

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«___» _____ 2024 г.

Моделирование поискового беспилотного летательного аппарата

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2024.406 ПЗ ВКР

Руководитель работы,
к.п.н., доцент каф. ЭВМ
_____ Ю.Г. Плаксина
«___» _____ 2024 г.

Автор работы,
студент группы КЭ-406
_____ А.А. Поляков
«___» _____ 2024 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
«___» _____ 2024 г.

Челябинск-2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«___» _____ 2023 г.

ЗАДАНИЕ
на выпускную квалификационную работу бакалавра
студенту группы КЭ-406
Полякову Андрею Алексеевичу
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

Тема работы: «Моделирование поискового беспилотного летательного аппарата»

Срок сдачи студентом законченной работы: 1 июня 2024 г.

Исходные данные к работе: длительность полета не менее 10 минут;
видеоизображение передается оператору в качестве 1080p;
дальность полета не менее 5км.

Перечень подлежащих разработке вопросов:

- 1) аналитический обзор научно-технической, нормативной и методической литературы по тематике работы;
- 2) формулирование требований, выбор среды и средств разработки, подбор компонентов;

- 3) сборка беспилотного летательного аппарата;
- 4) разработка программного кода летательного аппарата.

Дата выдачи задания: 1 декабря 2023 г.

Руководитель работы _____ / Ю.Г. Плаксина/

Студент _____ / А.А. Поляков/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы	01.02.2024	
Формулирование требований, выбор среды и средств разработки, подбор компонентов	03.03.2024	
Сборка беспилотного летательного аппарата	07.05.2024	
Разработка программного кода летательного аппарата	15.05.2024	
Компоновка текста работы и сдача на нормоконтроль	25.05.2024	
Подготовка презентации и доклада	30.05.2024	

Руководитель работы _____ / Ю.Г. Плаксина /

Студент _____ / А.А. Поляков /

АННОТАЦИЯ

Поляков А. А. Моделирование поискового беспилотного летательного аппарата – Челябинск: ЮУрГУ, ВШЭКН; 2024, 59 с., 10 ил., библиогр. список – 25 наим.

В данной выпускной квалификационной работе рассмотрены актуальные методы и технологии создания беспилотных летательных аппаратов (БПЛА) для проведения поисково-спасательных операций. Работа включает в себя аналитический обзор современных достижений в области беспилотных технологий, выбор и обоснование комплектующих для сборки БПЛА, а также разработку программного обеспечения для управления дроном и распознавания лиц.

Были проанализированы различные типы силовых установок, полетных контроллеров и других ключевых компонентов дронов, что позволило сформировать оптимальную конфигурацию для выполнения поставленных задач. Основное внимание уделено выбору полетного контроллера SucceX-E F4 с прошивкой BETAFLIGHTF4, моторам XING-E 2207 2750KV, пропеллерам IFLIGHT NAZGUL 5140, приемнику FrSky XM+ или R-XSR, и камере Caddx Ratel.

В работе подробно описан процесс сборки и настройки БПЛА, включая установку и подключение всех компонентов. Описаны алгоритмы обнаружения и распознавания лиц с использованием классификатора каскадов Хаара, метода LBP и метода HOG. Рассмотрены преимущества и недостатки каждого метода, что позволило выбрать наиболее подходящий алгоритм для реализации в проекте.

Итогом работы является создание высокоэффективного беспилотного летательного аппарата, способного выполнять сложные поисковые задачи с высокой степенью надежности и точности. Работа открывает новые перспективы для применения БПЛА в различных сферах и способствует дальнейшему развитию беспилотных технологий.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. АНАЛИТИЧЕСКИЙ ОБЗОР СОВРЕМЕННОЙ НАУЧНО-ТЕХНИЧЕСКОЙ, НОРМАТИВНОЙ, МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ, ЗАТРАГИВАЮЩЕЙ НАУЧНО-ТЕХНИЧЕСКУЮ ПРОБЛЕМУ, ИССЛЕДУЕМУЮ В РАБОТЕ.....	5
Вывод по разделу один	11
2. ФОРМУЛИРОВАНИЕ ТРЕБОВАНИЙ, ВЫБОР СРЕДЫ И СРЕДСТВ РАЗРАБОТКИ, ПОДБОР КОМПОНЕНТОВ.....	13
2.1. Формулирование требований	13
2.1.1. Функциональные требования.....	13
2.1.2. Нефункциональные требования.....	13
2.2. Выбор среды и средств разработки	14
2.2.1. Операционная система	14
2.2.2. Среда разработки	16
2.2.3. Система контроля версий.....	17
2.2.4. Языки программирования и библиотеки.....	18
2.2.5. Выбор полетного контроллера.....	19
2.2.6. Подбор компонентов	21
Вывод по разделу два.....	24
3. СБОРКА БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА.....	27
3.1. Перечень комплектующих	27
3.1. Установка компонентов	29
Вывод по разделу три.....	37
4. РАЗРАБОТКА ПРОГРАММНОГО КОДА ЛЕТАТЕЛЬНОГО АППАРАТА	38
4.1. Выбор алгоритма распознавания лиц.....	38

4.1.1. Классификатор каскадов Хаара	38
4.1.2. Классификатор LBP (Local Binary Patterns)	40
4.1.3. Метод HOG (Histogram of Oriented Gradients)	42
4.1.4. Сверточные нейронные сети (CNN)	44
4.1.5. Разработка алгоритма создания датасета	45
4.1.6. Разработка алгоритма распознавания.....	47
Вывод по разделу четыре.....	51
ЗАКЛЮЧЕНИЕ	52
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	54
ПРИЛОЖЕНИЯ.....	57
Приложение А	57
Приложение Б.....	58
Приложение В	60

ВВЕДЕНИЕ

В современном мире технологии развиваются очень быстро, и одним из самых перспективных направлений является использование беспилотных летательных аппаратов (БПЛА). Эти автономные устройства способны выполнять разнообразные задачи без непосредственного участия человека в таких областях, как сельское хозяйство, строительство, логистика, безопасность и многие другие.

