоха — это процесс прохождения всего набора данных через нейронную сеть вперёд и назад. Этот параметр используется во фреймворке PyTorch.

Для перевода итераций в эпохи используется формула (6).

$$Epoch = \frac{SetSize}{IterationsPerEpoch}, \quad (6)$$

где $SetSize$ — размер тренировочного набора данных;

$IterationsPerEpoch$ — количество итераций, требуемых для одной эпохи, рассчитывается по формуле (7).

$$IterationsPerEpoch = \frac{Iterations}{BatchSize}, \quad (7)$$

где $Iterations$ — количество итераций для обучения модели СНС;

$BatchSize$ — размер пакета.

Согласно формуле (7) при использовании размера пакета, равном 64, на одну эпоху приходится 191 итерация, следовательно, согласно формуле (6), модель YOLOv4 Tiny для обучения использует 85 эпох.

Все параметры, используемые для обучения этой модели СНС, приведены в Приложении И.

Результаты валидации на валидационной выборке обученной модели YOLOv4 Tiny по точности детектирования для различных метрик представлены в таблице 4. Из неё следует, что для объектов классов «Птица» и «Неизвестный объект» точность детектирования весьма низка, а также точность детектирования для всех классов по метрике mAP_{0,5} невысока.

Таблица 4 — Результаты по точности детектирования объектов при валидации модели СНС YOLOv4 Tiny

Класс	Количество изображений	Average Precision (AP)	Precision	Recall	F1	mAP _{0,5}
Птица	1 526	0,423	0,560	0,520	0,540	0,610
БПЛА самолётного типа		0,699				
БПЛА вертолётного типа		0,747				
Неизвестный объект		0,570				
Все классы		0,609				

3.3 Разработка и обучение модели СНС YOLOv5 Small

Разработка, программная реализация и настройка модели СНС YOLOv5 Small проводилась с использованием фреймворка PyTorch. PyTorch — это фреймворк машинного обучения, который используется для создания нейронных сетей, в том числе свёрточных.

Одной из основных особенностей PyTorch является простота и удобство использования. Этот фреймворк позволяет создавать и использовать модели нейронных сетей, не требуя больших усилий от пользователя. Одним важным отличием между фреймворками PyTorch и Darknet является подход к обработке данных. PyTorch позволяет использовать динамические графы вычислений, что делает его удобным инструментом для обработки неструктурированных данных, таких как тексты и изображения.

В качестве основы для разработки и программной реализации модели YOLOv5 Small использовалась архитектура, представленная в Приложении К.

Обучение модели СНС проводилось на обучающей выборке из основного дата-сета с помощью фреймворка PyTorch, при этом использовались следующие основные параметры:

- Начальная скорость обучения (lr_0): 0,01;
- Конечная скорость обучения (lr_f): 0,01;
- Импульс SGD (*momentum*): 0,937;
- Затухание веса оптимизатора (*weight_decay*): 0,0005;
- Количество эпох: 100;
- Размер выборки: 32;
- Функция активации: SiLU;
- Функция оптимизации: SGD;
- Функция потерь: BCEWithLogitsLoss.

Все параметры, используемые для обучения этой модели СНС, приведены в Приложении Л.

На рисунке 30 представлена матрица ошибок при детектировании объектов каждого класса по результатам обучения и валидации модели YOLOv5 Small, генерируемая фреймворком PyTorch с помощью библиотеки Tensorboard.

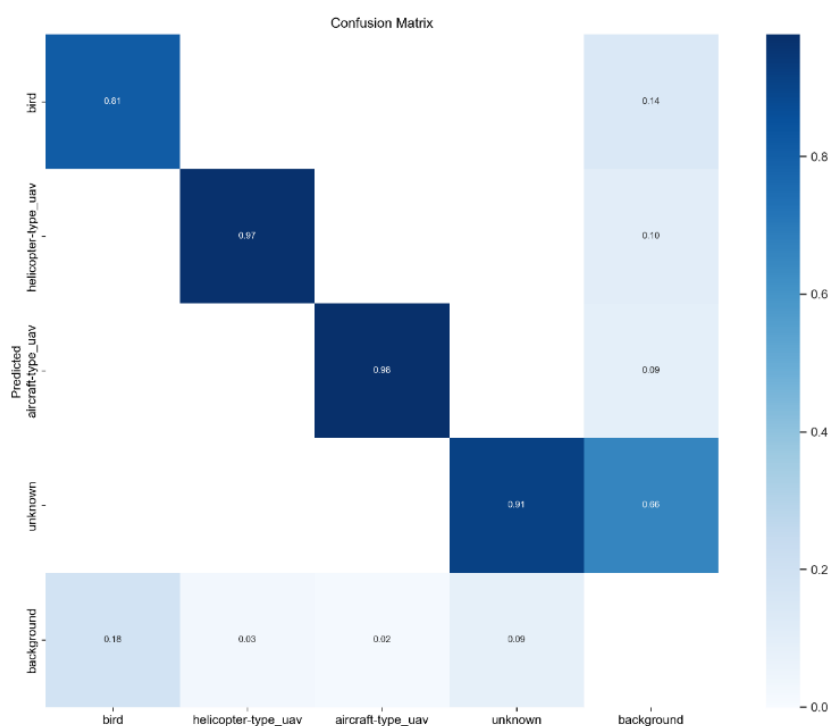


Рисунок 30 — Матрица ошибок для модели СНС YOLOv5 Small

Результаты точности детектирования при валидации обученной модели СНС YOLOv5 Small представлены в таблице 5. Отметим, что эти результаты по всем классам объектов и по всем метрикам весьма высоки и соответствуют требованию по точности детектирования из подраздела 1.3: значение метрики $mAP_{0,5}$ больше или равно 0,9 в случае объектов на изображениях разных масштабов.

Таблица 5 — Результаты точности детектирования при валидации модели СНС YOLOv5 Small

Класс	Количество изображений	Количество объектов	Precision	Recall	$AP_{0,5}$, $mAP_{0,5}$	$AP_{0,5:0,95}$, $mAP_{0,5:0,95}$
Птица	1 523	1 173	0,926	0,772	0,85	0,695
БПЛА самолётного типа		1 167	0,956	0,955	0,979	0,803
БПЛА вертолётного типа		893	0,944	0,958	0,968	0,804
Неизвестный объект		2 826	0,911	0,855	0,917	0,613
Все классы		6 059	0,934	0,855	0,928	0,729

3.4 Разработка и обучение модели СНС YOLOv7 Tiny

Разработка, программная реализация и настройка модели СНС YOLOv7 Tiny проводилась с использованием фреймворка PyTorch. В качестве основы для разработки и программной реализации модели YOLOv7 Tiny использовалась архитектура, представленная в Приложении М.

Для обучения модели СНС на обучающей выборке из основного датасета используются основные параметры, которые полностью совпадают с основными параметрами модели СНС YOLOv5 Small, описанными в подразделе 3.3. Такое стало возможным, так как архитектура каждой модели спроектирована с помощью фреймворка PyTorch. Все параметры, используемые для обучения СНС, приведены в Приложении Н.

На рисунке 31 представлена матрица ошибок по точности детектирования для обученной и валидированной модели YOLOv7 Tiny, генерируемая фреймворком PyTorch с помощью библиотеки Tensorboard.

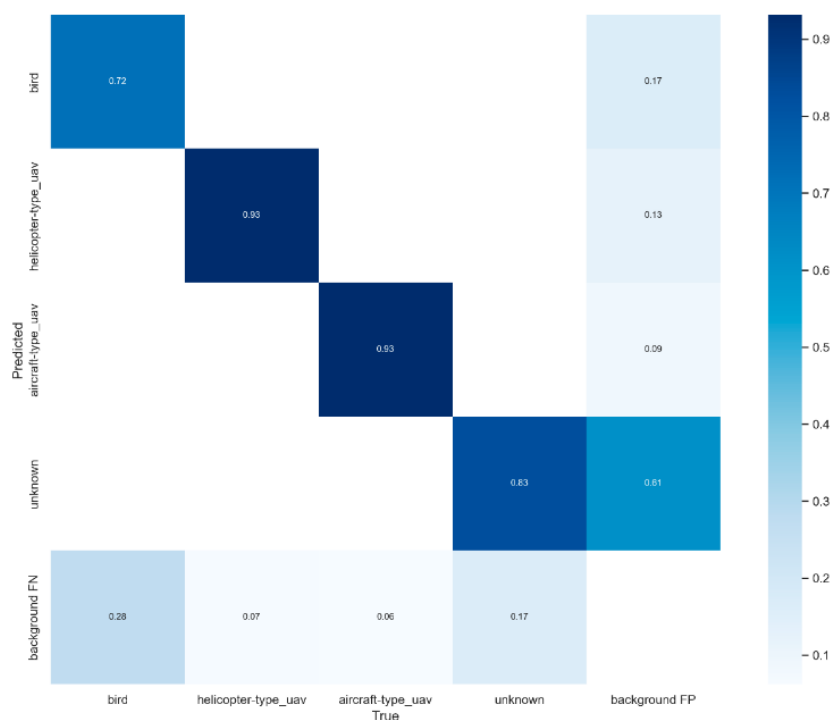


Рисунок 31 — Матрица ошибок для модели СНС YOLOv7 Tiny

Результаты по точности детектирования при валидации на соответствующей выборке модели СНС YOLOv7 Tiny представлены в таблице 6. Видим, что результаты для всех классов объектов весьма высоки, но уступают части результатов, полученных с помощью модели YOLOv5 Small.

Таблица 6 — Результаты точности детектирования при валидации модели СНС YOLOv7 Tiny

Класс	Количество изображений	Количество объектов	Precision	Recall	AP _{0,5} , mAP _{0,5}	AP _{0,5:0,95} , mAP _{0,5:0,95}
Птица	1 523	1 307	0,891	0,658	0,738	0,514
БПЛА самолётного типа		874	0,916	0,887	0,945	0,683
БПЛА вертолётного типа		1 208	0,905	0,909	0,934	0,682

Класс	Количество изображений	Количество объектов	Precision	Recall	AP _{0,5} , mAP _{0,5}	AP _{0,5:0,95} , mAP _{0,5:0,95}
Неизвестный объект		3 223	0,862	0,753	0,828	0,485
Все классы		6 612	0,882	0,806	0,857	0,590

3.5 Обучение моделей СНС YOLOv5 Small на изображениях с объектами трёх категорий размеров

Для последующего исследования точности детектирования и скорости вычисления модели СНС на изображениях с объектами трёх категорий размеров была взята модель YOLOv5 Small в связи с тем, что она имеет лучшие характеристики среди выбранных моделей СНС по результатам обучения и валидации на основном датасете. Поскольку были сформированы три дополнительных датасета, необходимо было обучить и валидировать три варианта этой модели (по сути, получить три обученные модели СНС YOLOv5 Small). Исходные параметры и архитектура для каждого варианта модели те же, что и при обучении модели YOLOv5 Small на основном датасете.

Результаты по точности детектирования объектов малых размеров на изображениях первого дополнительного датасета, полученные с помощью обученной и валидированной модели YOLOv5 Small, представлены в таблице 7.

Таблица 7 — Результаты точности детектирования объектов малых размеров на изображениях с помощью валидированной модели СНС YOLOv5 Small

Класс	Количество изображений	Количество объектов	Precision	Recall	AP _{0,5} , mAP _{0,5}	AP _{0,5:0,95} , mAP _{0,5:0,95}
Птица	1 055	737	0,850	0,446	0,524	0,232
БПЛА самолётного типа		182	0,885	0,780	0,835	0,438
БПЛА вертолётного типа		236	0,870	0,791	0,831	0,400

Класс	Количество изображений	Количество объектов	Precision	Recall	AP _{0,5} , mAP _{0,5}	AP _{0,5:0,95} , mAP _{0,5:0,95}
Неизвестный объект		2 756	0,842	0,743	0,808	0,374
Все классы		3 911	0,862	0,690	0,750	0,361

Результаты по точности детектирования изображений с помощью обученной и валидированной на втором дополнительном датасете модели СНС YOLOv5 Small для случая объектов средних размеров представлены в таблице 8.

Таблица 8 — Результаты точности детектирования изображений с помощью валидированной модели СНС YOLOv5 Small для случая объектов средних размеров

Класс	Количество изображений	Количество объектов	Precision	Recall	AP _{0,5} , mAP _{0,5}	AP _{0,5:0,95} , mAP _{0,5:0,95}
Птица	1 456	393	0,861	0,794	0,805	0,590
БПЛА самолётного типа		492	0,942	0,888	0,952	0,674
БПЛА вертолётного типа		593	0,945	0,921	0,958	0,681
Неизвестный объект		1 298	0,881	0,897	0,936	0,653
Все классы		2 776	0,907	0,875	0,924	0,649

Результаты по точности детектирования изображений с объектами больших размеров, полученные на третьем дополнительном датасете с помощью валидированной модели СНС YOLOv5 Small, показаны в таблице 9.

Таблица 9 — Результаты точности детектирования изображений с помощью валидированной модели СНС YOLOv5 Small для случая объектов больших размеров

Класс	Количество изображений	Количество объектов	Precision	Recall	$AP_{0,5}$, $mAP_{0,5}$	$AP_{0,5:0,95}$, $mAP_{0,5:0,95}$
Птица	1 483	631	0,913	0,927	0,955	0,785
БПЛА самолётного типа		188	0,947	0,989	0,983	0,793
БПЛА вертолётного типа		782	0,947	0,983	0,979	0,804
Неизвестный объект		323	0,958	0,985	0,980	0,827
Все классы		1 924	0,941	0,971	0,974	0,802

Из таблиц 7, 8 и 9 следует, что лучшую точность детектирования валидированные модели СНС YOLOv5 Small показывают на изображениях с большими объектами, а худшую — на изображениях с малыми по размерам объектами. При этом по метрикам $AP_{0,5}$ и $mAP_{0,5}$ для изображений с большими объектами порог 0,9 значительно превышен. Результаты по точности детектирования объектов средних размеров на изображениях занимают промежуточное положение.

Обученные и валидированные модели СНС YOLOv5 Small будут использованы в дальнейших исследованиях на тестовых изображениях с объектами трёх категорий размеров из соответствующих дополнительных датасетов.

ГЛАВА 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Предпроектный анализ

5.1.1 Потенциальные потребители результатов исследования

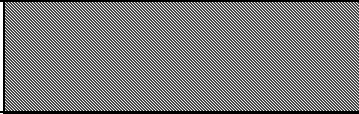
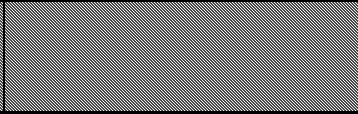
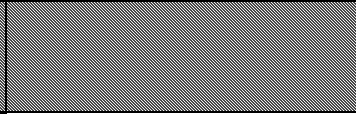
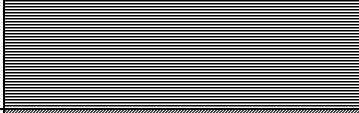
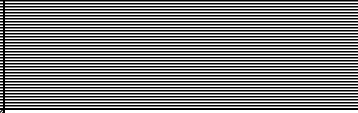
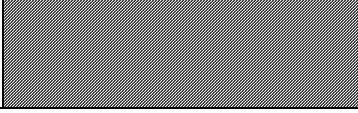
Целью данной работы является разработка и исследование моделей свёрточных нейронных сетей (СНС) для детектирования (распознавания, локализации и классификации) одного или нескольких объектов различных классов на изображении.

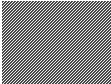
Разработка модели СНС заключается в создании новых моделей СНС на основе семейства моделей YOLO с использованием Inception – ResNet-модулей. Для обучения моделей СНС используется набор изображений, содержащих объекты четырёх классов: «Птица», «БПЛА самолётного типа», «БПЛА вертолётного типа» и «Неизвестный объект». Результатом работы модели СНС для каждого изображения является ограничивающий прямоугольник вокруг объекта и его класс.


Целевым рынком для данной разработки являются компании и научно-исследовательские лаборатории и центры, основной деятельностью которых является анализ изображений и разработка программного обеспечения для систем компьютерного зрения (СКЗ) с целью обнаружения и классификации объектов на них.

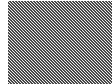
Сегментация рынка услуг проводится по степени развития и использования систем машинного обучения и компьютерного зрения. Результат сегментации представлен в таблице 18.

Таблица 18 — Карта сегментации степени развития и использования систем машинного обучения и компьютерного зрения

		Разработка и использование		
		Компания, разрабатывающая модели машинного обучения и модели СНС	Компания, разрабатывающая программное обеспечение с использованием машинного обучения и СНС	Компания, разрабатывающая СКЗ с использованием машинного обучения и СНС
Размер компании	Большой			
	Средний			
	Малый			

 ООО «Беспилотные системы»

 ООО «ЛЭМЗ–Т»

 ООО «НПО «Алмаз»

Таким образом рассмотрена возможная сегментация рынка по разработке СКЗ и использованию машинного обучения, в частности свёрточных нейронных сетей.

5.1.2 Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения

В последние годы значительное внимание уделяется решению задач распознаванию объектов и объектного детектирования. При этом исследовательский интерес большинства авторов ранних работ [5, 6], а также авторов многих современных работ [7, 8] в области распознавания / детектирования объектов на изображениях сосредоточен на достижении высокой точности распознавания / детектирования этих объектов с помощью моделей СНС, при этом эффективности алгоритмов вычисления СНС не уделяется должного внимания. Стремление к точности распознавания и обнаружения объектов приводит к разработке моделей СНС с высокой вычислительной сложностью, что в конечном итоге приводит к низкой алгоритмической эффективности этих моделей.

Примерами работ, направленных на создание более эффективных моделей СНС для детектирования объектов на изображениях, являются исследования якорных одноступенчатых моделей СНС [9, 10, 11, 12, 13, 14] и двухступенчатых моделей СНС [5, 29, 30, 6, 31, 32]. Согласно исследованиям, наиболее используемыми и актуальными архитектурами являются:

- YOLO (1);
- Mask R-CNN (2);
- Faster R-CNN (3).

Экспертная оценка основных технических характеристик этих решений представлена в таблице 19. В качестве набора технических критериев рассматриваются следующие требования:

1. Технические критерии оценки ресурсоэффективности
 - а) Точность — точность детектирования (классификации) объектов;
 - б) Требование к вычислительным ресурсам — насколько мощный компьютер необходим для разработки;
 - в) Возможность дообучения — можно ли использовать уже обученные модели для обучения новых;
 - г) Универсальность — насколько универсальна архитектура;
 - д) Востребованность — насколько востребована архитектура;
 - е) Требовательность к данным - насколько требовательна обученная модель к данным (изображениям).
2. Экономические критерии оценки эффективности
 - а) Стоимость разработки — насколько дорогой является разработка;
 - б) Ожидаемый жизненный цикл — как долго обученная модель может быть актуальной.

Таблица 19 — Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия, V_i	Баллы			Конкурентоспособность		
		B_1	B_2	B_3	K_1	KC_2	K_3
Технические критерии оценки ресурсоэффективности							
1. Точность	0,25	5	3	4	1,25	0,75	1,00
2. Требование к вычислительным ресурсам	0,10	4	5	5	0,40	0,50	0,50
3. Возможность дообучения	0,05	4	3	3	0,20	0,15	0,15
4. Универсальность	0,15	5	1	2	0,75	0,15	0,30
5. Востребованность	0,10	4	2	3	0,40	0,20	0,30
6. Требовательность к данным	0,15	5	2	3	0,75	0,3	0,45
Экономические критерии оценки эффективности							
1. Стоимость разработки	0,10	5	3	4	0,50	0,30	0,40
2. Ожидаемый жизненный цикл	0,10	5	4	2	0,50	0,40	0,20
Итого	1	37	23	26	4,75	2,75	3,30

На основе анализа можно сделать вывод, что конкурентные архитектурные решения для создания моделей СНС имеют меньшую точность и востребованность по сравнению с СНС класса YOLO. Кроме того, СНС класса YOLO обладают большей универсальностью, что позволяет решать различные задачи с помощью них.

На основе анализа можно сделать вывод, что СНС класса YOLO является наиболее мощным конкурентным решением. Этот вывод подтверждает решение использовать СНС класса YOLO в качестве основы для исследования.

5.1.3 SWOT-анализ

SWOT — Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы); представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Анализ делится на два этапа: первый - описание сильных и слабых сторон проекта, выявление возможно-

стей и угроз, которые возникли или могут возникнуть в проекте; второй — определение совместимости сильных и слабых сторон проекта с внешними условиями, и что необходимо сделать для стратегических изменений. Матрица SWOT-анализа представлена в таблице 20.