Особую значимость БПЛА играют в деятельности МЧС России. Они могут заменить самолеты и вертолеты при выполнении задач, сопряженных с риском для жизни экипажей и возможной утратой дорогостоящей пилотируемой авиационной техники. Первые БПЛА поступили на вооружение МЧС России в 2009 году, и уже летом 2010 года они использовались для мониторинга пожарной обстановки в Московской области, в частности, в Шатурском и Егорьевском районах. Это подтверждает их эффективность в работе ведомства.

Беспилотные летательные аппараты имеют ряд преимуществ перед пилотируемыми воздушными судами. Они позволяют свести риски для жизни людей к минимуму и сохранить дорогостоящую технику. Кроме того, БПЛА могут выполнять задачи в условиях, представляющих опасность для человека. Таким образом, использование беспилотных летательных аппаратов в МЧС России является актуальным и перспективным направлением, позволяющим повысить эффективность работы ведомства и обеспечить безопасность сотрудников [1].

Беспилотные летательные аппараты (БПЛА) все большее применяются в логистике, включая инвентаризацию товаров на складах. Оснащенные камерами и датчиками, они способны сканировать штрихкоды и передавать информацию о товарах в систему управления складом. Это позволяет быстро и точно отслеживать наличие и перемещение товаров.

Кроме того, БПЛА могут использоваться для внутренней доставки на складе. Они способны перевозить небольшие грузы, такие как документы, инструменты или запчасти, между различными зонами, что ускоряет процесс и снижает риск ошибок и потерь.

Также БПЛА могут выполнять мониторинг состояния склада. Оборудованные датчиками температуры, влажности и других параметров, они помогают контролировать условия хранения и предотвращать порчу товаров.

В результате, использование БПЛА на складах способствует автоматизации процессов, повышению эффективности работы, снижению затрат и улучшению качества обслуживания клиентов [2].

С помощью БПЛА можно выполнять широкий спектр задач, связанных со строительством. Они используются для поиска несанкционированных свалок или выявления незаконной застройки, контроля качества дорожного покрытия, контроля точности монтажа строительных конструкций и других задач. Оснащённые датчиками и камерами БПЛА позволяют проводить различные измерения и исследования. С их помощью проводятся замеры уровня шума, радиации, инсоляции (для выбора оптимального размещения солнечных панелей) и другие исследования [3].

Беспилотные летательные аппараты (БПЛА) могут использоваться для опыления сельскохозяйственных культур, в том числе картофеля. Это позволяет повысить эффективность процесса опыления и снизить затраты на использование традиционных методов. Принцип работы БПЛА для опыления заключается в том, что аппарат, оснащённый системой распыления, пролетает над полем и равномерно распределяет необходимые вещества. Дрон при высоте полёта 9 м, имея ширину захвата опрыскивания 10,4 м, опрыскивает поле площадью 120×20 м с высотой растений картофеля до 1,0 м за 15 мин включая дозаправку. Такой подход может быть особенно полезен в труднодоступных местах или на больших площадях, где использование ручного труда может быть затруднительным. Однако стоит учесть, что применение БПЛА для опыления требует тщательного планирования и соблюдения мер безопасности, чтобы избежать нежелательного воздействия на окружающую среду и здоровье людей [4].

1. АНАЛИТИЧЕСКИЙ ОБЗОР СОВРЕМЕННОЙ НАУЧНО-ТЕХНИЧЕСКОЙ, НОРМАТИВНОЙ, МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ, ЗАТРАГИВАЮЩЕЙ НАУЧНО-ТЕХНИЧЕСКУЮ ПРОБЛЕМУ, ИССЛЕДУЕМУЮ В РАБОТЕ

Дроны или беспилотные летательные аппараты (БПЛА) предоставляют уникальную возможность получать актуальную информацию о состоянии водных объектов, их качестве и объеме. Они позволяют своевременно принимать меры по предотвращению загрязнения и улучшению экологической ситуации. Отслеживание качества воды, контроль за распределением ресурсов, а также предупреждение об экологических катастрофах – все это становится реальностью благодаря применению дронов на водных областях. Важность этого подхода выходит за рамки научных исследований, простираясь до обеспечения устойчивого использования водных ресурсов для будущих поколений [5].

Также в статье [6] мы можем увидеть, что изучение атмосферы Арктики с помощью беспилотных летательных аппаратов имеет большое значение для составления надёжного прогноза погоды. Это особенно важно для участников работ, которые проводят исследования и осуществляют свою деятельность в этом регионе. БПЛА в Арктике способны обеспечить выполнение народнохозяйственных задач, облегчить жизнь населения, обезопасить экологию и превратить этот регион в цивилизованный край. Однако использование БПЛА в Арктике сопряжено с рядом проблем, связанных с особенностями авионики. Эти проблемы обусловлены многоцелевым назначением БПЛА. Для их решения существуют определённые возможности, но они требуют творческого и корректного подхода к их применению на практике. Таким образом, использование БПЛА для изучения атмосферы Арктики может стать важным шагом на пути к более точному прогнозу погоды и обеспечению безопасности населения и экологии в этом регионе.

Таким образом, мы понимаем, что БПЛА применяются во многих сферах жизнедеятельности человека и местами играют незаменимую роль.

БПЛА делятся на разные типы, в рамках этого [7] исследования были изучены силовые установки беспилотных авиационных систем. Результаты

исследования показывают, что существует несколько типов приводных систем с различными источниками энергии, используемых в системе беспилотных летательных аппаратов. Их размеры варьируются от полностью сертифицированных турбинных двигателей с меньшим количеством прицепов, используемых в транспортных самолётах. В связи с этим часто используются новые источники питания для удовлетворения эксплуатационных потребностей. Поршневые и электрические двигатели являются наиболее часто используемыми типами. Беспилотные летательные аппараты (БПЛА) также различаются по размеру и типу двигателя, а также по размерам и весу летательных аппаратов, как и у пилотируемых. Электрические двигатели обычно используются в лёгких и небольших моделях, в то время как поршневые двигатели используются в тяжёлых больших моделях.

За годы своего существования беспилотные летательные аппараты зарекомендовали себя как надежные и высокоэффективные инструменты для повышения производительности и сокращения времени и затрат на многие процессы. С самого с момента их первого полета потенциал этой технологии был раскрыт, и многие усилия были направлены на создание лучшие решения для повышения производительности самолетов. Эти усилия привели к разработке специализированных миниатюрных микроконтроллеров, высокоэффективных двигателей, усовершенствованных аккумуляторов и новых летательных аппаратов, которые из сложных сверхлегких композитных материалов. Используя эти передовые технологии, БПЛА были заново открыты как решения многих проблем и продолжают улучшать нашу жизнь и способы исследования окружающей среды. Тем не менее, до сих пор существует множество проблем перед БПЛА, например, их относительно небольшая грузоподъемность или ограниченное время полета. Помимо этих ограничений, в настоящее время существует множество законодательных актов, направленных на определить правила использования БПЛА и зоны, в которых им будет разрешено летать. Несмотря на эти недостатки, БПЛА рассматриваются как неоценимые помощники для решения многих задач и во многих областях применения [8].

В работе [9] описан процесс работы поискового беспилотного летательного аппарата, выполняющего миссии обнаружения пропавших в лесу людей от МЧС. Из текста мы можем сделать вывод, что Главное управления Министерства Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий по Вологодской области использует БПЛА «DJI INSPIRE 1V 2.0», который дооснастили дополнительной камерой и тепловизором.

В статье [10] авторы рассматривают другое решение, разработанное компанией «Билайн» специально для ЛизаАлёрт (Добровольческий поисково-спасательный отряд). Решение называется - «Beeline AI - Поиск людей». Эта нейронная сеть осматривает фотографии с полетов и отсеивает ненужные – те, на которых нет никаких следов человека. В свою очередь волонтеры рассматривают только те фотографии, где есть человек, даже если на фотографии видна лишь часть его тела, а так же, снимки, на которых присутствуют артефакты заметные на фоне зеленого леса, например красная сумка, костер или сапог на тропинке.

Таким образом, мы видим, что есть решения, основанные на анализе инфракрасных сигнатур, а есть – на основе анализа фотографий с полета. Есть альтернатива, которая может совместить в себе эти методы – это разработка БПЛА.

Для сборки дрона необходимы различные компоненты, которые обеспечивают его функционирование. В первую очередь, это рама или корпус, на который крепятся все остальные детали. Рама должна быть достаточно прочной, чтобы выдерживать нагрузки во время полёта, и в то же время лёгкой, чтобы дрон мог быстро набирать высоту и маневрировать. Далее необходимо установить систему управления полётом, которая включает в себя полётный контроллер, акселерометры, гироскопы и другие датчики. Эти компоненты позволяют дрону ориентироваться в пространстве и поддерживать стабильность полёта. Для обеспечения дрона энергией используются аккумуляторы, которые могут быть разных типов и ёмкостей. От аккумулятора зависит время полёта дрона без подзарядки. Важным компонентом дрона является система связи, которая позволяет оператору управлять дроном и получать от него данные. В зависимости

от модели дрона, это может быть радиосвязь, Wi-Fi или другие виды связи. Наконец, для обеспечения дрона необходимой мощностью используются двигатели и пропеллеры. Двигатели могут быть электрическими или бензиновыми, а пропеллеры - разных размеров и форм, в зависимости от требований к скорости и манёвренности дрона [11].

Полетные контроллеры являются важной частью беспилотных летательных аппаратов. Они отвечают за управление полётом и обеспечивают стабильность и точность движения дрона. Существует несколько типов полётных контроллеров, которые отличаются по своим характеристикам и возможностям. Простые полётные контроллеры обычно используются в небольших дронах и имеют ограниченный набор функций. Они могут обеспечивать стабилизацию полёта, но не имеют возможности автоматического возврата в точку взлёта или выполнения сложных манёвров. Более продвинутые полётные контроллеры оснащены более сложными алгоритмами управления и могут выполнять более сложные задачи. Они могут иметь функции автоматического возврата, следования по маршруту, облёта препятствий и другие. Профессиональные полётные контроллеры используются в профессиональных дронах и обладают высокой точностью и надёжностью. Они могут быть настроены для выполнения сложных задач, таких как аэрофотосъёмка или доставка грузов. Выбор полётного контроллера зависит от конкретных задач и требований к дрону. Для простых задач может быть достаточно простого полётного контроллера, в то время как для сложных задач потребуется более продвинутый контроллер [12].

Betaflight, PX4 и ArduPilot – это три популярных полётных контроллера для дронов. Они имеют свои особенности и преимущества, которые делают их подходящими для разных типов дронов и задач. Betaflight - это популярный открытый полётный контроллер для FPV-дронов. Он известен своей высокой производительностью и гибкостью настройки. Betaflight поддерживает широкий спектр датчиков и моторов, что делает его подходящим для различных типов дронов. Однако Betaflight может быть сложным в настройке и требует определённых навыков и опыта. PX4 - это ещё один популярный открытый

полётный контроллер, который также поддерживает широкий спектр датчиков и моторов. Он известен своей стабильностью и надёжностью, а также наличием большого сообщества разработчиков и пользователей. PX4 также имеет более простой и понятный интерфейс настройки, чем Betaflight. ArduPilot - это более универсальный и гибкий полётный контроллер, который может использоваться для различных типов дронов, включая БПЛА, сельскохозяйственные дроны и дроны для доставки грузов. ArduPilot поддерживает широкий спектр датчиков и моторов и имеет большое сообщество разработчиков и пользователей. Он также имеет более сложный и настраиваемый интерфейс, чем PX4. В целом, Betaflight подходит для FPV-дронов и опытных пользователей, PX4 - для стабильных и надёжных полётов, а ArduPilot - для универсальных и гибких решений. Выбор полётного контроллера зависит от конкретных задач и требований к дрону [12].