Таблица 20 — Матрица SWOT-анализа

	<p>Strengths:</p> <p>S1. Высокая точность детектирования.</p> <p>S2. Высокая скорость вычислений СНС.</p> <p>S3. Разработанные модели являются компактными.</p> <p>S4. Возможность дополнительного обучения моделей.</p>	<p>Weaknesses:</p> <p>W1. Высокие требования к вычислительным ресурсам.</p> <p>W2. Сложность эксплуатации для рядового пользователя.</p> <p>W3. Сложность сбора и аннотирования исходных данных.</p> <p>W4. Отсутствие финансирования.</p> <p>W5. Вероятность ошибочного детектирования.</p>
<p>Opportunities:</p> <p>O1. Количественное и качественное улучшение исходных данных.</p> <p>O2. Увеличение объёма данных путём аугментации.</p> <p>O3. Возможность точной настройки гиперпараметров во время разработки модели.</p> <p>O4. Возможность интеграции в реальные решения.</p>	<p>O1O2O4S1S2S3. Создание компактных, точных и быстрых моделей для использования в мобильных СКЗ.</p> <p>O2S4. Обучение новых моделей на основе существующих.</p> <p>O3S4. Модификация существующих моделей для повышения точности.</p> <p>O1O2O3S4. Создание более точных моделей.</p>	<p>O1O2W1. Увеличение требуемой мощности компьютера.</p> <p>O3W2. Повышение сложности разработки.</p> <p>O1W3. Увеличение времени подготовки разработки.</p> <p>O4W2. Повышение сложности интеграции в реальные системы.</p> <p>O4W4. Повышение сложности или невозможность разработки качественных моделей для интеграции.</p> <p>O4W5. Нежелательные ложные срабатывания в реальных решениях.</p>
<p>Threats:</p> <p>T1. Низкая скорость инте-</p>	<p>T1S1S2S3. Компактные, точные и быстрые модели быть не</p>	<p>T1W1W2. Низкая скорость интеграции может быть связана с</p>

	<p>Strengths:</p> <p>S1. Высокая точность детектирования.</p> <p>S2. Высокая скорость вычислений СНС.</p> <p>S3. Разработанные модели являются компактными.</p> <p>S4. Возможность дополнительного обучения моделей.</p>	<p>Weaknesses:</p> <p>W1. Высокие требования к вычислительным ресурсам.</p> <p>W2. Сложность эксплуатации для рядового пользователя.</p> <p>W3. Сложность сбора и аннотирования исходных данных.</p> <p>W4. Отсутствие финансирования.</p> <p>W5. Вероятность ошибочного детектирования.</p>
<p>грации в реальные решения.</p> <p>T2. Высокая стоимость получения новых релевантных исходных данных.</p> <p>T3. Появление лучших аналогов.</p>	<p>интегрированы в реальные системы.</p> <p>T1S4. Риск того, что при создании более совершенной модели интеграция предыдущей будет только завершена.</p> <p>T2S4. Невозможность создания более совершенных моделей из-за высокой стоимости сбора новых исходных данных.</p> <p>T3S1S2S3. Аналоги могут быть лучше, чем наши компактные, точные и быстрые модели.</p> <p>T3S4. Аналоги могут быть лучше, чем создание новых моделей.</p>	<p>необходимостью модификации модели под другую архитектуру.</p> <p>T1W2. Низкая скорость интеграции может быть связана с необходимостью реализации модели в виде ПЛИС.</p> <p>T2W3W4. Невозможность сбора и аннотирования дополнительных релевантных исходных данных.</p> <p>T3W4. Отказ от разработки моделей.</p> <p>T3W5. Отказ от использования моделей.</p>

Основываясь на результатах SWOT-анализа, можно заключить, что проблемы, которые могут возникнуть перед данным исследовательским проектом, могут быть решены при помощи его сильных сторон, в том числе улучшение технологий на основании данных, полученных при наблюдении за конкурентами.

5.1.4 Оценка готовности проекта к коммерциализации

Для исследовательского проекта чрезвычайно важно оценить его готовность к коммерциализации, а также уровень знаний, необходимых для его реализации. По результатам такого анализа можно сделать вывод о готовности исследовательского проекта к коммерциализации, а также о необходимости увеличения количества членов команды проекта. Перечень вопросов и оценка показателей степени проработки проекта с точки зрения коммерциализации и компетенций разработчика представлены в таблице 21.

Таблица 21 — Оценка степени готовности научного проекта к коммерциализации

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1	Определен имеющийся научно-технический задел	4	4
2	Определены перспективные направления коммерциализации научно-технического задела	3	5
3	Определены отрасли и технологии (товары, услуги) для предложения на рынке	3	3
4	Определена товарная форма научно-технического задела для представления на рынок	3	2
5	Определены авторы и осуществлена охрана их прав	3	4
6	Проведена оценка стоимости интеллектуальной собственности	4	4
7	Проведены маркетинговые исследования рынков сбыта	3	3
8	Разработан бизнес-план коммерциализации научной разработки	3	3
9	Определены пути продвижения	3	4

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
	научной разработки на рынок		
10	Разработана стратегия (форма) реализации научной разработки	4	5
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	3	3
12	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	4	4
13	Проработаны вопросы финансирования коммерциализации научной разработки	4	4
14	Имеется команда для коммерциализации научной разработки	3	3
15	Проработан механизм реализации научного проекта	5	5
	Итого	52	56

Значения оценки степени проработанности и знаний говорит о том, что перспективы развития и знания разработчика выше среднего. Для увеличения показателей необходимо проконсультироваться с экспертами в сферах, затрагиваемых разработкой, сформировать команду и найти финансирование.

5.1.5 Методы коммерциализации результатов научно-технического исследования

При коммерциализации научно-технических разработок продавец (а это, как правило, владелец соответствующих объектов интеллектуальной собственности), преследует вполне определённую цель, которая во многом зависит от того, куда в последующем он намерен направить (использовать, вложить) полученный коммерческий эффект. Это может быть получение средств для продолжения своих научных исследований и разработок (получение финанси-

ния, оборудования, уникальных материалов, других научно-технических разработок и т.д.), одноразовое получение финансовых ресурсов для каких-либо целей или для накопления, обеспечение постоянного притока финансовых средств, а также их различные сочетания.

В то же время от правильного выбора метода коммерциализации во многом зависит срок продвижения товара на рынок. Целью данного раздела является выбор метода коммерциализации объекта исследования и обоснование его целесообразности.

В данной работе возможны следующие методы коммерциализации научных разработок:

1. Торговля патентными лицензиями, т.е. передача третьим лицам права использования объектов интеллектуальной собственности на лицензионной основе. При этом в патентном законодательстве выделяющие виды лицензий: исключительные (простые), исключительные, полные лицензии, сублицензии, опционы.

2. Инжиниринг как самостоятельный вид коммерческих операций предполагает предоставление на основе договора инжиниринга одной стороной, именуемой консультантом, другой стороне, именуемой заказчиком, комплекса или отдельных видов инженерно-технических услуг, связанных с проектированием, строительством и вводом объекта в эксплуатацию, с разработкой новых технологических процессов на предприятии заказчика, усовершенствованием имеющихся производственных процессов вплоть до внедрения изделия в производство и даже сбыта продукции.

3. Передача интеллектуальной собственности в уставной капитал предприятия.

В контексте данной работы инжиниринг является наиболее подходящим способом коммерциализации, поскольку использование ресурсов заказчика упрощает и ускоряет разработку технологического решения.

5.2 Инициация проекта

Группа процессов инициации состоит из процессов, которые выполняются для определения нового проекта или новой фазы существующего. В рамках процессов инициации определяются изначальные цели и содержание и фиксируются изначальные финансовые ресурсы. Определяются внутренние и внешние заинтересованные стороны проекта, которые будут взаимодействовать и влиять на общий результат научного проекта. Данная информация закрепляется в Уставе проекта.

Устав проекта документирует бизнес-потребности, текущее понимание потребностей заказчика проекта, а также новый продукт, услугу или результат, который планируется создать.

5.2.1 Цели и результат проекта

В этом разделе представлена информация о заинтересованных сторонах проекта, иерархии целей проекта и критериях достижения целей.

Заинтересованные стороны проекта — это лица или организации, которые активно вовлечены в проект или чьи интересы могут быть затронуты положительно или отрицательно в результате выполнения или завершения проекта. Информация о заинтересованных сторонах проекта представлена в таблице 22.

Таблица 22 — Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
Компания-пользователь, научно-исследовательская лаборатория	Получение результатов обнаружения объектов на изображениях, разработка новых моделей СНС, разработка программного обеспечения с использованием моделей СНС.
Разработчик	Получение опыта, получение заработной платы.
Компания разработчика	Получение прибыли и инвестиций, разработка нового программного обеспечения, получение новых компетенций.

Научный руководитель, магистрант	Завершение выпускной квалификационной работы.
----------------------------------	---

Цель и результаты проекта представлены в таблице 23.

Таблица 23 — Цели и результат проекта

Цели проекта:	<ul style="list-style-type: none"> ▪ Изучить предметную область свёрточных нейронных сетей (СНС); ▪ Изучить подходы и технологии для сбора, аннотирования и аугментации данных; ▪ Исследовать компактные модели СНС с точки зрения точности детектирования (классификации) и скорости вычисления; ▪ Реализовать и исследовать новую архитектуру модели СНС, созданную на основе лучшей компактной модели по результатам исследования; ▪ Представить результаты исследования, выводы и предложения.
Ожидаемые результаты проекта:	<ul style="list-style-type: none"> ▪ Были созданы и обучены модели СНС для детектирования летающих объектов на изображениях; ▪ Выпускная квалификационная работа завершена.
Критерии приёмки результата проекта:	<ul style="list-style-type: none"> ▪ Успешное тестирование на тестовых выборках данных; ▪ Удовлетворительные значения метрик точности классификации; ▪ Удовлетворительная скорость вычислений модели СНС.
Требования к результату проекта:	<p>Точность детектирования летающих объектов на изображениях с помощью модели СНС должна быть высока, её значение по метрике $mAP_{0,5}$ больше или равно 0,9 в случае объектов на изображениях разного размера и больше или равно 0,7 в случае только малых объектов на изображениях.</p> <p>Скорость вычисления СНС (количество проанализированных изображений в секунду), измеряемая показателем FPS, должна быть не менее 25.</p> <p>Алгоритмическая эффективность алгоритма, вычисляемая в</p>

	количестве операций над 16-битными числами с плавающей запятой, и вес модели в МБ должны быть, по возможности, минимальными.
--	--

5.2.2 Организационная структура проекта

Состав рабочей группы данного проекта, роль каждого участника в данном проекте, а также функции, выполняемые каждым из участников и их трудозатраты в проекте представлены в таблице 24.

Таблица 24 — Рабочая группа проекта

№ п/п	ФИО, основное место работы, должность	Роль в проекте	Функции	Трудозатраты, час.
1	Марков Николай Григорьевич, д.т.н., Профессор Отделения информационных технологий, Инженерная школа информационных технологий и робототехники	Руководитель проекта	Постановка целей и задач; Разработка и мониторинг календарного плана; Проведение консультаций по целям и задачам выпускной квалификационной работы; Оценка эффективности полученных результатов.	160
2	Григорьев Евгений Викторович, Начальник сектора 147 ООО «ЛЭМЗ-Т»	Заказчик проекта	Постановка желаемых целей; Оценка результатов технологической практики; Консультации по необходимым изменениям в выпускной квалификационной работе.	48
3	Спицына Любовь Юрьевна, к.э.н., Доцент Отделения социально-гуманитарных наук, Школа базовой инженерной подготовки	Эксперт проекта	Консультирование по разделу выпускной квалификационной работы «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение».	48
4	Антоневич Ольга	Эксперт проекта	Консультирование по разде-	40

№ п/п	ФИО, основное место работы, должность	Роль в проекте	Функции	Трудозатраты, час.
	Алексеевна, к.б.н., Доцент Отделения общетехнических дисциплин, Школа базовой инженерной подготовки	екта	лу выпускной квалификационной работы «Социальная ответственность».	
5	Ануфриева Татьяна Николаевна, Старший преподаватель Отделения иностранных языков, Школа базовой инженерной подготовки	Эксперт проекта	Консультации по разделу на английском языке выпускной квалификационной работы.	16
6	Мунько Александр Сергеевич, Магистрант	Исполнитель по проекту	Изучение предметной области; Сбор и аннотирование данных (изображений); Разработка и исследование свёрточных нейронных сетей; Составление пояснительной записки.	880
Итого				1 192

5.2.3 Ограничения и допущения проекта

Ограничения проекта — это все факторы, которые могут ограничивать степень свободы членов команды проекта. Эти факторы перечислены в таблице 25.

Таблица 25 — Ограничения проекта

Фактор	Ограничения / Допущения
Источник финансирования	Средства ТПУ
Сроки реализации проекта	30.01.2023 — 27.06.2023

Фактор	Ограничения / Допущения
Дата утверждения плана проекта	03.03.2023
Дата завершения	27.06.2023
Продолжительность работы над проектом в день	8 часов

5.3 Планирование управления научно-техническим проектом

Группа процессов планирования состоит из процессов, осуществляемых для определения общего содержания работ, уточнения целей и разработки последовательности действий, требуемых для достижения данных целей.

5.3.1 Иерархическая структура работ проекта

Иерархическая структура работ (ИСР) — детализация укрупнённой структуры работ. В процессе создания ИСР структурируется и определяется содержание всего проекта. В процессе создания ИСР структурируется и определяется содержание всего проекта. Диаграмма ИСР показана на рисунке 40.

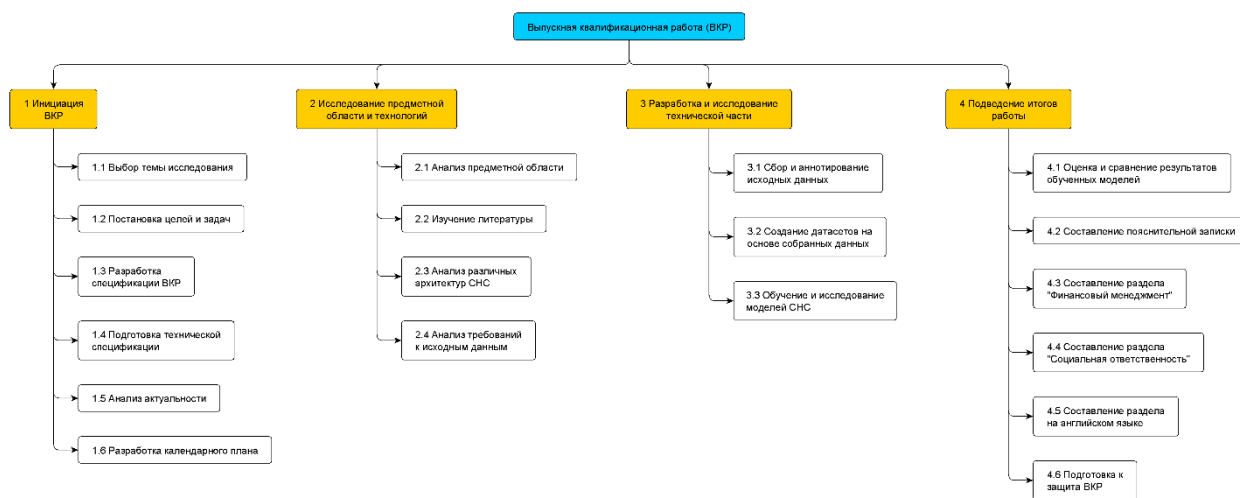


Рисунок 40 — Иерархическая структура работ

5.3.2 План проекта

В процессе организации работ по внедрению необходимо обоснованно планировать распределение труда и рабочего времени каждого участника процесса. Для этого был составлен полный перечень работ, представленный в таблице 26. Рабочие дни были рассчитаны исходя из шестидневной рабочей недели, с учётом праздничных дней Российской Федерации. Список участников представлен ниже:

- НР — Научный руководитель (Руководитель проекта), Марков Николай Григорьевич;
- З — Заказчик проекта, Григорьев Евгений Викторович;
- ФМ — Эксперт по разделу Финансового Менеджмента, Спицына Любовь Юрьевна;
- СО — Эксперт по разделу Социальной Ответственности, Антоневиц Ольга Алексеевна;
- АЧ — Эксперт по разделу на английском языке, Ануфриева Татьяна Николаевна;
- С — Исполнитель по проекту (Магистрант), Мунько Александр Сергеевич.

Таблица 26 — Календарный план проекта


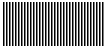
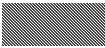
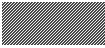


№	Название	Длительность, раб. дни	Дата начала работ	Дата окончания работ	Состав участников
1	Инициация ВКР				
1.1	Выбор темы исследования	5	30.01.2023	03.02.2023	НР, С
1.2	Постановка целей и задач	3	04.02.2023	07.02.2023	НР, З
1.3	Разработка спецификации ВКР	6	08.02.2023	14.02.2023	НР, С
1.4	Подготовка технической спецификации	3	15.02.2023	17.02.2023	С
1.5	Анализ актуальности	7	18.02.2023	28.02.2023	С
1.6	Разработка календарного плана	3	01.03.2023	03.03.2023	НР, С
2	Исследование предметной области и технологий				
2.1	Анализ предметной области	9	04.03.2023	15.03.2023	С
2.2	Изучение литературы	9	16.03.2023	24.03.2023	С
2.3	Анализ различных архитектур СНС	6	25.03.2023	31.03.2023	С

№	Название	Длительность, раб. дни	Дата нача- ла работ	Дата окон- чания ра- бот	Состав участников
2.4	Анализ требований к исходным данным	4	01.04.2023	05.04.2023	С
3	Разработка и исследование технической части				
3.1	Сбор и аннотирование исходных данных	15	06.04.2023	22.04.2023	С
3.2	Создание датасетов на основе собранных данных	5	23.04.2023	28.04.2023	С
3.3	Обучение и исследование моделей СНС	10	29.04.2023	13.05.2023	С
4	Подведение итогов работы				
4.1	Оценка и сравнение результатов обученных моделей	3	14.05.2023	17.05.2023	НР, З, С
4.2	Составление пояснительной записки	16	18.05.2023	05.06.2023	С
4.3	Составление раздела "Финансовый менеджмент"	6	06.06.2023	13.06.2023	С, ФМ
4.4	Составление раздела "Социальная ответственность"	5	14.06.2023	19.06.2023	С, СО
4.5	Составление раздела на английском языке	2	20.06.2023	21.06.2023	С, АЧ
4.6	Подготовка к защите ВКР	5	22.06.2023	27.06.2023	НР, С

Диаграмма Ганта — это тип столбчатой диаграммы, иллюстрирующей график проекта. На этой диаграмме по вертикальной оси перечислены задачи, которые необходимо выполнить, а по горизонтальной оси — временные интервалы. Ширина горизонтальных полос на графике показывает продолжительность каждого вида деятельности. Диаграмма Ганта показана в таблице 27.

Таблица 27 — Диаграмма Ганта

№	Исполнители	Т _к , раб. дни	Продолжительность выполнения работ																	
			Январь			Февраль			Март			Апрель			Май			Июнь		
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1.1	НР	2			■															
	С	3				■														
1.2	НР	3				■														
	З					▨														
1.3	НР	4				■														
	С	2					■													
1.4	С	3					■													
1.5	С	7					■	■												
1.6	НР	3								■										
	С									■										
2.1	С	9							■	■										
2.2	С	9								■	■	■								
2.3	С	6									■	■								
2.4	С	4										■	■							
3.1	С	15										■	■	■						
3.2	С	5											■	■						
3.3	С	10											■	■	■					
4.1	НР	3																		
	З																			
	С																			
4.2	С	16																		
4.3	С	6																		
	ФМ																			
4.4	С	5																		
	СО																			
4.5	С	2																		
	АЧ																			
4.6	НР	5																		
	С																			

	НР, Научный руководитель
	З, Заказчик проекта
	ФМ, Эксперт по разделу Финансового Менеджмента
	СО, Эксперт по разделу Социальной Ответственности
	АЧ, Эксперт по разделу на английском языке
	С, Исполнитель по проекту (Магистрант)

5.4 Бюджет научного исследования

5.4.1 Сырье, материалы, покупные изделия и полуфабрикаты

Расчёт стоимости материальных затрат производится на основе действующих прайс-листов или договорных цен. В стоимость материальных затрат входят транспортно-заготовительные расходы (3-5% от цены). В эту же статью входит стоимость документации (канцелярские принадлежности, тиражирование материалов). Результаты данной статьи представлены в таблице 28. Таким образом, мы получаем, что $C_m = 2831,85$ руб.

Таблица 28 — Сырье, материалы, комплектующие изделия и покупные полуфабрикаты

Наименование	Марка, размер	Количество	Цена за единицу, руб.	Сумма руб.
Бумага для печати	A4	1 уп (500 листов)	399	399
Бумага для печати	A3	1 уп (500 листов)	599	599
Картридж для принтера	НР 33А	1 шт	1 699	1 699
Всего за материалы				2 697
Транспортно-заготовительные расходы (3-5%)				134,85
Итого по статье, C_m				2 831,85

5.4.2 Затраты на электроэнергию

Для расчёта стоимости электроэнергии используется следующая формула:

$$C = M_э \times M_0 \times I_0, \quad (8)$$

где $M_э$ — тариф на электроэнергию (3,16 руб. за 1 кВт/ч [55]);

M_0 — мощность оборудования, кВт;

I_0 — время использования оборудования, часы;

Согласно формуле 1, стоимость электроэнергии за 880 часов работы при использовании оборудования мощностью 0,25 кВт составляет 695,2 руб.

5.4.3 Специальное оборудование для научных (экспериментальных) работ

В эту статью включены все затраты, связанные с приобретением специального оборудования, необходимого для выполнения работ. Стоимость специального оборудования определяется в соответствии с действующими прейскурантами, затраты по которым приведены в таблице 29.

Таблица 29 — Расчёт стоимости специального оборудования

№ п/п	Наименование оборудования	Кол-во единиц оборудования	Цена единицы оборудования, руб.	Общая стоимость оборудования, руб.
1	Персональный компьютер	1 шт	64 400	64 400
2	Студенческая лицензия для JetBrains PyCharm Professional	1 шт	0	0
Итого				64 400

5.4.4 Основная заработная плата

В данном разделе рассматриваются оклад, стимулирующие и надбавки. Расчёт основан на сложности каждого этапа, а также на месячной зарплате исполнителя. Расчёт основной заработной платы представлен в таблице 30.

Таблица 30 — Расчёт основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоёмкость, чел.-дн.	Заработная плата, приходящаяся на один чел.-дн., руб.	Всего заработная плата по тарифу, руб.
1	Инициация ВКР	НР	12	6 469,08	77 628,96
2	Инициация ВКР	З	3	3 780,80	11 342,40
3	Инициация ВКР	С	18	2 995,17	53 913,06
4	Исследование пред-	С	28	2 995,17	83 864,76

№ п/п	Наименование этапов	Исполнители по категориям	Трудоёмкость, чел.-дн.	Зарботная плата, приходящаяся на один чел.-дн., руб.	Всего зарботная плата по тарифу, руб.
	метной области и технологий				
5	Разработка и исследование технической части	С	30	2 995,17	89 855,10
6	Подведение итогов работы	НР	8	6 469,08	51 752,64
7	Подведение итогов работы	З	3	3 780,80	11 342,40
8	Подведение итогов работы	ФМ	6	4 627,79	27 766,74
9	Подведение итогов работы	СО	5	4 627,79	23 138,95
10	Подведение итогов работы	АЧ	2	4 627,79	9 255,58
11	Подведение итогов работы	С	37	2 995,17	110, 821,29
Итого					550 681,88

Статья включает основную заработную плату работников, непосредственно занятых выполнением проекта, (включая премии, доплаты) и дополнительную заработную плату.

$$C_{зп} = Z_{осн} + Z_{доп}, \quad (9)$$

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

$Z_{доп}$ — дополнительная заработная плата.

Основная заработная плата рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \times T_{раб}, \quad (10)$$

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

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

$T_{\text{раб}}$ — продолжительность работ, выполняемых научно-техническим работником, раб. дн.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \times M}{F_{\text{д}}}, \quad (11)$$

где $Z_{\text{м}}$ — месячный должностной оклад работника, руб.;

M — количество месяцев работы без отпуска в течение года (при отпуске в 48 раб. дней $M = 10,4$ месяца, 6-дневная неделя);

$F_{\text{д}}$ — действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (252 дня).

В таблице 31 показано количество календарных дней, нерабочих и праздничных дней, дней в связи с потерей рабочего времени и фактический годовой фонд рабочего времени.

Таблица 31 — Баланс рабочего времени

Показатели рабочего времени	Работник
Календарное число дней	365
Количество нерабочих дней	
▪ выходные дни	54
▪ праздничные дни	11
Потери рабочего времени	
▪ отпуск	48
▪ невыходы по болезни	0
Действительный годовой фонд рабочего времени	252

Месячный должностной оклад работника рассчитывается по формуле:

$$Z_{\text{м}} = Z_{\text{б}} \times (k_{\text{пр}} + k_{\text{д}}) \times k_{\text{р}}, \quad (12)$$

где $Z_{\text{б}}$ — базовый оклад, руб.;

$k_{\text{пр}}$ — премиальный коэффициент;

$k_{\text{д}}$ — коэффициент доплат и надбавок;

$k_{\text{р}}$ — районный коэффициент (для Томска равен 1,3).

Таким образом, расчёты месячной заработной платы представлены в таблице 32.

Таблица 32 — Расчёт основной заработной платы

Исполнители	З _б , руб.	k _{пр}	k _д	k _р	З _м , руб.	З _{дн} , руб.	T _{раб} , дн.	З _{осн} , руб.
Научный руководитель	52 700	1,1	1,188	1,3	156 750,88	6 469,08	20	129 381,6
Заказчик проекта	30 800				91 611,52	3 780,80	6	22 684,80
Эксперт по разделу Финансового Менеджмента	37 700				112 134,88	4 627,79	6	27 766,74
Эксперт по разделу Социальной Ответственности	37 700				112 134,88	4 627,79	5	23 138,95
Эксперт по английской части	37 700				112 134,88	4 627,79	2	9 255,58
Исполнитель по проекту (Магистрант)	24 400				72 575,36	2 995,17	113	338 454,21

5.4.5 Дополнительная заработная плата

В данную статью включается сумма выплат, предусмотренных законодательством о труде, например, оплата очередных и дополнительных отпусков; оплата времени, связанного с выполнением государственных и общественных обязанностей; выплата вознаграждения за выслугу лет и т.п.

Дополнительная заработная плата рассчитывается на основе 10-15% от основного оклада работников по следующей формуле:

$$Z_{\text{доп}} = Z_{\text{осн}} \times k_{\text{доп}}, \quad (13)$$

где $Z_{\text{доп}}$ — дополнительная заработная плата, руб.;

$Z_{\text{осн}}$ — основная заработная плата, руб.;

$k_{\text{доп}}$ — коэффициент дополнительной зарплаты (10%).

В таблице 33 представлен расчёт дополнительной заработной платы.

Таблица 33 — Расчёт дополнительной заработной платы

Исполнители	Основная зарплата, руб.	Дополнительная зарплата, руб.	Зар. плата исполнителя, руб.
Научный руководитель	129 381,6	12 938,16	142 319,76
Заказчик проекта	22 684,80	2 268,48	24 953,28
Эксперт по разделу Финансового Менеджмента	27 766,74	2 776,67	30 543,41
Эксперт по разделу Социальной Ответственности	23 138,95	2 313,90	25 452,85
Эксперт по английской части	9 255,58	925,60	10 181,14
Исполнитель по проекту (Магистрант)	338 454,21	33 845,42	372 299,63

Таким образом, общая сумма дополнительной заработной платы равна 55 068,18 руб.

5.4.6 Отчисления на социальные нужды

Отчисления на социальное страхование (так называемый трудовой налог) во внебюджетные фонды являются обязательными по нормам, установленным законодательством Российской Федерации на государственное социальное страхование, пенсионный фонд и медицинское страхование от расходов работников.

Платёж во внебюджетные фонды определяется по формуле:

$$C_{\text{внеб}} = k_{\text{внеб}} \times (Z_{\text{осн}} + Z_{\text{доп}}), \quad (14)$$

где $k_{\text{внеб}}$ — коэффициент отчислений на уплату во внебюджетные фонды.

В таблице 34 приведён расчёт отчислений на социальные нужды с коэффициентом отчислений, равным 30%.

Таблица 34 — Расчёт отчислений на социальные нужды

Исполнители	Основная зарплата, руб.	Дополнительная зарплата, руб.	Отчисления, руб.
Научный руководитель	129 381,6	12 938,16	42 695,93
Заказчик проекта	22 684,80	2 268,48	7 485,98

Эксперт по разделу Финансового Менеджмента	27 766,74	2 776,67	9 163,02
Эксперт по разделу Социальной Ответственности	23 138,95	2 313,90	7 635,86
Эксперт по английской части	9 255,58	925,60	30 544,67
Исполнитель по проекту (Магистрант)	338 454,21	33 845,42	111 689,89

Таким образом, общая сумма отчислений на социальные нужды равна 209 215,35 руб.

5.4.7 Накладные расходы

В эту статью включаются затраты на управление и хозяйственное обслуживание, которые могут быть отнесены непосредственно на конкретную тему. Кроме того, сюда относятся расходы по содержанию, эксплуатации и ремонту оборудования, производственного инструмента и инвентаря, зданий, сооружений и др. В расчётах эти расходы принимаются в размере 70-90% от суммы основной заработной платы научно-производственного персонала данной научно-технической организации.

Накладные расходы составляют 16% от суммы основной и дополнительной заработной платы, работников, непосредственно участвующих в выполнении темы.

Расчёт накладных расходов ведётся по следующей формуле:

$$C_{\text{накл}} = k_{\text{накл}} \times (Z_{\text{осн}} + Z_{\text{доп}}), \quad (15)$$

где $k_{\text{накл}}$ — коэффициент накладных расходов.

В таблице 35 представлен расчёт накладных расходов.

Таблица 35 — Расчёт накладных расходов

Исполнители	Основная зарплата, руб.	Дополнительная зарплата, руб.	Накладные расходы, руб.
Научный руководитель	129 381,6	12 938,16	22 771,16
Заказчик проекта	22 684,80	2 268,48	3 992,52
Эксперт по разделу Финан-	27 766,74	2 776,67	4 886,95

Исполнители	Основная зарплата, руб.	Дополнительная зарплата, руб.	Накладные расходы, руб.
сового Менеджмента			
Эксперт по разделу Социальной Ответственности	23 138,95	2 313,90	4 072,46
Эксперт по английской части	9 255,58	925,60	1 628,98
Исполнитель по проекту (Магистрант)	338 454,21	33 845,42	59 567,94

Таким образом, общая сумма накладных расходов равна 96 920,01 руб.

5.4.8 Формирование бюджетных расходов

Результаты формирования бюджетных расходов представлены в таблице 36.

Таблица 36 — Бюджет затрат на проект

Статьи	Стоимость, руб.
Расходы на сырье и материалы	2 831,85
Расходы на электроэнергию	695,20
Расходы на специальное оборудование для научных экспериментов	64 400,00
Основная заработная плата	550 681,88
Дополнительная зарплата	55 068,18
Социальные отчисления	209 215,35
Накладные расходы	96 920,01
Итого запланированные расходы	979 812,47

5.5 Оценка сравнительной эффективности исследования

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

Интегральный финансовый показатель разработки определяется как:

$$I_{\Phi}^p = \frac{\Phi_{pi}}{\Phi_{max}}, \quad (16)$$

где I_{Φ}^p — интегральный финансовый показатель разработки;
 Φ_{pi} — стоимость i -го варианта исполнения;
 Φ_{max} — максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

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

$$I_m^a = \sum_{i=1}^n a_i \times b_i^a, \quad (17)$$

$$I_m^p = \sum_{i=1}^n a_i \times b_i^p, \quad (18)$$

где I_m^a, I_m^p — интегральный показатель ресурсоэффективности вариантов;
 a_i — весовой коэффициент i -го параметра;
 b_i^a, b_i^p — бальная оценка i -го параметра для аналога и разработки, устанавливается экспертным путём по выбранной шкале оценивания;
 n — число параметров сравнения.

В качестве параметров сравнения были взяты критерии конкурентоспособности из подраздела 5.1.2 с теми же весовыми коэффициентами. Результаты определения интегрального показателя ресурсоэффективности приведены в таблице 37. Под аналогом 1 подразумевается Mask R-CNN, под аналогом 2 — Faster R-CNN из подраздела 5.1.2.

Таблица 37 — Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент параметра	Текущий проект	Аналог 1	Аналог 2
Точность	0,25	5	3	4
Требование к вычислительным ресурсам	0,10	4	5	5
Возможность дообучения	0,05	4	3	3
Универсальность	0,15	5	1	2
Востребованность	0,10	4	2	3

Критерии	Весовой коэффициент параметра	Текущий проект	Аналог 1	Аналог 2
Требовательность к данным	0,15	5	2	3
Стоимость разработки	0,10	5	3	4
Ожидаемый жизненный цикл	0,10	5	4	2
Итого	1	4,75	2,75	3,30

Интегральный показатель эффективности разработки ($I_{финр}^p$) и аналога ($I_{финр}^a$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{финр}^p = \frac{I_m^p}{I_{\Phi}^p} \quad (19)$$

$$I_{финр}^a = \frac{I_m^a}{I_{\Phi}^a} \quad (20)$$

Сравнение интегрального показателя эффективности текущего проекта и аналогов позволит определить сравнительную эффективность проекта. Сравнительная эффективность проекта определяется по формуле:

$$\mathcal{E}_{ср} = \frac{I_{финр}^p}{I_{финр}^a}, \quad (21)$$

где $\mathcal{E}_{ср}$ — сравнительная эффективность проекта;

$I_{финр}^p$ — интегральный показатель разработки;

$I_{финр}^a$ — интегральный технико-экономический показатель аналога.

Результаты расчётов представлены в таблице 38.

Таблица 38 — Сравнительная эффективность исследования

Показатели	Текущий проект	Аналог 1	Аналог 2
Интегральный финансовый показатель разработки	1	0,579	0,695
Интегральный показатель ресурсоэффективности разработки	4,75	2,75	3,30
Интегральный показатель эффективности	4,75	4,75	4,748

Показатели	Текущий проект	Аналог 1	Аналог 2
Сравнительная эффективность вариантов исполнения	1	1	1,0004

Сравнение значений интегральных показателей эффективности показывает, что текущий проект и аналоги схожи по эффективности, но аналоги являются менее точными, менее быстрыми и более зависимыми от входных данных.

5.6 Выводы по разделу

Таким образом, в данном разделе были рассмотрены этапы проектирования и создания конкурентоспособных разработок, отвечающих требованиям ресурсоэффективности и ресурсосбережения.

Целью данной работы является разработка и исследование моделей СНС для детектирования одного или нескольких объектов различных классов на изображении.

Была построена карта потенциальных потребителей, которая показала возможные темы компаний, использующих машинное обучение.

Был проведён SWOT-анализ, который описывает сильные и слабые стороны проекта, а также возможности и угрозы для проекта. Была проведена оценка готовности проекта к коммерциализации, перспективы развития были признаны выше среднего. Для повышения оценки готовности проекта к коммерциализации необходимо провести консультации с экспертами в сферах, затрагиваемых разработкой, сформировать команду и найти финансирование.

На этапе инициации проекта определяются внутренние и внешние заинтересованные стороны проекта, а также цели и ожидаемые результаты проекта. График проекта отображается на диаграмме Ганта, которая показывает, кто из исполнителей (руководитель, заказчик, эксперты и магистрант) выполнил какие задачи и за сколько рабочих дней.

Стоимость материалов включена в бюджет инженерного проекта. Рассчитана основная и дополнительная заработная плата исполнителей проекта. Бюджет проекта составил 979 812,47 руб. после вычетов на социальные нужды и накладные расходы.

Определена сравнительная эффективность проекта, которая, согласно расчётам, не уступает аналогам.

ГЛАВА 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Целью работы является разработка и исследование моделей свёрточных нейронных сетей (СНС) для детектирования (обнаружения, локализации и классификации) одного и более объектов различных классов на изображении.

Разработка модели СНС представляет собой создание новых моделей СНС на базе семейства моделей YOLO с использованием Inception – ResNet-модулей. Для обучения моделей СНС используется датасет изображений, содержащий объекты четырёх классов: «Птица», «БПЛА самолётного типа», «БПЛА вертолётного типа» и «Неизвестный объект». Результатом вычисления модели СНС для каждого изображения является ограничивающий прямоугольник вокруг объекта и его класс.

Разработанные модели СНС могут быть использованы в исследовательских учреждениях, предприятиях или независимыми исследователями в качестве детектирования объектов указанных выше классов на изображениях.

В данном разделе рассматриваются опасные и вредные факторы, оказывающие влияние на производственную деятельность разработчика, воздействие объекта исследования на окружающую среду, правовые и организационные вопросы и мероприятия в чрезвычайных ситуациях

Рабочей зоной при разработке алгоритма является рабочее место для выполнения работ сидя с использованием ПЭВМ.

6.1 Правовые и организационные вопросы обеспечения безопасности

6.1.1 Специальные правовые нормы трудового законодательства

Регулирование отношений между организацией и сотрудниками, в том числе требований к продолжительности рабочего времени и оплаты труда осуществляется законодательством РФ, а именно трудовым кодексом РФ [56].

Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. Для работников до 16 лет — не более 24 часов в неделю, от 16 до 18 — не более 35 часов, как и для инвалидов I и II группы. Для работников, условия труда на рабочих местах которых по результатам специальной

оценки условий труда отнесены к вредным условиям труда 3 или 4 степени или опасным условиям труда — не более 36 часов. Продолжительность рабочего времени и размер оплаты труда для конкретного работника устанавливается трудовым договором. В течение рабочего дня работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не менее 30 минут и не более двух часов. Всем работникам предоставляются выходные дни (ежедневный непрерывный отдых). При пятидневной рабочей неделе работникам предоставляются два выходных дня в неделю, при шестидневной рабочей неделе — один выходной день.

Организация обязана предоставлять ежегодные отпуска с сохранением места работы (должности) и среднего заработка, продолжительностью 28 календарных дней. Для работников, занятых на работах с опасными или вредными условиями, предусматривается дополнительный отпуск.

6.1.2 Организационные мероприятия при компоновке рабочей зоны

В связи с тем, что разработка и эксплуатация результатов происходит на ПЭВМ, требуется соблюдение правовых норм к ПЭВМ и организации рабочего места, описываемых в ГОСТ 12.2.032-78, ГОСТ 21889-76, ГОСТ 22269-76, ГОСТ Р 50923-96 и СанПиН 1.2.3685-21 [57, 58, 59, 60, 61].

Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы.

Конструкцией производственного оборудования и рабочего места должно быть обеспечено оптимальное положение работающего, которое достигается регулированием:

- высоты рабочей поверхности, сиденья и пространства для ног;
- высоты сиденья и подставки для ног (при нерегулируемой высоте рабочей поверхности).

В соответствии с ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономиче-

ские требования» [57] рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учётом его количества, размеров и характер выполняемой работы.

Выполнение требований на рабочем месте отражено ниже в таблице 39.

Таблица 39 — Требования к организации рабочего места при работе с ПЭВМ

Требование	Требуемое значение	Значение параметров в помещении
Высота рабочей поверхности стола	Регулируемая высота (680-800 мм.); Нерегулируемая высота (725 мм.).	Нерегулируемая высота (750 мм.)
Рабочий стул	Подъёмно-поворотный, регулируемый по высоте и углу наклона спинки.	Соответствует
Расположение экрана видеомонитора от глаз пользователя	600-700 мм.	Соответствует

При работе двумя руками органы управления размещают с таким расчётом, чтобы не было перекрещивания рук [57].

Согласно ГОСТ Р 50923-96 [60], дисплей на рабочем месте оператора должен располагаться так, чтобы изображение в любой его части было различимо без необходимости поднять или опустить голову. Также, дисплей на рабочем месте должен быть установлен ниже уровня глаз оператора, причём угол наблюдения экрана оператором относительно горизонтальной линии взгляда не должен превышать 60°.

Клавиатура на рабочем месте оператора должна располагаться с возможностью свободного перемещения так, чтобы обеспечивалась оптимальная видимость экрана [60].

На рабочем месте оператора должны обеспечиваться оптимальные микроклиматические условия в холодный и тёплый периоды года. Температура

воздуха на рабочем месте в холодный период года должна быть от 22 °С до 24 °С, в тёплый период года — от 23 °С до 25 °С, причём разница температуры на уровне пола и уровне головы оператора в положении сидя не должна превышать 3 °С. Относительная влажность воздуха на рабочем месте оператора должна составлять 40-60%. Скорость движения воздуха на рабочем месте оператора должна быть 0,1 м/с [60].