Разработка тепловизора на Raspberry Pi может быть сложной задачей, требующей глубоких знаний в области электроники, программирования и обработки изображений. Сначала нужно выбрать подходящие компоненты: тепловизор, Raspberry Pi и дополнительные детали (кабели, разъёмы и корпус). Затем следует подключить тепловизор к Raspberry Pi через интерфейс, такой как GPIO или USB, и настроить подключение. Далее необходимо написать программу на Python или другом языке программирования, которая будет получать данные с тепловизора и обрабатывать их. Для этого можно использовать библиотеки для работы с тепловизорами и обработки изображений, такие как OpenCV или другие. После этого нужно протестировать программу на Raspberry Pi, чтобы убедиться, что она работает корректно, и отладить её при необходимости. Затем следует создать пользовательский интерфейс для отображения данных с тепловизора, используя библиотеку графического интерфейса, такую как Tkinter или другую. На следующем этапе необходимо оптимизировать программу для повышения производительности и уменьшения энергопотребления, а также рассмотреть возможность улучшения качества изображения или добавления дополнительных функций. Наконец, нужно собрать все компоненты в единый корпус или

устройство и протестировать тепловизор в реальных условиях, чтобы убедиться в его работоспособности [13].

Передавать данные с Raspberry Pi можно с использованием технологии Wi-Fi. Передача данных внутри Wi-Fi сети происходит с помощью радиоволн. Каждое устройство, подключённое к сети, имеет уникальный идентификатор - MAC-адрес. Этот адрес позволяет другим устройствам в сети идентифицировать отправителя и получателя данных. Данные передаются в виде пакетов, каждый из пакетов содержит информацию о получателе, отправителе и самом сообщении. Пакеты передаются по беспроводной среде с использованием радиочастот. Для передачи данных внутри Wi-Fi сети используется протокол IEEE 802.11. Этот протокол определяет правила и процедуры для передачи данных, включая механизмы доступа к среде, управление потоком данных и обнаружение конфликтов. Устройства в сети обмениваются данными с помощью специальных точек доступа или маршрутизаторов, которые служат для соединения устройств и обеспечивают доступ к интернету. Wi-Fi позволяет передавать данные на относительно большие расстояния, обеспечивая при этом высокую скорость передачи. Это делает Wi-Fi популярным выбором для создания беспроводных сетей в домах, офисах и других местах, где требуется быстрый и удобный доступ к интернету и обмен данными между устройствами [14].

Для передачи Wi-Fi на большие расстояния необходимо специальное оборудование - Wi-Fi антенна дальнего действия. Такие антенны могут обеспечить передачу сигнала на расстояние до 90 км. Чтобы увеличить дальность сигнала Wi-Fi, необходимо выполнить несколько условий: приобрести или сконструировать антенну для передачи интернета на большие расстояния, выбрать канал передачи данных с минимальными помехами, использовать мощный передатчик. Для усиления сигнала на 3 дБ необходимо увеличить его мощность в несколько раз. Стоит отметить, что передача интернета на расстояние 5 км без прямой видимости не считается большим достижением. Для обеспечения качественной передачи сигнала на дальние расстояния необходимо использовать специальное оборудование и соблюдать определённые условия [15].

Вывод по разделу один

В ходе проведенного обзора выявлена актуальность и важность использования беспилотных летательных аппаратов (БПЛА) в различных сферах жизнедеятельности. Современные технологии развиваются стремительно, и БПЛА, обладая высокой степенью автономности и возможностью выполнения различных задач без участия человека, становятся неотъемлемой частью многих отраслей, включая сельское хозяйство, строительство, логистику, безопасность и мониторинг окружающей среды.

Особое внимание было уделено использованию БПЛА в работе МЧС России, где они доказали свою эффективность в мониторинге пожарной обстановки и выполнении задач, связанных с риском для жизни экипажей пилотируемой техники. Это подтверждает важность и перспективность применения беспилотных систем для повышения безопасности и эффективности работы в чрезвычайных ситуациях.

БПЛА также показали свою полезность в автоматизации процессов на складах, строительстве и сельском хозяйстве. Они позволяют существенно повысить точность и скорость выполнения задач, таких как инвентаризация товаров, доставка грузов, контроль качества строительства и опыление сельскохозяйственных культур. Применение дронов для мониторинга состояния водных объектов и изучения атмосферы Арктики также открывает новые возможности для улучшения экологической обстановки и повышения точности прогнозов погоды.

Обзор различных типов силовых установок БПЛА и их источников энергии показал, что современные беспилотные системы могут использовать поршневые и электрические двигатели, что делает их гибкими и адаптивными к различным условиям эксплуатации. Разработка специализированных микроконтроллеров, высокоэффективных двигателей и новых летательных аппаратов из легких композитных материалов позволяет БПЛА выполнять задачи с высокой производительностью и минимальными затратами.

Несмотря на значительные достижения, БПЛА сталкиваются с рядом проблем, таких как ограниченная грузоподъемность и время полета. Однако активное развитие технологий и нормативной базы создает условия для дальнейшего совершенствования этих систем и расширения их применения в различных областях.