6.2 Производственная безопасность

Для обеспечения безопасности во время эксплуатации и разработки программы, необходимо провести анализ вредных и опасных воздействий на человека, которые могут возникать при разработке или эксплуатации проекта. Производственный фактор является вредным, в том случае если он приводит к заболеванию работника. В случае если его воздействие может привести к травме, то фактор является опасным. Выявленные вредные и опасные факторы приведены в таблице 40.

Таблица 40 — Возможные опасные и вредные производственные факторы на рабочем месте

№	Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1	Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий;	ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов [62]; ГОСТ 12.1.019-2017 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты [63];
2	Производственные факторы, связанные с отсутствием или недостатком необходимого искусственного освещения	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 [64]; ГОСТ Р 55710-2013 Освещение рабочих мест внутри зданий. Нормы и методы измерений [65];

№	Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
3	Производственные факторы, обладающие свойствами физического воздействия на организм человека (рабочая поза)	ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования [57]; ГОСТ 21889-76 Система "Человек-машина". Кресло человека-оператора. Общие эргономические требования [58]. ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места [59]; ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения [60];
4	Производственные факторы, обладающие свойствами психофизиологического воздействия на организм человека (умственное перенапряжение, перенапряжение анализаторов, монотонность труда)	МР 2.2.9.2311-07 Профилактика стрессового состояния работников при различных видах профессиональной деятельности. Методические рекомендации [66];
5	Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [61];
6	Производственные факторы, связанные с акустическими колебаниями в производственной среде (повышенным уровнем и другими неблагоприятными характеристиками шума)	ГОСТ 12.1.003-2014 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности [67]; ГОСТ 12.1.029-80 Система стандартов безопасности труда (ССБТ). Средства и методы защиты от шума. Классификация [68]; СП 51.13330.2011. Защита от шума. Актуализированная редакция СНиП 23-03-2003 [69];

6.2.1 Опасные производственные факторы

6.2.1.1 Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий

Электробезопасность — система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества. Нормы электробезопасности на рабочем месте и вопросы требований к защите от поражения электрическим током приведены в ГОСТ 12.1.038-82 и ГОСТ 12.1.019-2017 [62, 63].

Электрический ток оказывает на человека термическое, электролитическое, биологическое и механическое воздействие. Действие электрического тока на человека приводит к травмам или гибели людей. Для переменного тока частотой 50 Гц допустимое значение напряжения прикосновения составляет не более 2,0 В, а силы тока — 0,3 мА, для переменного тока частотой 400 Гц, соответственно — 3,0 В и 0,4 мА, для постоянного тока — 8,0 В и 1,0 мА соответственно [62].

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землёй металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования. С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены и в обязательном порядке иметь нулевую фазу.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности:

- перед началом работы убедиться, что выключатели, розетки закреплены и не имеют оголённых токоведущих частей;
- не включать в сеть компьютеры и другую оргтехнику со снятыми крышками;
- при обнаружении неисправности компьютера необходимо выключить его и отключить от сети;
- запрещается загромождать рабочее место лишними предметами;
- при несчастном случае необходимо немедленно отключить питание электроустановки, вызвать скорую помощь и оказать пострадавшему первую помощь до прибытия врача, согласно правилам;
- дальнейшее продолжение работы возможно только после устранения причины поражения электрическим током.

6.2.2 Вредные производственные факторы

6.2.2.1 Производственные факторы, связанные с отсутствием или недостатком необходимого искусственного освещения

Естественное и искусственное освещение рабочего места оказывает влияние на физическое и психологическое состояние пользователя, что неблагоприятно сказывается на его работе. Не надлежащего качество освещения может привести к ухудшению зрения.

Согласно СП 52.13330.2016 [64] при работах III зрительного разряда и подразряда Г (работы высокой точности) освещённость при системе общего освещения должна быть не ниже $E_n = 200$ Лк.

Растёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента использования светового потока, учитывающим световой поток, отражённый от потолка и стен. Длина помещения $A = 4$ м., ширина $B = 2,4$ м., высота $H = 2,6$ м. Высота рабочей поверхности над полом $h_{рп} = 0,75$ м.

Площадь помещения:

$$S = A \times B = 4 \times 2,4 = 9,6 \text{ м}^2 \quad (22)$$

Приблизительные коэффициенты отражения из «МУ Методика расчета системы общего равномерного искусственного освещения» [70]:

- для светлого деревянного потолка $\rho_n = 50\%$;
- для стен, оклеенных светлыми обоями $\rho_{ст} = 30\%$;

Для нормальных помещений с хорошим отражением потолка и стен выбираются открытые двухламповые светильники типа ОД, ОДОР, ШОД, ОДО или ООД. Так как высота помещения 2,6 м, то в соответствии с наименьшей допустимой высотой подвеса над полом необходимо выбрать тип светильника ШОД. Наивыгоднейшее расположение светильников данного типа берётся с критерием $\lambda = 1,1 - 1,3$. Световой поток одной лампы $\Phi = 3750$ Лм. По паспорту длина светильника $l_{св} = 1228$ мм., ширина $b_{св} = 284$ мм., мощность лампы $P_{л} = 65$ Вт. Расстояние светильника от перекрытия $h_c = 0,1$ м.

Высота светильника над полом определяется по формуле:

$$h_{п} = H - h_c = 2,6 - 0,1 = 2,5 \text{ м} \quad (23)$$

Из методической таблицы наименьшая допустимая высота подвеса над полом для типов светильников ОД при одиночной установке является 3,5 м., что не соответствует фактическому значению.

Расчётная высота светильника над рабочей поверхностью определяется по формуле:

$$h = h_{п} - h_{рп} = 2,5 - 0,75 = 1,75 \text{ м} \quad (24)$$

Расстояние между светильниками определяется по формуле:

$$L = \lambda * h = 1,4 * 1,75 = 2,45 \text{ м} \quad (25)$$

Расстояние от стены до торца светильника определяется по формуле:

$$l = \frac{L}{3} = \frac{2,45}{3} \approx 0,816 \text{ м} \quad (26)$$

Количество рядов светильников с люминесцентными лампами определяется по формуле:

$$n_{ряд} = \frac{\left(B - \frac{2}{3} \times L\right)}{L} + 1 = \frac{\left(2,4 - \frac{2}{3} \times 2,45\right)}{2,45} + 1 \approx 1,313 = 1 \quad (27)$$

Количество светильников с люминесцентными лампами определяется по формуле:

$$n_{\text{св}} = \frac{\left(A - \frac{2}{3} \times L\right)}{l_{\text{св}} + 0,5} = \frac{\left(4 - \frac{2}{3} \times 2,45\right)}{1,228 + 0,5} \approx 1,323 = 1 \quad (28)$$

Общее количество светильников с люминесцентными лампами в помещении определяется по формуле:

$$N = n_{\text{ряд}} \times n_{\text{св}} = 1 \times 1 = 1 \quad (29)$$

Исходя из того, что в светильнике используется 2 лампы, общее количество ламп в помещении при $N = 1$ равно $N_{\text{л}} = 2$.

Индекс помещения определяется по формуле:

$$i = \frac{S}{h \times (A + B)} = \frac{9,6}{1,75 \times (4 + 2,4)} \approx 0,857 = 0,9 \quad (30)$$

Из методической таблицы коэффициент использования светового потока, показывающий какая часть светового потока ламп попадает на рабочую поверхность, для светильников типа ОД с люминесцентными лампами при $\rho_n = 50\%$, $\rho_{\text{ст}} = 30\%$ и индексе помещения $i = 0,9$ равен $\eta = 42\% = 0,42$.

Из методической таблицы коэффициент запаса для помещений с малым выделением пыли $K_3 = 1,5$, коэффициент неравномерности $Z = 1,1$.

Световой поток определяется по формуле:

$$\Phi_{\text{расч}} = \frac{E_{\text{н}} \times S \times K_3 \times Z}{N_{\text{л}} \times \eta} = \frac{200 \times 9,6 \times 1,5 \times 1,1}{2 \times 0,42} \approx 3771,42 = 3771 \text{ лм} \quad (31)$$

Выбираем ближайшую стандартную лампу — ЛД 65 Вт с потоком 3750 Лм. Делаем проверку выполнения условия:

$$\begin{aligned} -10\% &\leq \frac{\Phi_{\text{стан}} - \Phi_{\text{расч}}}{\Phi_{\text{стан}}} \times 100\% \leq +20\% = \\ &= -10\% \leq \frac{3750 - 3771}{3750} \times 100\% \leq +20\% = \\ &= 10\% \leq -0,56\% \leq +20\% \end{aligned} \quad (32)$$

Таким образом: $10\% \leq -0,56\% \leq +20\%$, необходимый световой поток светильника не выходит за пределы требуемого диапазона.

Номинальная мощность осветительной системы рассчитывается по формуле:

$$P = N_{\text{л}} \times P_{\text{л}} = 2 \times 65 = 130 \text{ Вт} \quad (33)$$

Таким образом световой поток светильника не выходит за пределы требуемого диапазона.

6.2.2.2 Производственные факторы, обладающие свойствами физического воздействия на организм человека

Рабочая поза работника при работе с ПЭВМ относится к свободной. К свободным позам относят удобные позы сидя, которые дают возможность изменения рабочего положения тела или его частей (откинуться на спинку стула, изменить положение ног, рук). Также, работа с ПЭВМ характеризуется недостаточной уровнем двигательной активности (гипокинезия) [66].

Для поддержания корректной рабочей позы работника необходимо соблюдение ГОСТ 12.2.032-78, ГОСТ 21889-76, ГОСТ 22269-76 и ГОСТ Р 50923-96 [57, 58, 59, 60], описывающих общие требования к рабочему месту оператора ПЭВМ.

Длительное (25 и более процентов времени смены) пребывание работников различных профессий в физиологически нерациональных рабочих позах является утомительным для организма из-за постоянной статической нагрузки на отдельные мышечные группы, в основном на группы мышц рук, шеи, плечевого пояса, поясничной области, что может явиться причиной возникновения патологических нарушений [66].

Для устранения накопленной усталости и нагрузки на организм человека необходимо выполнять комплекс физических упражнений на координацию движений [66].

6.2.2.3 Производственные факторы, обладающие свойствами психофизиологического воздействия на организм человека

При разработке и исследовании моделей свёрточных нейронных сетей необходим контроль процессов сбора и разметки изображений, обучение самой

модели, исследование и изучение результатов модели, которые вызывают зрительную и умственную нагрузку на организм человека.

При умственной нагрузке необходима длительность сосредоточенного внимания, выраженная ответственность, плотность сигналов и сообщений в единицу времени по МР 2.2.9.2311-07 [66]. Оказывает угнетающее влияние на психическую деятельность ухудшаются функции внимания (объем, концентрация, переключение), памяти (кратковременной и долговременной), восприятия (появляется большое число ошибок). При зрительной нагрузке необходима высокая координация сенсорных и моторных элементов зрительной системы. Вызывает головную боль, ухудшение зрения, астенопию — патологического состояния, связанного с быстрым переутомлением глаз.

Для устранения накопленной усталости и нагрузки на организм человека необходимо выполнять комплекс упражнений на концентрацию внимания, комплекс упражнений на глаз, использовать методику психической саморегуляции [66].

6.2.2.4 Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды

Комфортные условия для работы создаются оптимальным сочетанием температуры, относительной влажности и скорости движения воздуха. На рабочих местах пользователей ПЭВМ должны обеспечиваться оптимальные параметры микроклимата в соответствии с СанПиН 1.2.3685-21 [61]. Требования к параметрам микроклимата определены в таблице 41; согласно СанПиН 1.2.3685-21 [61], степень физической тяжести работы инженера-программиста относится к категории лёгких работ — Ia.

Таблица 41 — Допустимые величины параметров микроклимата на рабочих местах в помещениях

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	24,1 – 25,0	19,0 – 26,0	15 – 75	0,1
Тёплый	25,1 – 28,0	20,0 – 29,0	15 – 75	0,2

При температуре воздуха на рабочих местах 25 °С и выше максимально допустимые величины относительной влажности воздуха не должны выходить за пределы:

- 70% — при температуре воздуха 25 °С;
- 65% — при температуре воздуха 26 °С;
- 60% — при температуре воздуха 27 °С;
- 55% — при температуре воздуха 28 °С.

При температуре воздуха 26-28 °С скорость движения воздуха для тёплого периода года должна соответствовать диапазонам: 0,1-0,2 м.

Для поддержания оптимальных значений микроклимата используется системы отопления и кондиционирования воздуха, тепловая изоляция нагретых поверхностей оборудования. При исследовании микроклимата было выявлено, что в помещении, где выполнялась работа, параметры микроклимата соответствуют требованиям СанПиН.

6.2.2.5 Производственные факторы, связанные с акустическими колебаниями в производственной среде

Шум является одним из распространённых в производстве вредных факторов. Его создают работающее оборудование, преобразователи напряжения, работающие осветительные приборы дневного света и другие источники шума. Шум может стать причиной снижения работоспособности и повышенной утомляемости. Значительные превышения уровня шума на рабочем месте вызывают необратимые изменения в органах слуха человека, также оказывают неблаго-

приятное влияние на весь организм человека через нервную систему. В результате ослабляется внимание, ухудшается память, снижается реакция, что вызывает увеличение числа ошибок при работе.

Требования к допустимому уровню шума были описаны в СП 51.13330.2011 [69]. Согласно данному документу, допустимый уровень шума составляет 60 дБА для рабочих помещений, и 50 дБА для офисных помещений и научно-исследовательских организаций.

Допустимый уровень шума на рабочем месте не превышает установленного значения. Источники шума (АЦПУ, принтеры и т.п.), уровни шума которого превышают нормированные, находятся вне помещения с ПЭВМ.

Для защиты от воздействия шума на организм человека возможно применение архитектурно-планировочных методов, которые включают в себя: рациональное размещение технологического оборудования, машин и механизмов, рациональное размещение рабочих мест, создание шумозащитных зон в различных местах нахождения человека согласно ГОСТ 12.1.029-80 [68].

6.3 Экологическая безопасность

На данном рабочем месте выявлен предполагаемый источник загрязнения окружающей среды, а именно воздействие на литосферу и атмосферу в результате образования отходов при поломке предметов вычислительной техники и оргтехники и неправильной утилизации отходов. Нормативы экологической безопасности установлены ГОСТ 17.4.3.04-85 и ГОСТ Р 53692-2009 [71, 72].

Согласно ГОСТ Р 53692-2009 [72], вышедшее из строя ПЭВМ и сопутствующая оргтехника относится к IV классу опасности и подлежит специальной утилизации. Для оказания наименьшего влияния на окружающую среду, необходимо проводить специальную процедуру утилизации ПЭВМ и оргтехники, при которой более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки. При этом она должна соответствовать процедуре утилизации ГОСТ Р 53692-2009 [72].

Первым этапом является утилизация обезвреженных (инертных) отходов. Во время утилизации может быть произведена переработка бракованных или

вышедших из употребления видов продукции, изделий, их составных частей и отходов от них путём разборки (разукрупнения), переплавки, использования других технологий с обеспечением рециркуляции (восстановления) органической и неорганической составляющих.

Вторым этапом является безопасное размещение отходов I-IV классов опасности на соответствующих полигонах или уничтожение.

Литиевые батареи, используемые в ПЭВМ, и люминесцентные лампы, применяющиеся для искусственного освещения рабочих мест, требуют особой утилизации.

При неправильной утилизации литиевых батарей, такие металлы, как кобальт, никель и марганец — токсичны и могут загрязнять источники водоснабжения и экосистемы, влияя на литосферу и гидросферу. Согласно СанПиН 1.2.3685-21 [61], определены предельно допустимые концентрации (ПДК) вредных веществ, представленных в таблице 42.

Таблица 42 — Предельно допустимые концентрации (ПДК) вредных веществ

Наименование вещества	Предельно допустимые концентрации, мг/м ³			Класс опасности
	Концентрация, предотвращающая раздражающее действие, рефлекторные реакции, запахи при воздействии до 20–30 минут максимальная разовая	Концентрация, обеспечивающая допустимые (приемлемые) уровни риска при воздействии не менее 24 часов среднесуточная	Концентрация, обеспечивающая допустимые (приемлемые) уровни риска при хроническом (не менее 1 года) воздействии среднегодовая	
Кобальт	—	0,0004	0,0001	II
Никель	—	0,001	0,00005	II
Марганец	0,01	0,001	0,00005	II
Ртуть	—	0,0003	0,00003	I

В люминесцентных лампах присутствует от 10 до 70 мг. ртути, которая относится к чрезвычайно опасным химическим веществам (ПДК указаны в таб-

лице 4) и может стать причиной отравления живых существ, а также загрязнения атмосферы, гидросферы и литосферы. Сроки службы таких ламп составляют около пяти лет, после чего их необходимо сдавать на переработку в специальных пунктах приёма.

6.4 Безопасность в чрезвычайных ситуациях

К чрезвычайным ситуациям, которые могут возникнуть во время разработки можно отнести возможность возникновения и распространения пожара, выход из строя электрической системы и землетрясение. Наиболее типичной чрезвычайной ситуацией является пожар, так как специфика работы заключается во взаимодействии с ПЭВМ, что может являться причиной возгорания и угрожать жизни людей.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д. В помещениях с ПЭВМ повышен риск возникновения пожара из-за присутствия множества факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха [73].

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие различного рода неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также использовать электропровода и кабели с повреждённой или потерявшей защитные свойства изоляцией [73].

Для предотвращения возникновения и распространения пожара необходимо соблюдение техники пожарной безопасности. Общие требования к пожарной безопасности описаны в ГОСТ 12.1.004-91 [73]. Исходя из возможных причин возникновения пожара для его предотвращения могут быть применены следующие меры:

- регулярная проверка исправности работы электрических приборов, целостности соединительных проводов и вилок;
- соблюдение работниками техники безопасности при работе с электрическими приборами;

- заземления частей электрических приборов для снижения статического заряда.

В случае возникновения пожара необходимо:

- незамедлительно сообщить об это в пожарную охрану;
- принять меры по эвакуации людей;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

Для тушения пожара следует применять ручные углекислотные огнетушители (ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей [73].

6.5 Выводы по разделу

Значение всех производственных факторов на изучаемом рабочем месте соответствует нормам, которые также были продемонстрированы в данном разделе. Для минимизации влияния физиологического и психофизиологического воздействия на организм человека, достаточно соблюдать меры, приведённые в МР 2.2.9.2311-07 [66].

Категория помещения по электробезопасности, согласно ПУЭ, соответствует первому классу — «помещения без повышенной опасности» [74].

Согласно правилам по охране труда при эксплуатации электроустановок персонал должен обладать I группой допуска по электробезопасности. Присвоение I группы по электробезопасности проводится работником из числа электротехнического персонала, имеющего группу III по электробезопасности или специалистом по охране труда, имеющим группу IV по электробезопасности или выше, назначенным распоряжением руководителя организации [75].

Категория тяжести труда в лаборатории по СанПиН 1.2.3685-21 относится к категории Ia (работы, производимые сидя и сопровождающиеся незначительным физическим напряжением) [61].

Помещение, в котором проводилось исследование, относится к категории помещений группы В1-В4. Характеристика веществ и материалов, находящихся в помещении: горючие и трудногорючие жидкости, твёрдые горючие и трудногорючие вещества и материалы (в том числе пыли и волокна), вещества и материалы, способные при взаимодействии с водой, кислородом воздуха или друг с другом только гореть, при условии, что помещения, в которых они находятся (обращаются), не относятся к категории А или Б [76].

Согласно постановлению «Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий», рассмотренный объект оказывает незначительное негативное воздействие на окружающую среду и относится к объектам III категории [77].

ЗАКЛЮЧЕНИЕ

Во время выполнения ВКР и акселерации работы в период технологической практики в ООО «ЛЭМЗ–Т» были получены следующие результаты.

1. Проанализированы известные модели СНС, потенциально перспективные для детектирования летающих объектов на изображениях с целью создания мобильных СКЗ реального времени. Выявлено, что наиболее перспективными для реализации этой цели являются следующие компактные модели СНС класса YOLO: YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny.

2. Для формирования основного и дополнительных датасетов разработаны алгоритмы по преобразованию форматов аннотаций и оригинальный алгоритм нахождения ограничивающих объекты прямоугольников на изображениях. Все они программно-реализованы на языке Python 3 в среде Anaconda с использованием библиотек OpenCV, Numpy, Pandas, Shapely, Scikit Learn, YAML и стандартных библиотек Python.