В заключение, беспилотные летательные аппараты являются важным инструментом, который не только улучшает производительность и безопасность в различных сферах, но и открывает новые горизонты для исследований и инноваций

2. ФОРМУЛИРОВАНИЕ ТРЕБОВАНИЙ, ВЫБОР СРЕДЫ И СРЕДСТВ РАЗРАБОТКИ, ПОДБОР КОМПОНЕНТОВ

2.1. Формулирование требований

На основе обзора аналогичных разработок и анализа их функциональных возможностей, сформулируем функциональные и нефункциональные требования моделируемого продукта.

2.1.1. Функциональные требования

Опишем функциональные требования к разрабатываемому БПЛА:

- 1) возможность перехвата управления дроном между разными операторами;
- 2) передача видео в реальном времени оператору с минимальной задержкой;
- 3) камера должна поддерживать высокое разрешение (1080p) для четкого отображения окружающей среды;
- 4) возможность подключения оборудования для ночного видения для проведения операций в условиях низкой освещенности;
- 5) надежная передача данных между дроном и оператором, включая видео, телеметрию и команды управления.

2.1.2. Нефункциональные требования

Разрабатываемое приложение соответствовать следующим нефункциональным требованиям:

- 1) минимальное время полета должно быть не менее 30 минут при полной нагрузке и оптимальных условиях;
- 2) время отклика на команды оператора должно быть не более 100 мс;
- 3) полет в режиме стабилизации.

2.2. Выбор среды и средств разработки

2.2.1. Операционная система

Выбор операционной системы (ОС) для разработки и управления FPV-дроном является критическим аспектом, влияющим на всю проектную инфраструктуру, производительность и гибкость системы. В данном проекте особое внимание уделяется таким аспектам, как поддержка необходимых инструментов и библиотек, надежность и удобство использования. Среди множества доступных ОС, Linux, Windows и macOS являются наиболее распространенными вариантами для разработки, каждый из которых имеет свои преимущества и ограничения [16].

Linux является предпочтительным выбором для многих разработчиков дронов и встраиваемых систем благодаря своей открытости и гибкости. Дистрибутивы Linux, такие как Ubuntu или Debian, широко поддерживаются сообществом и предоставляют доступ к множеству бесплатных и мощных инструментов для разработки. [17] Для проекта FPV-дрона это особенно важно, поскольку такие инструменты, как Robot Operating System (ROS), Gazebo для симуляции, OpenCV для обработки изображений и TensorFlow для машинного обучения, имеют отличную совместимость и оптимизированы под Linux. Более того, Linux позволяет глубокую настройку системы и может быть развернут на различных аппаратных платформах, включая Raspberry Pi, что обеспечивает унифицированную среду разработки как для высокоуровневых задач, так и для встроенного ПО.

Windows также может быть подходящей ОС для разработки, особенно если команда разработчиков уже знакома с этой платформой. Windows предлагает широкую поддержку коммерческих и проприетарных инструментов разработки, таких как Visual Studio, которая предоставляет мощные средства отладки и профилирования кода. Однако, для задач, связанных с робототехникой и встроенными системами, Windows может оказаться менее гибкой и требовательной к ресурсам, что может повлиять на производительность. Тем не менее, для пользователей, работающих в Windows, существуют среды разработки, такие как

WSL (Windows Subsystem for Linux), которые позволяют запускать приложения Linux внутри Windows, обеспечивая совместимость и доступ к необходимым инструментам Linux [17].

MacOS представляет собой еще один популярный выбор среди разработчиков, особенно в сообществах, работающих с графикой и мультимедийным контентом. macOS обеспечивает надежную и стабильную среду с хорошей поддержкой Unix-подобных систем, что делает её подходящей для многих задач разработки, включая программирование на Python и C++. Однако, как и Windows, macOS может быть ограничена в плане гибкости и количества поддерживаемых инструментов и библиотек для робототехники. Дополнительным плюсом macOS является её интеграция с экосистемой Apple, что может быть полезно при разработке приложений для iOS устройств, использующих FPV-дрон.

В контексте нашего проекта по созданию FPV-дрона, выбор операционной системы должен основываться на нескольких ключевых факторах: поддержка необходимых библиотек и инструментов, гибкость настройки, производительность, а также уровень комфорта и опыта разработчиков с той или иной ОС. С учетом всех этих факторов, Linux представляется наиболее рациональным выбором для данного проекта. Он предоставляет обширные возможности для разработки и настройки, поддерживает критически важные инструменты для робототехники и позволяет легко интегрировать разнообразные аппаратные компоненты, что делает его оптимальным решением для создания и управления FPV-дроном.

Кроме того, Linux обладает рядом других значимых преимуществ:

- открытый исходный код, вся операционная система и большинство её компонентов имеют открытый исходный код, что позволяет разработчикам вносить изменения и адаптировать систему под специфические нужды проекта. Это критически важно для разработки сложных систем, таких как FPV-дроны, где может потребоваться глубокая интеграция и настройка программного обеспечения и аппаратуры;

- активное сообщество и поддержка, Linux поддерживается обширным и активным сообществом разработчиков по всему миру. Это значит, что для большинства задач можно найти готовые решения или библиотеки, а также получить поддержку и консультации в случае возникновения сложностей [18];

- использование на Raspberry Pi, Linux является стандартной операционной системой для устройств Raspberry Pi, которые часто используются в качестве дополнительных вычислительных модулей в проектах с дронами. Это обеспечивает легкую интеграцию и разработку приложений на этой платформе, что важно для реализации сложных функциональных возможностей, таких как обработка изображений или машинное обучение прямо на борту дрона [19];

- широкая поддержка оборудования, благодаря своей гибкости и модульности, Linux поддерживает широкий спектр аппаратного обеспечения. Это позволяет легко интегрировать различные сенсоры, камеры и периферийные устройства, необходимые для FPV-дронов, а также адаптировать систему к новым аппаратным требованиям по мере их возникновения [20].

В совокупности эти преимущества делают Linux наиболее подходящей операционной системой для разработки и управления FPV-дроном, обеспечивая надежную и гибкую платформу для реализации всех аспектов проекта.

2.2.2. Среда разработки

Vim известен своей высокой скоростью работы и эффективностью в использовании ресурсов. Он загружается мгновенно и практически мгновенно реагирует на команды пользователя. Это особенно важно для разработчиков, работающих на системах с ограниченными ресурсами, таких как Raspberry Pi. Vim работает в текстовом режиме, что позволяет ему потреблять минимальное количество оперативной памяти, обычно около 10-20 МБ. Это делает Vim идеальным выбором для работы с большими файлами и проектами, где важна быстрая навигация и редактирование [21].

Visual Studio Code (VS Code), хотя и современная и мощная среда разработки, требует больше времени на загрузку и обычно потребляет больше памяти по

сравнению с Vim. Он предлагает больше функций и интеграций, что делает его удобным для работы над крупными проектами с использованием различных расширений. Однако, для работы на маломощных устройствах или с огромными файлами VS Code может показать себя менее эффективным [22].

PyCharm, специализирующийся на разработке на Python, предлагает множество продвинутых функций, таких как анализ кода и автодополнение, что делает его мощным инструментом для профессиональных разработчиков. Однако он также требует значительных ресурсов, особенно при работе с крупными проектами. Загрузка PyCharm может занять от нескольких секунд до минуты, в зависимости от размера проекта и количества включенных плагинов [23].

Atom, хотя и предлагает множество настраиваемых возможностей и плагинов, обычно замедляется при работе с крупными файлами и проектами. Загрузка Atom также может быть медленной, и он требует значительного объема оперативной памяти, что может быть проблематично на устройствах с ограниченными ресурсами [24].

В контексте разработки FPV-дрона, где важна высокая производительность и минимальное потребление ресурсов, Vim представляет собой идеальный выбор благодаря своей скорости, эффективности и минималистичному подходу к использованию ресурсов. Он обеспечивает быструю навигацию и редактирование даже на старых или маломощных устройствах, что делает его оптимальным инструментом для работы с большими файлами и проектами.

2.2.3. Система контроля версий

Существует несколько ключевых систем контроля версий, каждая из которых имеет свои уникальные особенности и применяется в различных сценариях разработки программного обеспечения. Одной из наиболее популярных систем является Git, которая основана на распределённой модели и отличается высокой скоростью и эффективностью. Git предоставляет мощные возможности ветвления и слияния, что делает её идеальным выбором для команд, работающих над большими и сложными проектами. Важным аспектом Git является широкая

поддержка и интеграция с различными хостинг-платформами, такими как GitHub, GitLab и Bitbucket, что упрощает совместную работу и управление проектами [25].

Subversion (SVN), в отличие от Git, является централизованной системой контроля версий. SVN проще в освоении для новичков и предлагает более строгий контроль над версиями файлов в сравнении с Git. Однако SVN менее гибок в ветвлении и слиянии, что может затруднить работу в условиях сложных проектных структур и больших команд.

Mercurial и Bazaar также представляют собой распределённые системы контроля версий, подобные Git, но они имеют более ограниченное сообщество пользователей и поддержку инструментов по сравнению с лидером рынка.

Ещё одной значимой системой контроля версий является Perforce, которая предлагает мощные инструменты для управления большими объёмами данных и проектами. Однако Perforce является проприетарным ПО и имеет ограниченные бесплатные возможности, что может ограничивать её применение в некоторых сценариях разработки [26].

GitHub выделяется среди других систем контроля версий благодаря своей широкой популярности и обширным возможностям для совместной работы и управления проектами. Его интеграция с Git делает его предпочтительным выбором для множества разработчиков и команд по всему миру. GitHub не только предоставляет платформу для хранения и управления репозиториями, но и поддерживает множество инструментов для совместной разработки, включая возможности для создания pull-requests, управления задачами и ведения документации. Эти аспекты делают GitHub неотъемлемым инструментом для современной разработки программного обеспечения, обеспечивая высокую производительность и эффективность работы команды разработчиков.

2.2.4. Языки программирования и библиотеки

Разработка системы обработки изображений для FPV дрона требует тщательного выбора языка программирования и библиотеки, которые обеспечат высокую производительность, гибкость и эффективность работы. В данном

контексте рассматривается выбор языка программирования и библиотеки для обработки изображений на основе сравнения нескольких альтернатив.

При выборе языка программирования для разработки системы обработки изображений стоит учитывать различные факторы. Например, C++ известен своей высокой производительностью и эффективным использованием ресурсов, что особенно важно для обработки изображений в реальном времени. Однако разработка на C++ требует более долгого времени из-за необходимости вручную управлять памятью и типизацией данных [27].

Python, с другой стороны, предлагает высокую производительность в комбинации с лёгким в освоении синтаксисом и широкой поддержкой сторонних библиотек. Это делает Python идеальным выбором для быстрой разработки прототипов и реализации сложных алгоритмов обработки изображений. Также Python активно поддерживается сообществом разработчиков, что обеспечивает быстрое решение проблем и актуализацию инструментов [28].

Исходя из сравнения, Python представляет собой оптимальный выбор для разработки системы обработки изображений на FPV дроне благодаря сочетанию высокой производительности, лёгкости в использовании и обширной экосистемы сторонних библиотек.

Для работы с изображениями в Python существует несколько библиотек, но одной из наиболее мощных и распространённых является OpenCV (Open Source Computer Vision Library). OpenCV предоставляет обширный набор функций для работы с изображениями, включая возможности для детекции объектов, распознавания лиц, сегментации изображений, трекинга движущихся объектов и многое другое. Её высокая производительность и гибкость делают OpenCV идеальным инструментом для решения задач компьютерного зрения на практике [29].