3. Сформирован основной датасет в виде размеченных (аннотированных) RGB изображений. В датасет включено 15 235 изображений и файлов аннотаций. Предложены два подхода к созданию дополнительных датасетов. С учетом их на базе основного датасета с помощью дополнительно разработанного алгоритмического и программного обеспечения сформированы три дополнительных датасета, каждый из которых включает изображения с объектами только одной категории по размерам. Первый такой датасет с объектами малых размеров на изображениях состоит из 10 674 изображений и аннотаций. Второй датасет включает 14 740 изображений со среднемасштабными объектами и столько же аннотаций. Третий датасет состоит из 14 926 изображений с объектами больших размеров и столько же аннотаций.

4. Разработаны, программно-реализованы, обучены и валидированы с использованием основного датасета три компактные модели СНС YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny.

5. Проведены комплексные исследования трёх обученных компактных моделей СНС с использованием тестовой выборки из основного датасета, кото-

рые показали, что модель YOLOv5 Small является наиболее эффективной среди этих моделей, поскольку удовлетворяет всем предъявляемым требованиям. Её точность детектирования объектов по метрике $mAP_{0,5}$ выше порогового значения 0,9; значение FPS значительно больше порогового значения 25, а алгоритмическая эффективность такова, что модель может выполняться в мобильных СКЗ реального времени, обладающих малыми вычислительными ресурсами.

6. Модель YOLOv5 Small также обучена на обучающих и валидационных выборках и три её варианта комплексно исследованы на тестовых выборках из дополнительных датасетов. Результаты исследований показали, что варианты модели СНС YOLOv5 Small, обученные на выборках с изображениями объектов только больших или только средних размеров, можно рекомендовать к использованию в мобильных СКЗ реального времени. Точность детектирования объектов с малыми размерами на изображениях для всех классов невысока, а для класса «Птица» значительно ниже порогового значения, поэтому такой вариант модели не рекомендуется к использованию в мобильных СКЗ.

7. На основе модели YOLOv5 Small разработана, программно-реализована, обучена и исследована новая модель СНС. Из результатов её исследования следует, что новую модель СНС можно рекомендовать к использованию в мобильных СКЗ реального времени для детектирования объектов всех четырёх классов без относительно их категории по размерам.

8. Основные результаты работы внедрены в ООО «ЛЭМЗ–Т», являющемся Центром исследований и разработок ПАО «НПО «Алмаз», при выполнении НИР по созданию мобильных СКЗ различного назначения, что подтверждено соответствующим актом.

9. Результаты работы докладывались на XX Международной научно-практической конференции студентов и молодых учёных «Молодёжь и современные информационные технологии», получен диплом III степени за лучший доклад (Приложение Р). Подготовленная по результатам работы конференции статья опубликована в сборнике трудов конференции.

СПИСОК ПУБЛИКАЦИЙ

Мунько А. С. Детектирование летающих объектов на изображениях с помощью свёрточных нейронных сетей семейства YOLO // Сборник трудов XX Международной научно-практической конференции студентов и молодых ученых «Молодежь и современные информационные технологии. Томск, 20-24 марта 2023. Томск: изд-во ТПУ. 2023. – С. 53-56.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. 2015. № 7553 (521). С. 436–444. – URL: <https://www.nature.com/articles/nature14539>.
2. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems. 2012. (25). С. 1097–1105. – URL: <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>.
3. Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // ICML 2019. 2020. – URL: <https://arxiv.org/abs/1905.11946>.
4. Tan M., Pang R., Le Q.V. EfficientDet: Scalable and Efficient Object Detection // CVPR 2020. 2020. – URL: <https://arxiv.org/abs/1911.09070>.
5. Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation // CVPR 2014. 2014. – URL: <https://arxiv.org/abs/1311.2524>.
6. He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN. 2018. – URL: <https://arxiv.org/abs/1703.06870>.
7. Zoph B., Cubuk E., Ghiasi G., Lin T., Shlens J., Le Q. Learning Data Augmentation Strategies for Object Detection. 2019. – URL: <https://arxiv.org/abs/1906.11172>.
8. Li Y., Chen Y., Wang N., Zhang Z. Scale-Aware Trident Networks for Object Detection // ICCV 2019. 2019. – URL: <https://arxiv.org/abs/1901.01892>.
9. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A. SSD: Single Shot MultiBox Detector // ECCV 2016. 2016. С. 21–37. – URL: <https://arxiv.org/abs/1512.02325>.
10. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // CVPR 2016. 2016. С. 779–788. – URL: <https://arxiv.org/abs/1506.02640>.

11. Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. 2016. – URL: <https://arxiv.org/abs/1612.08242>.
12. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. 2018. – URL: <https://arxiv.org/abs/1804.02767>.
13. Bochkovskiy A., Wang C.-Y., Liao H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. – URL: <https://arxiv.org/abs/2004.10934>.
14. Wang C.-Y., Bochkovskiy A., Liao H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 2022. – URL: <https://arxiv.org/abs/2207.02696>.
15. Law H., Deng J. CornerNet: Detecting Objects as Paired Keypoints // ECCV 2018. 2019. – URL: <https://arxiv.org/abs/1808.01244>.
16. Zhou X., Wang D., Krähenbühl P. Objects as Points. 2019. – URL: <https://arxiv.org/abs/1904.07850>.
17. Tian Z., Shen C., Chen H., He T. FCOS: Fully Convolutional One-Stage Object Detection // ICCV 2019. 2019. C. 9627–9636. – URL: <https://arxiv.org/abs/1904.01355>.
18. Liu Z., Sun M., Zhou T., Huang G., Darrell T. Rethinking the Value of Network Pruning // ICLR 2019. 2019. – URL: <https://arxiv.org/abs/1810.05270>.
19. Pedoeem J., Huang R. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. 2018. – URL: <https://arxiv.org/abs/1811.05588>.
20. Wong A., Famuori M., Shafiee M., Li F., Chwyl B., Chung J. YOLO Nano: a Highly Compact You Only Look Once Convolutional Neural Network for Object Detection. 2019. – URL: <https://arxiv.org/abs/1910.01271>.
21. Инфосистемы Джет. Object Detection. Распознавай и властвуй. Часть 1 [Электронный ресурс] // Хабр. 2020. – URL: <https://habr.com/ru/company/jetinfosystems/blog/498294/> (дата обращения: 20.02.2023).
22. Lebedzinski P. A Single Number Metric for Evaluating Object Detection Models [Электронный ресурс] // Medium. 2021. – URL:

<https://towardsdatascience.com/a-single-number-metric-for-evaluating-object-detection-models-c97f4a98616d> (дата обращения: 17.04.2023).

23. Felzenszwalb P.F., Girshick R.B., McAllester D., Ramanan D. Object Detection with Discriminatively Trained Part-Based Models // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. № 9 (32). С. 1627–1645. – URL: <https://ieeexplore.ieee.org/document/5255236>.

24. Verma Y. R-CNN vs Fast R-CNN vs Faster R-CNN - A Comparative Guide [Электронный ресурс] // Analytics India Magazine. 2021. – URL: <https://analyticsindiamag.com/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-a-comparative-guide/> (дата обращения: 17.04.2023).

25. Lin T.-Y., Goyal P., Girshick R., He K., Dollár P. Focal Loss for Dense Object Detection. 2018. – URL: <https://arxiv.org/abs/1708.02002>.

26. Law H., Teng Y., Russakovsky O., Deng J. CornerNet-Lite: Efficient Keypoint Based Object Detection. 2020. – URL: <https://arxiv.org/abs/1904.08900>.

27. Duan K., Bai S., Xie L., Qi H., Huang Q., Tian Q. CenterNet: Keypoint Triplets for Object Detection. 2019. – URL: <https://arxiv.org/abs/1904.08189>.

28. Rashwan A., Kalra A., Poupart P. Matrix Nets: A New Deep Architecture for Object Detection. 2019. – URL: <https://arxiv.org/abs/1908.04646>.

29. Girshick R. Fast R-CNN. 2015. – URL: <https://arxiv.org/abs/1504.08083>.

30. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2016. – URL: <https://arxiv.org/abs/1506.01497>.

31. Pang J., Chen K., Shi J., Feng H., Ouyang W., Lin D. Libra R-CNN: Towards Balanced Learning for Object Detection. 2019. – URL: <https://arxiv.org/abs/1904.02701>.

32. Dai J., Li Y., He K., Sun J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. 2016. – URL: <https://arxiv.org/abs/1605.06409>.

33. Yang Z., Liu S., Hu H., Wang L., Lin S. RepPoints: Point Set Representation for Object Detection. 2019. – URL: <https://arxiv.org/abs/1904.11490>.

34. Roman. Алгоритм YOLO простым языком [Электронный ресурс] // Nuances of Programming. 2021. – URL: <https://medium.com/nuances-of-programming/алгоритм-yolo-простым-языком-a4e666be4d41> (дата обращения: 27.11.2022).
35. Nelson J. Your Comprehensive Guide to the YOLO Family of Models [Электронный ресурс] // Roboflow Blog. 2021. – URL: <https://blog.roboflow.com/guide-to-yolo-models/> (дата обращения: 12.02.2023).
36. Cornell Data Science. YOLOv4 Paper Summary & Analysis [Электронный ресурс] // Nerd For Tech. 2021. – URL: <https://medium.com/nerd-for-tech/yolov4-paper-summary-analysis-602dd584fa86> (дата обращения: 03.12.2022).
37. Vochkovskiy A. AlexeyAB/darknet [Электронный ресурс] // GitHub. 2013. – URL: <https://github.com/AlexeyAB/darknet> (дата обращения: 07.03.2023).
38. Sharma A. Introduction to the YOLO Family [Электронный ресурс] // PyImageSearch. 2022. – URL: <https://pyimagesearch.com/2022/04/04/introduction-to-the-yolo-family/> (дата обращения: 11.12.2022).
39. Ultralytics, Jocher G. Ultralytics/yolov5 [Электронный ресурс] // GitHub. 2020. – URL: <https://github.com/ultralytics/yolov5> (дата обращения: 07.03.2023).
40. Boesch G. YOLOv7: The Most Powerful Object Detection Algorithm [Электронный ресурс] // viso.ai. 2022. – URL: <https://viso.ai/deep-learning/yolov7-guide/> (дата обращения: 12.12.2022).
41. Wong K.-Y. Official YOLOv7 [Электронный ресурс] // GitHub. 2022. – URL: <https://github.com/WongKinYiu/yolov7> (дата обращения: 07.03.2023).
42. CVAT. Using YOLO for annotation in CVAT [Электронный ресурс] // CVAT. 2023. – URL: <https://www.cvat.ai/post/yolo> (дата обращения: 27.11.2022).
43. COCO Consortium. COCO - Common Objects in Context [Электронный ресурс] // cocodataset.org. – URL: <https://cocodataset.org/> (дата обращения: 03.04.2023).

44. Roboflow. YOLO Darknet TXT Annotation Format [Электронный ресурс] // Roboflow. – URL: <https://roboflow.com/formats/yolo-darknet-txt> (дата обращения: 14.02.2023).
45. Roboflow. YOLOv5 PyTorch TXT Annotation Format [Электронный ресурс] // Roboflow. – URL: <https://roboflow.com/formats/yolov5-pytorch-txt> (дата обращения: 14.02.2023).
46. Roboflow. Tensorflow Object Detection CSV Annotation Format [Электронный ресурс] // Roboflow. – URL: <https://roboflow.com/formats/tensorflow-object-detection-csv> (дата обращения: 14.02.2023).
47. Roboflow. YOLO Keras TXT Annotation Format [Электронный ресурс] // Roboflow. – URL: <https://roboflow.com/formats/yolo-keras-txt> (дата обращения: 14.02.2023).
48. MTSD. AllDup - Duplicate File Finder Software [Электронный ресурс] // AllDup. – URL: https://www.alldup.info/alldup_help/alldup.php (дата обращения: 14.02.2023).
49. Kisantal M., Wojna Z., Murawski J., Naruniec J., Cho K. Augmentation for small object detection. 2019. – URL: <https://arxiv.org/abs/1902.07296> (дата обращения: 15.04.2023).
50. Roboflow, Inc. [Электронный ресурс]. – URL: <https://app.roboflow.com/> (дата обращения: 15.04.2023).
51. Wang C.-Y., Liao H.-Y.M., Yeh I.-H., Wu Y.-H., Chen P.-Y., Hsieh J.-W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. 2019. – URL: <https://arxiv.org/abs/1911.11929> (дата обращения: 17.04.2023).
52. Wang C.-Y., Bochkovskiy A., Liao H.-Y.M. Scaled-YOLOv4: Scaling Cross Stage Partial Network. 2021. – URL: <https://arxiv.org/abs/2011.08036> (дата обращения: 17.04.2023).
53. Szegedy C., Ioffe S., Vanhoucke V., Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. 2016. – URL: <https://arxiv.org/abs/1602.07261> (дата обращения: 18. 04.2023).

54. Manna S. Building Inception-Resnet-V2 in Keras from scratch [Электронный ресурс] // Medium. 2020. – URL: <https://medium.com/the-owl/building-inception-resnet-v2-in-keras-from-scratch-a3546c4d93f0> (дата обращения: 18.04.2023).

55. Приказ № 6-181 от 18.11.2022 О тарифах на электрическую энергию для населения и приравненных к нему категорий потребителей Томской области на 2023 год [Электронный ресурс] // Томскэнергосбыт. 2022. – URL: <https://tomskenergosbyt.ru/individuals/tariffes.php> (дата обращения: 16.05.2023).

56. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 19.12.2022, с изм. от 11.04.2023).

57. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования.

58. ГОСТ 21889-76 Система "Человек-машина". Кресло человека-оператора. Общие эргономические требования.

59. ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места.

60. ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения.

61. СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.

62. ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов.

63. ГОСТ 12.1.019-2017 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты.

64. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.

65. ГОСТ Р 55710-2013 Освещение рабочих мест внутри зданий. Нормы и методы измерений.
66. МР 2.2.9.2311-07 Профилактика стрессового состояния работников при различных видах профессиональной деятельности. Методические рекомендации.
67. ГОСТ 12.1.003-2014 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности.
68. ГОСТ 12.1.029-80 Система стандартов безопасности труда (ССБТ). Средства и методы защиты от шума. Классификация.
69. СП 51.13330.2011 Защита от шума. Актуализированная редакция СНиП 23-03-2003.
70. МУ Методика расчета системы общего равномерного искусственного освещения.
71. ГОСТ 17.4.3.04-85 Охрана природы. Почвы. Общие требования к контролю и охране от загрязнения.
72. ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов.
73. ГОСТ 12.1.004-91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования.
74. Правила устройства электроустановок (ПУЭ). Седьмое издание.
75. Правила по охране труда при эксплуатации электроустановок от 15.12.2020 N 903н (с изм. от 29.04.2022).
76. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
77. Приказ Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий от 31.12.2020 года N 2398 (с изм. от 07.10.2021).

ПРИЛОЖЕНИЕ А

(обязательное)

Chapter 1

Convolutional neural network models for mobile computer vision systems

Студент:

Группа	ФИО	Подпись	Дата
8ВМ1И	Мунько Александр Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор (ОИТ, ИШИТР)	Марков Н. Г.	Д.Т.Н., профессор		

Консультант — лингвист

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель (ОИЯ, ШБИП)	Ануфриева Т. Н.			

CHAPTER 1. CONVOLUTIONAL NEURAL NETWORK MODELS FOR MOBILE COMPUTER VISION SYSTEMS

1.1 Main classes of computer vision problems

The development of computer vision theory has already allowed solving a large number of applied tasks with different degrees of accuracy in robot-technology, creating unmanned transport (cars, trains, air and underwater vessels), in analyzing aerospace images, etc. Modern achievements in deep learning [1, 2, 3] have allowed to reach significant success in solving many tasks of computer vision. In particular, the application of convolutional neural network (CNN) models has led to breakthrough results in solving a number of problems of object detection on images [4].

Modern computer vision (CV) system based on CNN models allow to solve four classes of tasks of object recognition on images. They include the following classes of tasks:

1. Semantic Image Segmentation — the CNN model outputs for each pixel of the image the probability that it belongs to each class; This sort of problems are often referred to pixel-by-pixel classification problems.
2. Object Recognition — the CNN model receives an image with only one object as input, and its output shows the coordinates of the rectangle containing the object and its class; Thus, the problem of localizing and classifying one object in the image is solved.
3. Object Detection — often said to detect objects in an image: in the case of several objects in an image, the CNN model must circle a rectangle and predict the class of each object; this task is one of the fundamental tasks of computer vision and implies one-time detection, localization (determination of the location) and classification of all objects in the image. The location of an object is determined by the coordinates of its bounding box.
4. Instance Segmentation — these are the most difficult problems for CNN models when there are several objects of different classes in the image and we have not only to detect and classify objects, but also to distinguish different objects with the same class from each other.

In recent years, considerable attention has been paid to solving problems of the second and third classes. The research interest of most of the authors of early works [5, 6], as well as the authors of many modern works [7, 8] in the field of detection of objects on images is focused on achieving high accuracy of these objects' detection with the help of CNN models, while the efficiency of algorithms for computing CNN is not given proper attention. The pursuit of the accuracy of object detection leads to the development of CNN models with high computational complexity, leading to low algorithmic efficiency of these models. Hereinafter, the algorithmic efficiency will be understood as the amount of computing device resources required to achieve a certain performance in solving a given problem. Estimation of such efficiency is the main way to measure algorithmic progress in classical computer science problems. This means that complex investigations of efficiency of the proposed models of the CNN for solving problems of the second and third classes of computer vision are necessary during the creation of CV system on the basis of the CNN models.

1.2 Mobile computer vision systems

A modern trend in the development of CV system is the creation of mobile (transportable and wearable, including embedded systems) variants of CV system. Mobile CV system are also in demand when solving problems of detecting and classifying flying objects in airspace. Images and video are obtained by monitoring the airspace with the help of appropriate photo or video equipment as part of such CV system. On the received images and video one or more flying objects can appear, and in the case of several objects they can belong to different classes. Mobile CV system based on CNN models must detect, locate, track the movement of objects and classify flying objects. In fact, each CV system with the help of one or another model of CNN must solve the task of object detection. When creating mobile CV system, it is possible to use CNN models that allow to recognize a single flying object in the image, that is, to solve the task of object detection. However, CV system with CNN model can show high recognition accuracy only in the case of a single object in the analyzed image. When creating mobile CV system, it makes sense to use CNN models that

solve the problem of detecting a single flying object on the image. Below we consider only the CNN models that solve tasks of the third class: detecting objects on images.

Application of the above-mentioned models of CNN with high computational complexity in mobile CV system when solving many real CV tasks is practically impossible. The fact is that in many mobile robotics complexes, unmanned vehicles and UAVs, and similar systems, the problems of detecting objects on images using mobile CV systems included in their composition must be solved in real time. In case of flying objects during airspace monitoring, the real-time scale is determined by the speed of movement of such objects. At the same time, the tasks of object detection on images usually have to be solved in conditions of limited memory resources and computing device performance of mobile CV system. Especially these limitations have to be taken into account when developing CV system for UAVs and in the case of wearable CV system, where there are also additional severe restrictions on power consumption. All this allows us to believe that the application of complex CNN models in many real-world problems of mobile CV system is very difficult. In this regard, there is a growing interest in efficient and compact models of CV system in the development of mobile real-time CNN. Examples of works aimed at creating more efficient models of CNN for object detection on images are studies of one-step CNN models [9, 10, 11, 12, 13, 14], anchorless architectures [15, 16, 17], and compression of existing CNN models [18, 19, 20]. In fact, to solve real CV tasks using mobile CV system, one needs CNN models that allow a balance between detection accuracy and their algorithmic efficiency, and finally allow mobile CV system to function in real time and detect objects in images with acceptable accuracy.

Another important problem in the creation of many CV system, including the multi-biased systems, is the need to detect objects of different scales (linear size, area) with the help of such CV system. First of all, the detection of small-scale objects is particularly difficult for modern models of the CNN. Today, despite the general success in achieving high accuracy of detection with such models, there is still a significant gap between the accuracy of detection of objects of small and large scales.

All this points to the relevance of creating high-precision CNN models to solve the problem of detecting objects on different scales with the help of mobile CV system.

1.3 Requirements for mobile CV system to detect flying objects on images

As shown above, mobile (transportable or wearable) CV system based on modern CNN models are in great demand today for solving many applied CV problems. The computational capabilities of mobile CV system are usually quite limited, which imposes strict requirements on the computational speed of CNN models. It indicates the urgency to develop and investigate new models and to select the ones that meet the speed and accuracy requirements for recognition of objects in images put forward by the design of specific mobile CV system. These mutually exclusive requirements are not easy to satisfy when creating mobile navigation systems operating in real-time mode.

These mutually exclusive requirements should also be met by mobile CV system designed for detection (recognition, localization and classification) of flying objects in airspace. One or more flying objects can appear on the images and videos obtained during airspace monitoring by means of appropriate photo or video equipment in the composition of such CV system, and in case of several objects they can belong to different classes. The CV system should not only detect and classify them, but also to accompany moving objects in space (object tracking). At the same time, the real-time scale of operation of the mobile CV system is determined by the speed of movement of such objects in the airspace.

The practice of airspace monitoring shows that often the CV system must simultaneously detect objects of different sizes (scales). Detection and classification of small objects are of particular difficulty. For example, birds and UAVs have small sizes and often their contours are very close, so it is difficult to distinguish them. Today, despite the general success in achieving high accuracy of object detection using CNN models, there is still a significant gap between the accuracy of detection of objects of small and large scales.

On the basis of the CNN model of the YOLO class a mobile CV system is designed. The designed mobile CV system must be able to detect flying objects of the following four classes on the images:

- Bird(s);
- Aircraft-type UAVs;
- Helicopter-type UAVs;
- Unknown objects.

It is proposed to divide these flying objects in the images into three categories according to their size (by area):

- Small — up to 32×32 pixels;
- Medium — 33×33 to 96×96 pixels;
- Large — 97×97 pixels and larger.

Let us consider the quantitative characteristics of all of the above requirements to mobile CV system and, accordingly, to the underlying model of CNN. The accuracy of detecting (classifying) objects on the image is calculated using common metrics. In our study, we will use the metrics Precision — expression (1), Recall — expression (2), AP — area under the curve Precision-Recall (AUC PR), estimated using a threshold value IoU [21], mAP — the average AP value for all classes of objects (expression 3) and metric F1 — expression (4), which is a harmonic mean of Precision and Recall [22].

$$Precision = \frac{TP}{TP + FP}, \quad (1)$$

where TP — true-positive values;

FP — false-positive values.

$$Recall = \frac{TP}{TP + FN}, \quad (2)$$

where TP — true-positive values;

FN — false-negative values.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (3)$$

where N — number of object classes.

For $AP_{0.5}$ and $mAP_{0.5}$, the value of IoU is 0.5, while for the calculation of $AP_{0.5:0.95}$ and $mAP_{0.5:0.95}$ an interval of $IoU \in [0.5; 0.95]$ with step 0.05 is used. It is these metrics that are most frequently used in the works of other authors on CNN models of the YOLO class.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Flying object detection (classification) accuracy on CNN model images is high if the value of $AP_{0.5}$ metric for individual object classes and the averaged metric $mAP_{0.5}$ is greater than or equal to 0.9 for objects on images of different scales. For small objects, the value of $AP_{0.5}$ metric for individual object classes and the average metric $mAP_{0.5}$ is greater than or equal to 0.7.

The next quantitative characteristic of mobile CV system is the speed of detecting objects on the image (the speed of calculations of the CNN model), measured as the number of images analyzed per second in FPS (Frames Per Second). The FPS value, allowing to consider that the mobile CV system satisfies the requirement for the speed of detection of objects using the CNN model, should be not less than 25.

The last quantitative characteristic is the algorithmic efficiency of the CNN model. Taking into account that it is defined as the amount of computing device resources required to achieve certain performance of the CNN model in solving a given problem, it is necessary to calculate two characteristics when estimating the algorithmic efficiency of the CNN model: the computational complexity and the size of the CNN model. The computational complexity of the CNN model in GFLOPs is the number of multiplications, addition and comparison operations on 16-bit floating-point numbers. The size of the CNN model (architecture compactness), in MB, is the memory size of the CV system computing device required to store the weight coefficients of convolutional filters and intermediate buffers when calculating the CNN model. The threshold values of computational complexity and the size of the CNN model that allow to assume that the mobile CV system (and, accordingly, the CNN model in its basis) meets the requirements of its computing device developers are

specified in the technical specification for the creation of CV system. We will estimate the values of these two parameters during the study of CNN models, focusing on some average characteristics of modern and widespread graphics processors (e.g., Nvidia), systems on a chip with programmable logic integrated circuits (Programmable logic device, PLD) (e.g., Xilinx company) and similar computing devices, unless otherwise stipulated in the technical specification for CV system.

1.4 Analysis of CNN models for object detection

The results of the analysis of CNN models used to select models that meet at least some of the requirements of mobile CV system for the detection of flying objects are offered below.

1.4.1 First CNN models for object detection

Early algorithms for detecting objects in an image based on deep learning adapted classifiers to solve this problem. CNN for classification was applied sequentially to different regions of the image by the sliding window method [23], or, in later works, to certain generated regions of interest in the image [5]. A high classifier score in some region of the image was regarded as a detection of an object in that region. This two-stage CNN architecture for object detection is complicated and slow - several hundred to several thousand network passes over different image areas are required to process one image, so such architectures are not suitable for mobile CV system.

One of the most widely used approaches to the detection of objects on images is the use of the R-CNN (Region based Convolutional Neural Network) model, which was proposed in 2014. The R-CNN model consists of two main parts: the first one - Region Proposal Network (RPN) to identify areas in the image that may contain objects, and the second one — classification and localization of objects. This model showed high detection accuracy, but had low image processing speed [24].

The Fast R-CNN model was developed to solve the low speed problem of the R-CNN model. It uses a single convolutional network for feature extraction and object classification, and combines RPN and object detector into a single architecture.

The Fast R-CNN model showed higher speed compared to R-CNN while maintaining a high level of accuracy in object detection [24].

The next model Mask R-CNN is an extension of the Faster R-CNN, which allows not only the detection of objects, but also the segmentation of images. Thus, it makes it possible to determine not only the location of the object, but also its shape. In spite of the fact that the listed models give very high accuracy of object detection, the speed of their calculations, even for the improved models Fast R-CNN and Mask R-CNN, is very low. Moreover, their algorithmic efficiency is low. All this does not allow us to consider such models as the basis for creating mobile CV system, and even less so for real-time CV system.

1.4.2 CNN Models of the YOLO class

YOLO (You Only Look Once) models are another popular model for object detection, which can also solve the problem of image segmentation. The researchers who proposed the YOLO model [10] were the first to apply a different, non classical approach to solving the problem of detection: they considered detection as a problem of regression of frames (rectangles that bound objects) and classes of objects. The YOLO class CNN models, like other CNN models with a single-stage architecture, process the image at its original scale in a single pass. The advantages of such single-stage CNN models as compared to two-stage models are as follows:

- predictions are based on a global context, as the model processes the whole image;
- predicting the frames and classes of all objects in the image in a single pass through the network greatly speeds up the detection process.

CNN models for predicting frames and classes are divided into one-stage and two-stage, anchor and anchorless models (Table 1).

Table 1 — CNN models for predicting bounding rectangles and classes

	Anchor	Anchorless
One-stage	SSD [9], YOLO [10], RetinaNet [25]	CornerNet [15, 26], CenterNet [27], MatrixNet [28], FCOS [17]

	Anchor	Anchorless
Two-stage	R-CNN [5], Fast R-CNN [29], Faster R-CNN [30], Mask R-CNN [6], Libra R-CNN [31], R-FCN [32]	RepPoints [33]

The YOLO model is a modern deep learning algorithm, which is widely used for object detection. It was developed by Joseph Redmon and Ali Farhadi in 2016 [34]. It was this model that started the class of YOLO models created in the following years.

The main difference of the YOLO model from other CNN models used for object detection is that it identifies objects very quickly in real time. The principle of YOLO model operation implies input of the whole image at once, which passes through the CNN only once, while in other algorithms this process takes place repeatedly. The YOLO model has the advantage of high-speed object detection, which other algorithms don't have [34].

Models CNN class YOLO can find several limiting rectangles for the same object, so to select only the one that most accurately describes the object, the formula for calculating IoU and the algorithm Non-Maximum Suppression (NMS) [21].

The YOLOv2 model was released in 2017 and received an honorable mention at CVPR 2017 [11]. A number of iterative improvements were made to its architecture over the original YOLO model, including BatchNorm, higher resolution, and binding blocks.

The YOLOv3 model was developed in 2018 and built on previous models (Figure 1) by adding an objectivity score to the bounding rectangle prediction, adding connections to the backbone network layers, and making predictions at three separate levels of detail to improve model performance on smaller objects [12].

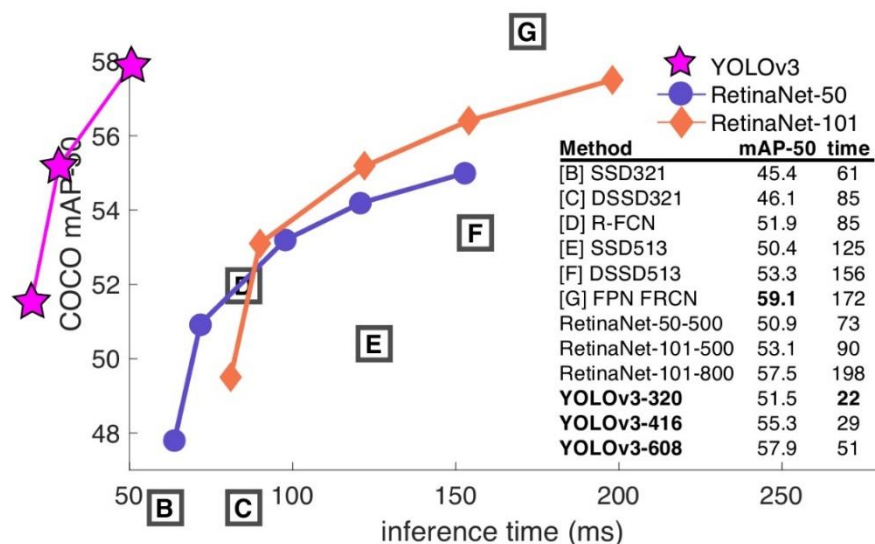


Figure 1 — Comparison on MS COCO dataset of mAP metrics and image processing speed using YOLOv3 model with other CNN models

After the release of the YOLOv3 CNN, Joseph Redmon withdrew from computer vision research. However, researchers such as Alexey Bochkovskiy and innovators such as Glenn Jocher began to open source about their advances in CV research. Groups such as Baidu have also released their own implementations of the YOLO model (e.g., PP-YOLO), demonstrating improved mAP metric values and reduced model computation latency [35].

The YOLOv4 CNN model is a one-stage model of object detection and classification, based on the previous original YOLO models. Modern image object detectors usually consist of two components: a backbone and a head (Fig. 2), which is also the case for this model.

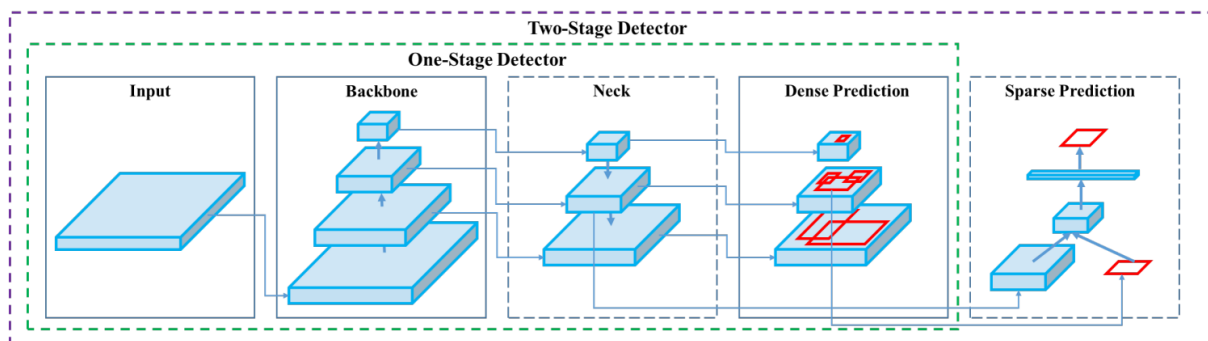


Figure 2 — The architecture of CNN models in general form

The base is usually pre-trained on a large image classification dataset, such as ImageNet, and serves to code relevant information about the input data. The head

predicts object classes and information about the bounding rectangle. Also, in the YOLOv4 model a special layer, the neck, is a link between the base and head layers. It serves to collect feature maps from different stages of the network [36] and increase the speed of image processing (Fig. 3). YOLOv4 model uses new features: WRC, CSP, CmBN, SAT, Mish-activation, Mosaic data augmentation, CmBN, DropBlock regularization and CIoU loss [13]. The YOLOv4 model, like the YOLOv3 model, is implemented using the Darknet framework [37].

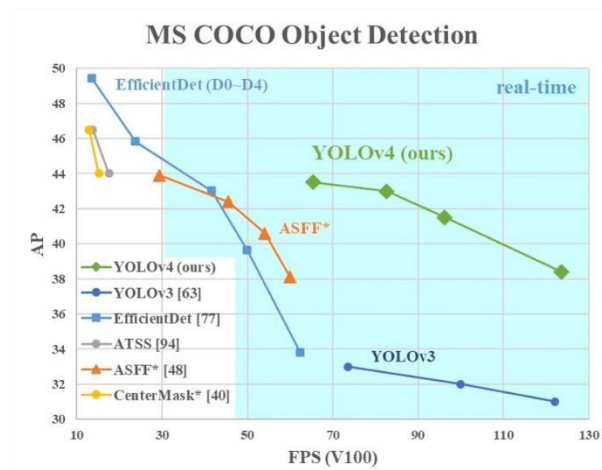


Figure 3 — Comparison on MS COCO dataset of mAP metrics and image processing speed using YOLOv4 model with other CNN models

The YOLOv5 CNN model has a family of architectures for object detection and classification pre-trained on the MS COCO dataset (Figure 4). This is the only object detector from the YOLO class that does not have a scientific paper justifying the improvement of the architecture over previous versions of YOLO. The lack of a scientific paper is due to the fact that this architecture is being developed by the community. The YOLOv5 CNN model is implemented by means of the PyTorch framework, eliminating the limitations of the Darknet framework (based on the C programming language and not created with regard to execution in production environments) [38, 39].

The YOLOv5 model is one of the official state-of-the-art models that provides architecture modifications, fine-tuning, testing and deployment on various target platforms. Like the YOLOv4 model, the YOLOv5 model uses cross-stack partial links with Darknet-53 as the foundation and Path Aggregation Network as the neck. Key

enhancements to this model include a new increase in mosaic data and auto-learning bindings to bounding rectangles [38].

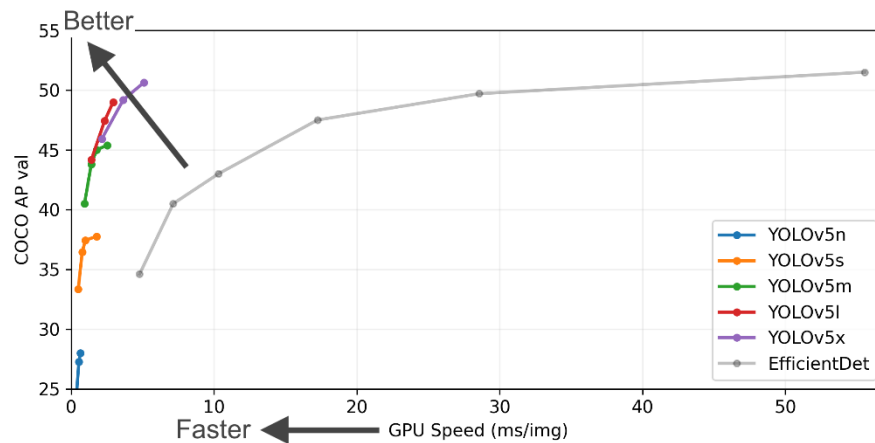


Figure 4 — Comparison on MS COCO dataset of mAP metrics and image processing speed using YOLOv5 model with other CNN models

The YOLOv7 CNN model is the ideological successor of the YOLOv4 model and introduces new features to its architecture: E-ELAN backbone computational block and the BoF (Bag of Freebies) trainable block [40]. Unlike YOLOv4, YOLOv7 is implemented using the PyTorch framework [41]. Its capabilities are shown in Figure 5.

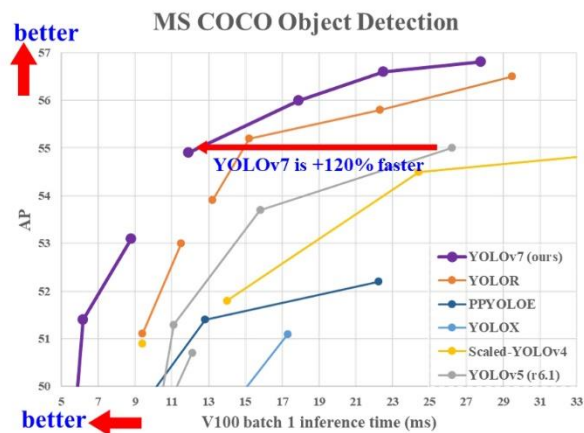


Figure 5 — Comparison on MS COCO dataset of mAP metrics and image processing speed using YOLOv7 model with other CNN models

Usually, models of CNN with a one-stage architecture when detecting objects are calculated at a speed sufficient for the functioning of mobile CV systems, which have a computing device in the form of a modern GPU. However, most of such mod-

els, including many models of YOLO class, although much faster and more efficient than two-stage models, still are not suitable as a basis for creating many mobile CV systems due to severe limitations of memory resources and performance of computing devices of such CV systems.

In this connection, models with reduced sets of convolutional layers, called Tiny YOLO CNN [10], were proposed based on the considered YOLO CNN models. Their calculations impose much lower memory and performance requirements on the CV system computing device. However, studies have shown that the accuracy of such models is lower than the full-size YOLO models [10, 11]. We note that in addition to the Tiny YOLO models, one of the compact architectures of CNN for object detection is the YOLO Nano CNN model [20]. Unfortunately, it shows even lower detection accuracy compared to the Tiny YOLO CNN models.

1.4.3 Compact CNN models from the YOLO class

The analysis of modern YOLO models [41, 35] has shown that a subset of YOLOv4, YOLOv5, and YOLOv7 models from this class are the most promising for reaching a compromise in satisfying the requirements for the CV system from subsection 1.3.

Moreover, in our opinion, the compact (small number of convolutional layers) CNN models: YOLOv4 Tiny, YOLOv5 Small and YOLOv7 Tiny should be investigated and the most efficient of this subset should be identified. First, the compact CNN models do have a smaller number of convolutional layers compared to the full-size models from this subset (Table 2 and Table 3), which directly affects the computational speed of CNN models and their algorithmic efficiency. Table 3 shows that these compact CNN models have very high algorithmic efficiency compared to other models from the analyzed subset of models from the YOLO class.

Table 2 — Basic characteristics of compact models of the YOLO class CNN

Characteristics	YOLOv4 Tiny	YOLOv5 Small	YOLOv7 Tiny
Number of all layers	37	49	78
Number of convolutional layers	21	35	55

Characteristics	YOLOv4 Tiny	YOLOv5 Small	YOLOv7 Tiny
Activation function	Mish	SiLU	LeakyReLU
Optimization function	SGD	SGD	SGD
Loss function	CIoU	BCEWithLogitsLoss	BCEWithLogitsLoss

Second, the CNN models chosen for the study have quite good performance on the AP/mAP metrics, for example, for the MS COCO dataset [43] we have:

- YOLOv4 Tiny CNN — $AP_{0.5} \approx 42.1\%$ [13, 37, 14];
- YOLOv5 Small CNN — $mAP_{0.5} \approx 56.8$ [39];
- YOLOv7 Tiny CNN — $AP_{0.5} \approx 52.8\%$ [41, 14].

Table 3 — Dimensions and computational complexity (number of operations) of YOLO-class CNN models

CNN model	Model size, MB	Number of operations, GFLOPs
YOLOv4 Tiny	23.1	6.9
YOLOv4	245.0	60.1
YOLOv5 Nano	3.9	4.5
YOLOv5 Small	14.1	16.5
YOLOv5 Medium	40.8	49.0
YOLOv7 Tiny	11.6	5.8
YOLOv7	72.0	105.2

1.5 Purpose and objectives of the research

As mentioned above, mobile (transportable or wearable) CV systems based on modern CNN models are in great demand today. The three compact models from the YOLO class: YOLOv4 Tiny, YOLOv5 Small and YOLOv7 Tiny, chosen as a result of the analysis in section 1.4.3, are used as CNN models for mobile CV system designed to detect flying objects on the images. As a result, it is possible to formulate the following goal and objectives of the research of the FQW.