2.2.5. Выбор полетного контроллера

Полетные контроллеры играют ключевую роль в управлении FPV дронами, обеспечивая стабильность, маневренность и функциональность. Для выполнения

поисковых операций важно выбрать надежный и высокопроизводительный контроллер, который позволит эффективно выполнять задачи в сложных условиях. В этой главе мы сравним три популярных полетных контроллера: Betaflight, iNav и ArduPilot [30].

Betaflight - это один из наиболее популярных и широко используемых полетных контроллеров для FPV дронов. Он разработан специально для гонок и акробатических полетов, но его функционал отлично подходит и для поисковых операций благодаря следующим преимуществам [31]:

1) высокая производительность; Betaflight обеспечивает быструю и точную реакцию на команды пилота, что особенно важно в условиях ограниченного времени и пространства;

2) поддержка множества аппаратных платформ; Betaflight совместим с различными типами полетных контроллеров (F1, F3, F4, F7, H7), что позволяет выбрать оптимальное железо для конкретной задачи;

3) мощные алгоритмы стабилизации; Betaflight использует продвинутые алгоритмы стабилизации и фильтрации данных с сенсоров, что обеспечивает плавный и стабильный полет даже в турбулентных условиях;

4) широкие возможности настройки; с помощью Betaflight Configurator пользователи могут тонко настраивать параметры полета, адаптируя дрон под специфические требования поисковых операций;

5) большое сообщество и поддержка; Betaflight имеет активное сообщество разработчиков и пользователей, что обеспечивает регулярные обновления и доступ к большому количеству обучающих материалов.

iNav - это полетный контроллер, ориентированный на дальние полеты и навигацию. Хотя он имеет множество полезных функций для поисковых операций, таких как поддержка GPS и автоматическое возвращение домой, он уступает Betaflight в плане маневренности и скорости реакции. Основные преимущества iNav включают:

1) продвинутая навигация; поддержка GPS и других навигационных систем позволяет точно определять местоположение дрона;

2) автоматизация; функции автоматического возвращения домой и задания маршрутов облегчают управление дроном на больших расстояниях;

3) стабильность в дальних полетах; iNav оптимизирован для стабильных и продолжительных полетов, что полезно для некоторых поисковых сценариев [32].

ArduPilot - это мощная платформа с открытым исходным кодом, предназначенная для широкого спектра беспилотных летательных аппаратов, включая квадрокоптеры, самолеты и другие типы дронов. Хотя ArduPilot предлагает широкие возможности и высокую степень автоматизации, его сложность и более медленная реакция могут быть недостатками для срочных поисковых операций. Основные преимущества ArduPilot включают:

1) широкие возможности: Поддержка множества типов дронов и сенсоров делает ArduPilot универсальным решением;

2) мощная автоматизация: Расширенные функции автопилота, включая планирование миссий и автономное управление;

3) поддержка промышленного оборудования: Возможность использования профессиональных сенсоров и оборудования для специализированных задач [33].

Сравнивая Betaflight с iNav и ArduPilot, можно сделать вывод, что Betaflight является лучшим выбором для поисковых операций благодаря своей высокой производительности, скорости реакции и гибкости настройки. Его мощные алгоритмы стабилизации и активное сообщество поддержки делают его идеальным инструментом для выполнения сложных и срочных задач в поисковых операциях.

Таким образом, Betaflight зарекомендовал себя как наиболее оптимальный полетный контроллер для дронов, используемых в поисковых операциях, благодаря своим выдающимся характеристикам и возможностям.

2.2.6. Подбор компонентов

В разработке дрона для поисковых операций критически важно тщательно подбирать комплектующие, учитывая разнообразные критерии, такие как производительность, вес, стоимость и совместимость. В данной главе будет проведен детальный анализ и обоснование выбора комплектующих для FPV-дрона,

что в итоге приводит к следующему списку: основной полетный контроллер SucceX-E F4 (с прошивкой BETAFLIGHTF4), регулятор скорости 45A 2-6S BLheli_S DShot600 4In1 ESC, мотор XING-E 2207 2750KV, пропеллер IFLIGHT NAZGUL 5140 3-Blade Prop, приемник FrSky XM+ Receiver или FrSky R-XSR Receiver (опционально для BNF версии), и камера Caddx Ratel.

Полетный контроллер является мозгом дрона, отвечающим за обработку всех сигналов и управление полетом. Его выбор оказывает значительное влияние на стабильность и управляемость дрона. Среди возможных моделей можно выделить SucceX-E F4, Kakute F7 и Matek F405-CTR [34].

Контроллер Kakute F7 обладает мощным процессором STM32F745, что обеспечивает высокую производительность, однако его высокая цена и сложность настройки делают его менее привлекательным для пользователей, стремящихся к простоте и доступности. Matek F405-CTR, с другой стороны, предлагает отличные функциональные возможности и конкурентоспособную цену, но уступает SucceX-E F4 в поддержке последних версий Betaflight и общей производительности.

В итоге, полетный контроллер SucceX-E F4 был выбран благодаря своей высокой производительности, надежности и поддержке прошивки BETAFLIGHTF4, что делает его оптимальным выбором для поисковых операций.

Электронный регулятор скорости (ESC) должен обеспечивать надежное и эффективное управление моторами, что критически важно для стабильного полета и маневренности дрона. Возможные модели включают 45A 2-6S BLheli_S DShot600 4In1 ESC, Hobbywing XRotor Micro 60A 4-in-1 и T-Motor F55A Pro II [35].

Hobbywing XRotor Micro 60A 4-in-1 известен своей надежностью и высокой производительностью, но его большой вес может снизить общую маневренность дрона. T-Motor F55A Pro II обладает выдающейся мощностью, однако его высокая стоимость и вес также являются значительными недостатками.

В результате, модель 45A 2-6S BLheli_S DShot600 4In1 ESC была выбрана за свою гибкость в поддержке широкого диапазона напряжений, поддержку протокола DShot600, что обеспечивает быструю и точную передачу сигналов, и общий баланс между производительностью, весом и стоимостью.