The purpose of this work is to develop and investigate models of the YOLO class (often said family) to detect flying objects of different sizes (scales) on the images and choose as a result of research, including the category of objects by size, the most effective model of CNN of this class to create a mobile real-time CV system.

In order to realize this goal, it is necessary to solve the following tasks:

1. Generate two sets of RGB-images of flying objects of four classes: “Bird (flock of birds)”, “UAV of airplane type”, “UAV of helicopter type” and “Unknown object”. Each image of the first set contains one or more objects of these classes, and the objects will have different sizes (different-scale objects). A part of the second set contains only small-size objects but they belong to any of the four classes, the second, equal to the first, part of the images in the first set contains images with average-size objects of the four classes and the third equal part contains images with large-size objects of any of the four classes. The second set should be formed according to the sizes of each object category described in Subsection 1.3. The volume of images in the second set should be three times greater than in the first set.

2. Mark up and annotate the objects of the mentioned classes on RGB-images of the sets and form as a result four datasets (the first dataset, called the main one, is formed from images of the first set and three additional datasets — from images of the second set: one dataset from images containing objects of each size category) for training, validation and research of developed models of CNN of YOLO class.

3. Develop and train the three compact models of the selected CNN as a result of the analysis, using training and validation samples from the prepared datasets.

4. Study the efficiency of the trained compact models of CNN using test samples of each of the four datasets. Based on the results, identify the most efficient compact CNN model.

5. Based on the most efficient model, develop a new CNN model of the YOLO class using Inception – ResNet modules and feature forwarding mechanism.

6. Train the new CNN model using prepared training and validation samples of each of the three datasets generated from the images of the second set.

7. Study the effectiveness of each variant of the trained new CNN model using a test sample in each of the three data sets.

8. Analyze all the results of comprehensive studies of three compact and new models of CNN and, taking into account the previously formed requirements to

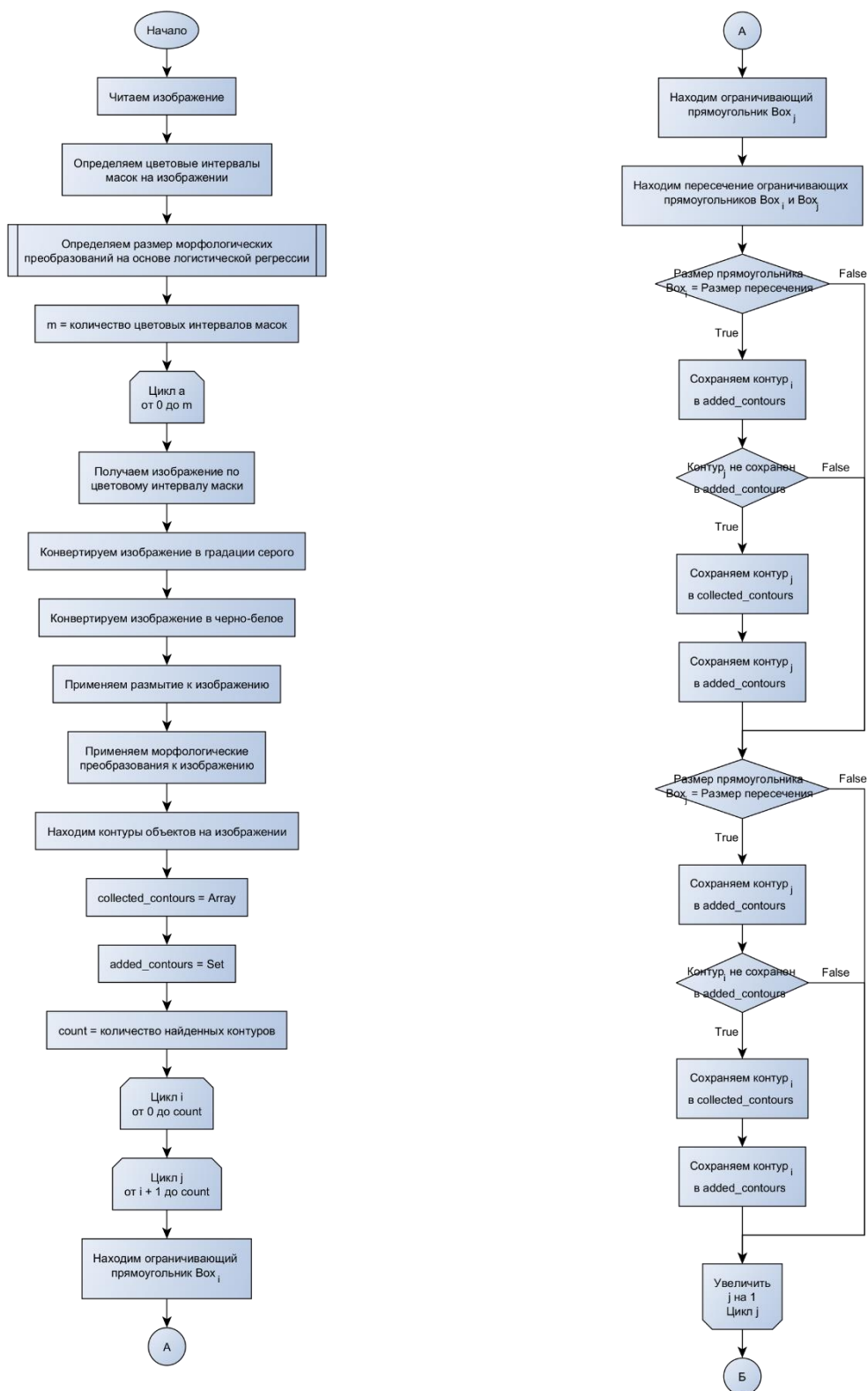
mobile real-time CV system, select the most suitable model of the CNN to be implemented as part of the CV system to detect flying objects of different scales on the images.

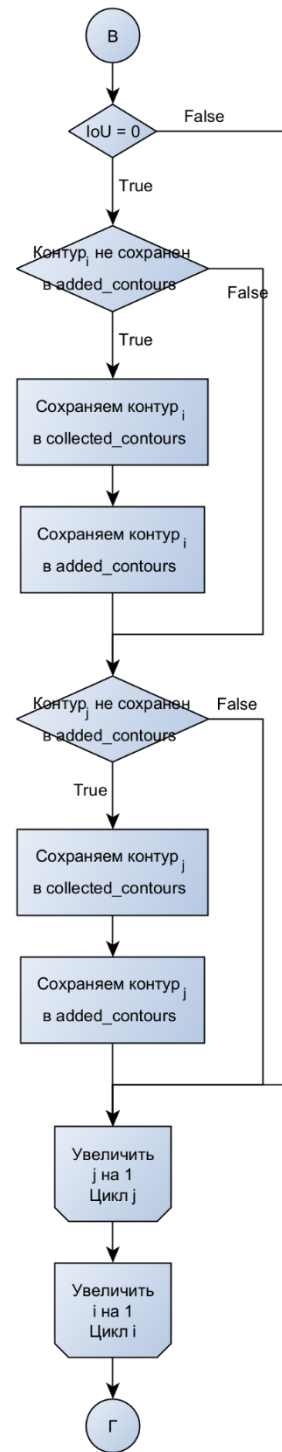
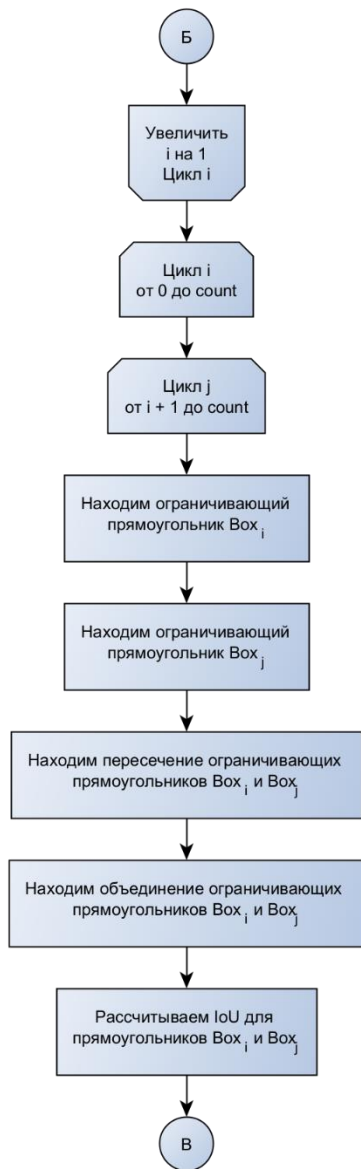
9. Make up recommendations to use of the developed CNN models in mobile computer vision systems.

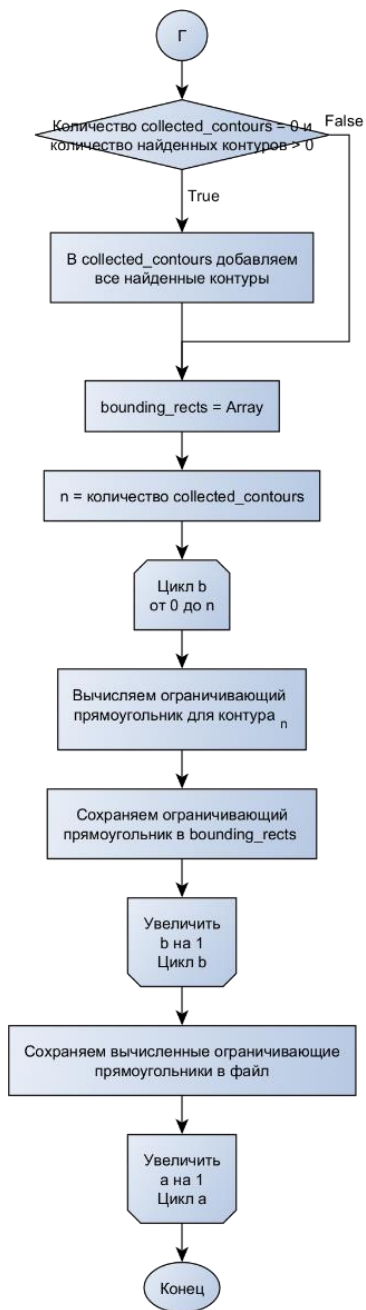
ПРИЛОЖЕНИЕ Б

Схема алгоритма нахождения ограничивающих прямоугольников

(справочное)







ПРИЛОЖЕНИЕ В

Программа для нахождения ограничивающих прямоугольников на основе библиотеки OpenCV

(справочное)

```
# Импортируем библиотеки
import cv2
from shapely.geometry import Polygon
import numpy as np
import pandas as pd
from tqdm import tqdm
from os import listdir, makedirs
from os.path import isfile, join
import pickle
from utils import to_factors

# Загружаем модель логистической регрессии
with open('morph_regression_model.pkl', 'rb') as file:
    morph_regression_model = pickle.load(file)

# Определяем константы
base_dir = 'PATH_TO_BASE_DIR'
data_dir = 'NAME_OF_DATA_DIR'
images_dir = 'images'
labels_dir = 'labels'

images_dirname = join(base_dir, data_dir, images_dir)
labels_dirname = join(base_dir, data_dir, labels_dir)
filenames = [file for file in listdir(labels_dirname) if
isfile(join(labels_dirname, file))]

debug_output = False

colors = {
    'aircraft-type_uav': (255, 0, 0),
    'helicopter-type_uav': (0, 255, 255),
    'bird': (255, 255, 0),
    'unknown': (255, 0, 255)
}

bounds = pd.DataFrame(columns=['filename', 'width', 'height', 'class',
'xmin', 'ymin', 'xmax', 'ymax'])

# Создаём директории
if debug_output:
    makedirs(join('./parsed', data_dir, './images'))

# Составляем координаты масок для каждого изображения
for filename in tqdm(filenames):
```

```

# Читаем изображение
image = cv2.imread(join(labels_dirname, filename))
image_height, image_width, _ = image.shape
if debug_output:
    drawing = image.copy()

# Определяем маски на изображении
masks = {
    'aircraft-type_uav': cv2.inRange(image, (200, 0, 0), (255, 150,
150)),
    'helicopter-type_uav': cv2.inRange(image, (0, 200, 200), (150,
255, 255)),
    'bird': cv2.inRange(image, (200, 200, 0), (255, 255, 150)),
    'unknown': cv2.inRange(image, (200, 0, 200), (255, 150, 255))
}

# Определяем размер размытия на основе логистической регрессии
morph_size = morph_regression_model.predict([to_factors(image_width,
image_height)])[0]

# Для каждой маски определяем объекты на изображении
for entity_type in masks:
    mask = masks[entity_type]

    # Получаем изображение по маске
    mask_image = cv2.bitwise_and(image, image, mask=mask)
    # Конвертируем изображение в серые тона
    grayscale_image = cv2.cvtColor(mask_image, cv2.COLOR_RGB2GRAY)
    # Конвертируем изображение в черно-белое
    _, threshold_image = cv2.threshold(grayscale_image, 0, 255,
cv2.THRESH_BINARY)
    # Размываем границы объектов
    blur_image = cv2.blur(threshold_image, (5, 5))
    # Применение морфологических преобразований
    structuring_element =
cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (morph_size, morph_size))
    morph_image = cv2.morphologyEx(blur_image, cv2.MORPH_CLOSE,
structuring_element)

    # Нахождение контуров на изображении
    contours, _ = cv2.findContours(morph_image, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    count = len(contours)
    collected_contours = []
    added_contours = set()

    # Удаляем лишние контуры, если они являются субконтуром объекта
    for a in range(count):
        for b in range(a + 1, count):

```

```

        (a_x, a_y, a_width, a_height) =
cv2.boundingRect(contours[a])
        (b_x, b_y, b_width, b_height) =
cv2.boundingRect(contours[b])

        a_box = [(a_x, a_y), (a_x + a_width, a_y), (a_x +
a_width, a_y + a_height), (a_x, a_y + a_height)]
        b_box = [(b_x, b_y), (b_x + b_width, b_y), (b_x +
b_width, b_y + b_height), (b_x, b_y + b_height)]

        a_poly = Polygon(a_box)
        b_poly = Polygon(b_box)
        intersection = a_poly.intersection(b_poly)

        if a_poly.area == intersection.area:
            added_contours.add(a)
            if not b in added_contours:
                collected_contours.append(contours[b])
                added_contours.add(b)

        if b_poly.area == intersection.area:
            added_contours.add(b)
            if not a in added_contours:
                collected_contours.append(contours[a])
                added_contours.add(a)

# Добавляем оставшиеся неучтённые контуры
for a in range(count):
    for b in range(a + 1, count):
        (a_x, a_y, a_width, a_height) =
cv2.boundingRect(contours[a])
        (b_x, b_y, b_width, b_height) =
cv2.boundingRect(contours[b])

        a_box = [(a_x, a_y), (a_x + a_width, a_y), (a_x +
a_width, a_y + a_height), (a_x, a_y + a_height)]
        b_box = [(b_x, b_y), (b_x + b_width, b_y), (b_x +
b_width, b_y + b_height), (b_x, b_y + b_height)]

        a_poly = Polygon(a_box)
        b_poly = Polygon(b_box)
        intersection = a_poly.intersection(b_poly)
        union = a_poly.union(b_poly)
        iou = intersection.area / union.area

        if iou == 0:
            if not a in added_contours:
                collected_contours.append(contours[a])
                added_contours.add(a)
            if not b in added_contours:
                collected_contours.append(contours[b])

```

```

        added_contours.add(b)

    # В том случае, если на изображении не было субконтуров,
    используем все найденные
    if len(collected_contours) == 0 and len(contours) > 0:
        collected_contours = contours

    # Собираем границы контуров
    bounding_rects = []
    for contour in collected_contours:
        bounding_rects.append(cv2.boundingRect(contour))

    # Сохраняем найденные границы контуров в датафрейм
    entity_bounds = pd.DataFrame(data=[
        [join(images_dirname, filename), image.shape[1],
        image.shape[0], entity_type, x, y, x + width, y + height] for
        (x, y, width, height) in bounding_rects
    ], columns=bounds.columns)
    bounds = pd.concat([bounds, entity_bounds])

    if debug_output:
        # Отображаем на изображении найденные границы
        shapes = np.zeros_like(drawing, dtype=np.uint8)
        for (x, y, width, height) in bounding_rects:
            cv2.rectangle(shapes, (x, y), (x + width, y + height),
            colors[entity_type], cv2.FILLED)

            alpha = 0.85
            mask_shapes = shapes.astype(bool)
            drawing[mask_shapes] = cv2.addWeighted(drawing, alpha,
            shapes, 1 - alpha, 0)[mask_shapes]

            for (x, y, width, height) in bounding_rects:
                cv2.rectangle(drawing, (x, y), (x + width, y + height),
                colors[entity_type], 1)

        if debug_output:
            cv2.imwrite(join('./parsed', data_dir, 'images', filename),
            drawing)

    bounds = bounds.reset_index(drop=True)

    # Сохраняем датафрейм
    if debug_output:
        bounds.to_csv(join('./parsed', data_dir, 'data.csv'), index=False)
    else:
        bounds.to_csv(f'./store/{data_dir}.csv', index=False)

```

ПРИЛОЖЕНИЕ Г

Программа обучения логистической регрессии для формирования размера структурирующего элемента морфологических преобразований

(справочное)

```
import numpy as np
from sklearn.linear_model import LogisticRegression
import pickle

def to_factors(width, height):
    area = width * height

    return [
        width,
        height,
        area,
        width / height,
        height / width
    ]

model_name = 'morph_regression_model.pkl'

X_train = np.array([
    to_factors(4096, 4096),
    to_factors(2560, 1622),
    to_factors(2048, 1266),
    to_factors(2048, 2048),
    to_factors(775, 518),
    to_factors(512, 512)
])

y_train = np.array([3, 3, 6, 6, 12, 12])

model = LogisticRegression()
model.fit(X_train, y_train)

with open(model_name, 'wb') as file:
    pickle.dump(model, file)
```

ПРИЛОЖЕНИЕ Д

Программа для формирования основного датасета

(справочное)

```
from sklearn.model_selection import train_test_split
from tqdm import tqdm
from glob import glob
from pathlib import Path
from os import makedirs
from os.path import join, basename, splitext
from shutil import copy
from collections import defaultdict, Counter

seed = 64

input_dirname = './cleaned'
output_dirname = './dataset'

def map_filenames(entries):
    mapped = {}
    for entry in entries:
        key = splitext(basename(entry))[0]
        mapped[key] = entry
    return mapped

image_filenames = map_filenames(glob(f'{input_dirname}/images/*'))
label_filenames = map_filenames(glob(f'{input_dirname}/labels/*'))

files = { key: { 'image': image_filenames[key], 'labels':
label_filenames[key] } for key in image_filenames }

classes_map = {
    '0': 'bird',
    '1': 'helicopter-type_uav',
    '2': 'aircraft-type_uav',
    '3': 'unknown'
}

classes = defaultdict(set)

for key in tqdm(files.keys()):
    with open(files[key]['labels'], 'r') as file:
        defined_classes = set([line.split(' ', maxsplit=1)[0] for line in
file.readlines()])
```



```

    for defined_class in defined_classes:
        classes[defined_class].add(key)

train_keys = []
val_keys = []
test_keys = []

for class_id in classes.keys():
    keys = list(classes[class_id])
    local_train_keys, local_val_keys = train_test_split(keys,
test_size=0.2, random_state=seed)
    local_val_keys, local_test_keys = train_test_split(local_val_keys,
test_size=0.5, random_state=seed)

    train_keys.extend(local_train_keys)
    val_keys.extend(local_val_keys)
    test_keys.extend(local_test_keys)

counts = Counter()

def copy_entry(original_path, new_dirname, new_basename):
    extension = splitext(original_path)[-1]
    new_path = join(new_dirname, f'{new_basename}{extension}')
    copy(original_path, new_path)

def move_entries(items, name):
    images_dir = join(output_dirname, 'images', name)
    labels_dir = join(output_dirname, 'labels', name)

    makedirs(images_dir, exist_ok=True)
    makedirs(labels_dir, exist_ok=True)

    for item in tqdm(items, desc=name):
        index = counts[name]

        copy_entry(files[item]['image'], images_dir, index)
        copy_entry(files[item]['labels'], labels_dir, index)

        counts[name] += 1

move_entries(train_keys, 'train')
move_entries(val_keys, 'val')
move_entries(test_keys, 'test')

```

ПРИЛОЖЕНИЕ Е

Программа для формирования датасетов по размерам объектов

(справочное)

```
import cv2
import numpy as np
import json
from sklearn.model_selection import train_test_split
from tqdm import tqdm
from math import inf
from glob import glob
from hashlib import sha256
from pathlib import Path
import pickle
from os import makedirs
from os.path import join, basename, splitext, exists
from shutil import copy
from collections import defaultdict, Counter

seed = 64

input_dirname = './cleaned'
output_dirname = './dataset'

classes_map = {
    '0': 'bird',
    '1': 'helicopter-type_uav',
    '2': 'aircraft-type_uav',
    '3': 'unknown'
}

sizes_map = {
    'small': [(0, 0), (32, 32)],
    'medium': [(32, 32), (96, 96)],
    'big': [(96, 96), (inf, inf)]
}

computed_key = sha256(
    json.dumps({
        'input_dirname': input_dirname,
        'output_dirname': output_dirname,
        'classes_map': classes_map,
        'sizes_map': sizes_map
    }).encode('utf-8')
).hexdigest()
computed_filename = join('./computed', f'{computed_key}.pkl')
```

```

def map_filenames(entries):
    mapped = {}
    for entry in entries:
        key = splitext(basename(entry))[0]
        mapped[key] = entry
    return mapped

image_filenames = map_filenames(glob(f'{input_dirname}/images/*'))
label_filenames = map_filenames(glob(f'{input_dirname}/labels/*'))

files = { key: { 'image': image_filenames[key], 'labels':
label_filenames[key] } for key in image_filenames }

def dict_unwrap(entry):
    if isinstance(entry, defaultdict):
        return {
            key: dict_unwrap(value) for (key, value) in entry.items()
        }
    return entry

if exists(computed_filename):
    with open(computed_filename, 'rb') as file:
        classes = pickle.load(file)
else:
    classes = defaultdict(lambda: defaultdict(lambda: list()))

    for key in tqdm(files.keys()):
        image = cv2.imdecode(np.fromfile(files[key]['image'], np.uint8),
cv2.IMREAD_UNCHANGED)
        height, width, _ = image.shape

        with open(files[key]['labels'], 'r') as file:
            for line in file.readlines():
                class_id, params = line.split(' ', maxsplit=1)
                center_x, center_y, size_x, size_y = map(float,
params.split(' '))
                box_width = size_x * width
                box_height = size_y * height

                for size in sizes_map:
                    (box_min_width, box_min_height), (box_max_width,
box_max_height) = sizes_map[size]
                    if box_min_width <= box_width < box_max_width and
box_min_height <= box_height < box_max_height:
                        classes[size][key].append(line)
                        break

```

```

with open(computed_filename, 'wb') as file:
    pickle.dump(dict_unwrap(classes), file)

def create_entry(content, original_path, new_dirname, new_basename):
    extension = splitext(original_path)[-1]
    new_path = join(new_dirname, f'{new_basename}{extension}')

    with open(new_path, 'w') as file:
        file.writelines([line.strip() for line in content])

def copy_entry(original_path, new_dirname, new_basename):
    extension = splitext(original_path)[-1]
    new_path = join(new_dirname, f'{new_basename}{extension}')
    copy(original_path, new_path)

for size in classes:
    train_set = defaultdict(list)
    val_set = defaultdict(list)
    test_set = defaultdict(list)

    for class_id in classes[size].keys():
        keys = list(classes[size][class_id])
        train_keys, val_keys = train_test_split(keys, test_size=0.2,
random_state=seed)
        val_keys, test_keys = train_test_split(val_keys, test_size=0.5,
random_state=seed)

        for key in train_keys:
            train_set[key].extend(classes[size][class_id][key])
        for key in val_keys:
            val_set[key].extend(classes[size][class_id][key])
        for key in test_keys:
            test_set[key].extend(classes[size][class_id][key])

counts = Counter()

def move_entries(items, name):
    images_dir = join(output_dirname, size, 'images', name)
    labels_dir = join(output_dirname, size, 'labels', name)

    makedirs(images_dir, exist_ok=True)
    makedirs(labels_dir, exist_ok=True)

    for (item, labels) in tqdm(items.items(), desc=f'{size} {name}'):
        index = counts[name]

        copy_entry(files[item]['image'], images_dir, index)

```

```
index)         create_entry(labels, files[item]['labels'], labels_dir,  
  
              counts[name] += 1  
  
move_entries(train_set, 'train')  
move_entries(val_set, 'val')  
move_entries(test_set, 'test')
```

ПРИЛОЖЕНИЕ Ж

Архитектура модели СНС YOLOv4 Tiny

(справочное)

layer	filters	size/strd(dil)	input	output
0 conv	32	3 x 3/ 2	416 x 416 x 3 ->	208 x 208 x 32 0.075 BF
1 conv	64	3 x 3/ 2	208 x 208 x 32 ->	104 x 104 x 64 0.399 BF
2 conv	64	3 x 3/ 1	104 x 104 x 64 ->	104 x 104 x 64 0.797 BF
3 route	2		1/2 ->	104 x 104 x 32
4 conv	32	3 x 3/ 1	104 x 104 x 32 ->	104 x 104 x 32 0.199 BF
5 conv	32	3 x 3/ 1	104 x 104 x 32 ->	104 x 104 x 32 0.199 BF
6 route	5 4		->	104 x 104 x 64
7 conv	64	1 x 1/ 1	104 x 104 x 64 ->	104 x 104 x 64 0.089 BF
8 route	2 7		->	104 x 104 x 128
9 max		2x 2/ 2	104 x 104 x 128 ->	52 x 52 x 128 0.001 BF
10 conv	128	3 x 3/ 1	52 x 52 x 128 ->	52 x 52 x 128 0.797 BF
11 route	10		1/2 ->	52 x 52 x 64
12 conv	64	3 x 3/ 1	52 x 52 x 64 ->	52 x 52 x 64 0.199 BF
13 conv	64	3 x 3/ 1	52 x 52 x 64 ->	52 x 52 x 64 0.199 BF
14 route	13 12		->	52 x 52 x 128
15 conv	128	1 x 1/ 1	52 x 52 x 128 ->	52 x 52 x 128 0.089 BF
16 route	10 15		->	52 x 52 x 256
17 max		2x 2/ 2	52 x 52 x 256 ->	26 x 26 x 256 0.001 BF
18 conv	256	3 x 3/ 1	26 x 26 x 256 ->	26 x 26 x 256 0.797 BF
19 route	18		1/2 ->	26 x 26 x 128
20 conv	128	3 x 3/ 1	26 x 26 x 128 ->	26 x 26 x 128 0.199 BF
21 conv	128	3 x 3/ 1	26 x 26 x 128 ->	26 x 26 x 128 0.199 BF
22 route	21 20		->	26 x 26 x 256
23 conv	256	1 x 1/ 1	26 x 26 x 256 ->	26 x 26 x 256 0.089 BF
24 route	18 23		->	26 x 26 x 512
25 max		2x 2/ 2	26 x 26 x 512 ->	13 x 13 x 512 0.000 BF
26 conv	512	3 x 3/ 1	13 x 13 x 512 ->	13 x 13 x 512 0.797 BF
27 conv	256	1 x 1/ 1	13 x 13 x 512 ->	13 x 13 x 256 0.044 BF
28 conv	512	3 x 3/ 1	13 x 13 x 256 ->	13 x 13 x 512 0.399 BF
29 conv	27	1 x 1/ 1	13 x 13 x 512 ->	13 x 13 x 27 0.005 BF
30 yolo				
31 route	27		->	13 x 13 x 256
32 conv	128	1 x 1/ 1	13 x 13 x 256 ->	13 x 13 x 128 0.011 BF
33 upsample		2x	13 x 13 x 128 ->	26 x 26 x 128
34 route	33 23		->	26 x 26 x 384
35 conv	256	3 x 3/ 1	26 x 26 x 384 ->	26 x 26 x 256 1.196 BF
36 conv	27	1 x 1/ 1	26 x 26 x 256 ->	26 x 26 x 27 0.009 BF
37 yolo				

ПРИЛОЖЕНИЕ И

Параметры модели СНС YOLOv4 Tiny

(справочное)

```
[net]
# Testing
#batch=1
#subdivisions=1
# Training
batch=64
subdivisions=8
width=416
height=416
channels=3
momentum=0.949
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches=16000
policy=steps
steps=12800,14400
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=2
pad=1
activation=mish

[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=mish

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=mish

[route]
layers=-1
groups=2
group_id=1

[convolutional]
batch_normalize=1
```

```
filters=32
size=3
stride=1
pad=1
activation=mish

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=mish

[route]
layers = -1,-2

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=mish

[route]
layers = -6,-1

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=mish

[route]
layers=-1
groups=2
group_id=1

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=mish

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=mish

[route]
```



```
layers = -1,-2

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=mish

[route]
layers = -6,-1

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=mish

[route]
layers=-1
groups=2
group_id=1

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=mish

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=mish

[route]
layers = -1,-2

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=mish

[route]
layers = -6,-1

[maxpool]
size=2
```

```

stride=2

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=mish

#####

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=mish

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=mish

[convolutional]
size=1
stride=1
pad=1
filters=27
activation=linear

[yolo]
mask = 3,4,5
anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
classes=4
num=6
jitter=.3
scale_x_y = 1.05
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
ignore_thresh = .7
truth_thresh = 1
random=0
resize=1.5
nms_kind=greedynms
beta_nms=0.6

[route]
layers = -4

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1

```

```
pad=1
activation=mish

[upsample]
stride=2

[route]
layers = -1, 23

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=mish

[convolutional]
size=1
stride=1
pad=1
filters=27
activation=linear

[yolo]
mask = 0,1,2
anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
classes=4
num=6
jitter=.3
scale_x_y = 1.05
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
ignore_thresh = .7
truth_thresh = 1
random=0
resize=1.5
nms_kind=greedynms
beta_nms=0.6
```

ПРИЛОЖЕНИЕ К

Архитектура модели CHC YOLOv5 Small

(справочное)

	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2,
'nearest']					
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	361984	models.common.C3	[512, 256, 1,
False]					
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2,
'nearest']					
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90880	models.common.C3	[256, 128, 1,
False]					
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	296448	models.common.C3	[256, 256, 1,
False]					
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C3	[512, 512, 1,
False]					
24	[17, 20, 23]	1	24273	models.yolo.Detect	[4, [[10, 13, 16,
30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256,					
512]]					

ПРИЛОЖЕНИЕ Л

Параметры модели СНС YOLOv5 Small

(справочное)

lr0: 0.01
lrf: 0.01
momentum: 0.937
weight_decay: 0.0005
warmup_epochs: 3.0
warmup_momentum: 0.8
warmup_bias_lr: 0.1
box: 0.05
cls: 0.5
cls_pw: 1.0
obj: 1.0
obj_pw: 1.0
iou_t: 0.2
anchor_t: 4.0
fl_gamma: 0.0
hsv_h: 0.015
hsv_s: 0.7
hsv_v: 0.4
degrees: 0.0
translate: 0.1
scale: 0.5
shear: 0.0
perspective: 0.0
flipud: 0.0
fliplr: 0.5
mosaic: 1.0
mixup: 0.0
copy_paste: 0.0

ПРИЛОЖЕНИЕ М

Архитектура модели СНС YOLOv7 Tiny

(справочное)

	from	n	params	module	arguments
0		-1	1	928 models.common.Conv	[3, 32, 3, 2,
None, 1, LeakyReLU(negative_slope=0.1)]					
1		-1	1	18560 models.common.Conv	[32, 64, 3, 2,
None, 1, LeakyReLU(negative_slope=0.1)]					
2		-1	1	2112 models.common.Conv	[64, 32, 1, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
3		-2	1	2112 models.common.Conv	[64, 32, 1, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
4		-1	1	9280 models.common.Conv	[32, 32, 3, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
5		-1	1	9280 models.common.Conv	[32, 32, 3, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
6	[-1, -2, -3, -4]	1	0	models.common.Concat	[1]
7		-1	1	8320 models.common.Conv	[128, 64, 1, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
8		-1	1	0 models.common.MP	[]
9		-1	1	4224 models.common.Conv	[64, 64, 1, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
10		-2	1	4224 models.common.Conv	[64, 64, 1, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
11		-1	1	36992 models.common.Conv	[64, 64, 3, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
12		-1	1	36992 models.common.Conv	[64, 64, 3, 1,
None, 1, LeakyReLU(negative_slope=0.1)]					
13	[-1, -2, -3, -4]	1	0	models.common.Concat	[1]
14		-1	1	33024 models.common.Conv	[256, 128, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
15		-1	1	0 models.common.MP	[]
16		-1	1	16640 models.common.Conv	[128, 128, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
17		-2	1	16640 models.common.Conv	[128, 128, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
18		-1	1	147712 models.common.Conv	[128, 128, 3,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
19		-1	1	147712 models.common.Conv	[128, 128, 3,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
20	[-1, -2, -3, -4]	1	0	models.common.Concat	[1]
21		-1	1	131584 models.common.Conv	[512, 256, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
22		-1	1	0 models.common.MP	[]
23		-1	1	66048 models.common.Conv	[256, 256, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
24		-2	1	66048 models.common.Conv	[256, 256, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
25		-1	1	590336 models.common.Conv	[256, 256, 3,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
26		-1	1	590336 models.common.Conv	[256, 256, 3,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
27	[-1, -2, -3, -4]	1	0	models.common.Concat	[1]
28		-1	1	525312 models.common.Conv	[1024, 512, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					
29		-1	1	131584 models.common.Conv	[512, 256, 1,
1, None, 1, LeakyReLU(negative_slope=0.1)]					

30	-2 1 131584	models.common.Conv	[512, 256, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
31	-1 1 0	models.common.SP	[5]
32	-2 1 0	models.common.SP	[9]
33	-3 1 0	models.common.SP	[13]
34	[-1, -2, -3, -4] 1 0	models.common.Concat	[1]
35	-1 1 262656	models.common.Conv	[1024, 256, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
36	[-1, -7] 1 0	models.common.Concat	[1]
37	-1 1 131584	models.common.Conv	[512, 256, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
38	-1 1 33024	models.common.Conv	[256, 128, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
39	-1 1 0	torch.nn.modules.upsampling.Upsample	[None, 2,
'nearest']			
40	21 1 33024	models.common.Conv	[256, 128, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
41	[-1, -2] 1 0	models.common.Concat	[1]
42	-1 1 16512	models.common.Conv	[256, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
43	-2 1 16512	models.common.Conv	[256, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
44	-1 1 36992	models.common.Conv	[64, 64, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
45	-1 1 36992	models.common.Conv	[64, 64, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
46	[-1, -2, -3, -4] 1 0	models.common.Concat	[1]
47	-1 1 33024	models.common.Conv	[256, 128, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
48	-1 1 8320	models.common.Conv	[128, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
49	-1 1 0	torch.nn.modules.upsampling.Upsample	[None, 2,
'nearest']			
50	14 1 8320	models.common.Conv	[128, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
51	[-1, -2] 1 0	models.common.Concat	[1]
52	-1 1 4160	models.common.Conv	[128, 32, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
53	-2 1 4160	models.common.Conv	[128, 32, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
54	-1 1 9280	models.common.Conv	[32, 32, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
55	-1 1 9280	models.common.Conv	[32, 32, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
56	[-1, -2, -3, -4] 1 0	models.common.Concat	[1]
57	-1 1 8320	models.common.Conv	[128, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
58	-1 1 73984	models.common.Conv	[64, 128, 3, 2,
None, 1,	LeakyReLU(negative_slope=0.1)]		
59	[-1, 47] 1 0	models.common.Concat	[1]
60	-1 1 16512	models.common.Conv	[256, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
61	-2 1 16512	models.common.Conv	[256, 64, 1, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
62	-1 1 36992	models.common.Conv	[64, 64, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
63	-1 1 36992	models.common.Conv	[64, 64, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
64	[-1, -2, -3, -4] 1 0	models.common.Concat	[1]
65	-1 1 33024	models.common.Conv	[256, 128, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		

66	-1 1 295424	models.common.Conv	[128, 256, 3,
2, None, 1,	LeakyReLU(negative_slope=0.1)]		
67	[-1, 37] 1 0	models.common.Concat	[1]
68	-1 1 65792	models.common.Conv	[512, 128, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
69	-2 1 65792	models.common.Conv	[512, 128, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
70	-1 1 147712	models.common.Conv	[128, 128, 3,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
71	-1 1 147712	models.common.Conv	[128, 128, 3,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
72	[-1, -2, -3, -4] 1 0	models.common.Concat	[1]
73	-1 1 131584	models.common.Conv	[512, 256, 1,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
74	57 1 73984	models.common.Conv	[64, 128, 3, 1,
None, 1,	LeakyReLU(negative_slope=0.1)]		
75	65 1 295424	models.common.Conv	[128, 256, 3,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
76	73 1 180672	models.common.Conv	[256, 512, 3,
1, None, 1,	LeakyReLU(negative_slope=0.1)]		
77	[74, 75, 76] 1 25250	models.yolo.IDetect	[4, [[10, 13,
16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256,			512

ПРИЛОЖЕНИЕ Н

Параметры модели СНС YOLOv7 Tiny

(справочное)

```
lr0: 0.01
lrf: 0.01
momentum: 0.937
weight_decay: 0.0005
warmup_epochs: 3.0
warmup_momentum: 0.8
warmup_bias_lr: 0.1
box: 0.05
cls: 0.5
cls_pw: 1.0
obj: 1.0
obj_pw: 1.0
iou_t: 0.2
anchor_t: 4.0
fl_gamma: 0.0
hsv_h: 0.015
hsv_s: 0.7
hsv_v: 0.4
degrees: 0.0
translate: 0.1
scale: 0.5
shear: 0.0
perspective: 0.0
flipud: 0.0
fliplr: 0.5
mosaic: 1.0
mixup: 0.05
copy_paste: 0.0
paste_in: 0.05
loss_ota: 1
```

ПРИЛОЖЕНИЕ П

Акт о внедрении

(справочное)

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ОБЪЕДИНЕНИЕ «АЛМАЗ»

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ЛЭМЗ-Т»

(ООО «ЛЭМЗ-Т»)

634055, Томская область, город Томск, проспект Развития, 8
Тел. +7 (3822) 488-527, Факс +7 (3822) 488-526, 488-528
E-mail: info@lemz-t.ru www.lemz-t.ru
ОГРН 1137017022570, ИНН 7017341160, КПП 701701001



АКТ О ВНЕДРЕНИИ

результатов выпускной квалификационной работы Мунько А.С.

Настоящим актом подтверждаем, что в ООО «ЛЭМЗ-Т», являющемся Центром исследований и разработок ПАО «НПО «Алмаз» имени академика А. А. Расплетина, внедрены следующие результаты, полученные студентом группы 8 ВМИИ Томского Политехнического Университета Мунько Александром Сергеевичем при выполнении выпускной квалификационной работы (магистерской диссертации) «Детектирование летающих объектов с помощью сверточных нейронных сетей класса YOLO».

1. Алгоритмы и программы для создания датасетов с целью обучения, валидации и исследования моделей сверточных нейронных сетей (СНС) класса YOLO.
2. Четыре разработанных датасета на основе наборов изображений летающих объектов четырех классов и трех категорий по размерам.
3. Разработанные, программно – реализованные и обученные три компактные модели СНС YOLOv4 Tiny, YOLOv5 Small и YOLOv7 Tiny, входящие в класс моделей СНС YOLO.
4. Разработанная, программно – реализованная и обученная новая модель СНС, основанная на наиболее эффективной из исследуемых моделей СНС компактной модели YOLOv5 Small.
5. Результаты комплексных исследований эффективности трех разработанных и обученных компактных моделей СНС и разработанной и обученной новой модели СНС.
6. Рекомендации по использованию разработанных моделей СНС в мобильных системах компьютерного зрения.

Эти результаты были использованы при выполнении НИР «АСКУ», цель которой - создание мобильных систем компьютерного зрения на основе современных моделей СНС.

Начальник сектора 147

Ведущий инженер сектора 143

 Е.В. Григорьев

 М. А. Гураков

ПРИЛОЖЕНИЕ Р

Диплом III степени за доклад

(справочное)

ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ



ДИПЛОМ
III степени

ВРУЧАЕТСЯ

МУНЬКО АЛЕКСАНДРУ СЕРГЕЕВИЧУ

за доклад, представленный на

XX Международной научно-практической конференции
студентов, аспирантов и молодых ученых
«Молодёжь и современные информационные технологии»

и оцененный в **94** балла,

**«ДЕТЕКТИРОВАНИЕ ЛЕТАЮЩИХ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ С ПОМОЩЬЮ
СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ СЕМЕЙСТВА YOLO»**

20-22 марта 2023 г.

Проректор по цифровизации
директор ИШИТР



А.С. Фадеев