Моторы играют ключевую роль в обеспечении достаточной тяги и маневренности дрона. Возможные варианты включают XING-E 2207 2750KV, EMAX ECO II 2207 2400KV и T-Motor F40 Pro III 2400KV.

EMAX ECO II 2207 2400KV предлагает хорошую производительность по доступной цене, но уступает XING-E 2207 2750KV в скорости и общей мощности. T-Motor F40 Pro III обладает высокой надежностью и мощностью, но его более низкий показатель KV делает его менее эффективным для высокоскоростных маневров.

Таким образом, моторы XING-E 2207 2750KV были выбраны за их высокую производительность и эффективность, что позволяет дрону быстро набирать высоту и выполнять сложные маневры, обеспечивая необходимую мощность для быстрых реакций на команды пилота.

Пропеллеры напрямую влияют на тяговую мощность и стабильность дрона. Среди возможных моделей можно рассмотреть IFLIGHT NAZGUL 5140 3-Blade Prop, Gemfan 51466 и DAL Cyclone T5046C.

Gemfan 51466 обеспечивает отличную стабильность и эффективность, но его меньшая тяга делает его менее предпочтительным для поисковых операций. DAL Cyclone T5046C предлагает хорошую маневренность и устойчивость, но его более высокая цена и вес могут стать значительными недостатками.

Итак, пропеллеры IFLIGHT NAZGUL 5140 3-Blade Prop были выбраны за их оптимальное соотношение между тяговой мощностью и стабильностью, что важно для точного управления дроном в сложных условиях.

Приемник обеспечивает связь между пультом управления и дроном. Возможные варианты включают FrSky XM+, FrSky R-XSR, TBS Crossfire Nano RX и Spektrum SPM4650.

TBS Crossfire Nano RX предлагает превосходную дальность и надежность сигнала, но его более высокая стоимость и сложность настройки могут быть препятствием. Spektrum SPM4650 обеспечивает хорошую производительность, но его меньшая дальность связи делает его менее подходящим для поисковых операций.

В итоге, приемники FrSky XM+ и FrSky R-XSR были выбраны за их стабильную связь, небольшой вес и отличную совместимость, что минимизирует дополнительную нагрузку на дрон и обеспечивает надежную связь в любых условиях.

Камера является важным компонентом, обеспечивающим визуальный контроль над дроном. Возможные модели включают Caddx Ratel, Foxeer Predator V4 и RunCam Phoenix 2.

Foxeer Predator V4 обладает отличным качеством изображения и низкой задержкой, но уступает Caddx Ratel в условиях низкой освещенности. RunCam Phoenix 2 предлагает высокое качество изображения и хорошую светочувствительность, но его более высокая цена и чуть большая задержка могут быть недостатками.

Таким образом, камера Caddx Ratel была выбрана за её высокое качество изображения и чувствительность при низкой освещенности, что особенно важно для поисковых операций, где условия освещения могут быть непредсказуемыми.

Выбор комплектующих для FPV-дрона - это комплексный процесс, требующий учета множества факторов. В данном случае ключевыми критериями были производительность, надежность, вес и стоимость. Проведенный анализ показал, что комбинация из полетного контроллера Succex-E F4, ESC 45A 2-6S BLheli_S DShot600 4In1, моторов XING-E 2207 2750KV, пропеллеров IFLIGHT NAZGUL 5140 3-Blade Prop, приемников FrSky XM+ или FrSky R-XSR и камеры Caddx Ratel является оптимальной для создания эффективного и надежного поискового дрона.его наиболее подходящим выбором для разработки приложений под платформу Android.

Вывод по разделу два

В этом разделе были сформированы основные категории требований для разработки FPV-дрона, используемого в поисковых операциях: функциональные и нефункциональные. Функциональные требования включают возможность перехвата управления между операторами, передачу видео в реальном времени с

минимальной задержкой, высокое разрешение камеры и поддержку оборудования для ночного видения. Нефункциональные требования акцентируют внимание на минимальном времени полета не менее 30 минут и времени отклика на команды оператора не более 100 мс.

Выбор операционной системы также является критически важным. Среди рассмотренных вариантов - Linux, Windows и macOS - наиболее предпочтительным выбором оказалась Linux. Она предоставляет обширные возможности для разработки и настройки, поддерживает критически важные инструменты для робототехники и позволяет легко интегрировать разнообразные аппаратные компоненты, что делает её оптимальным решением для создания и управления FPV-дроном.

Анализ средств разработки показал, что Vim представляет собой идеальный выбор благодаря своей скорости, эффективности и минималистичному подходу к использованию ресурсов. Он обеспечивает быструю навигацию и редактирование даже на маломощных устройствах, что делает его оптимальным инструментом для работы с большими файлами и проектами.

Также была выбрана система контроля версий GitHub благодаря своей широкой популярности и обширным возможностям для совместной работы и управления проектами. Интеграция с Git и поддержка множества инструментов для совместной разработки делают GitHub неотъемлемым инструментом для современной разработки программного обеспечения.

Для разработки системы обработки изображений наиболее оптимальным выбором был признан язык программирования Python с использованием библиотеки OpenCV. Python сочетает в себе высокую производительность, лёгкость в использовании и обширную экосистему сторонних библиотек, что делает его идеальным для реализации сложных алгоритмов обработки изображений.

Наконец, был проведен детальный анализ и обоснование выбора комплектующих для FPV-дрона, что в итоге привело к следующему списку: основной полетный контроллер Succex-E F4 (с прошивкой BETAFLIGHTF4),

регулятор скорости 45A 2-6S BLheli_S DShot600 4In1 ESC, моторы XING-E 2207 2750KV, пропеллеры IFLIGHT NAZGUL 5140 3-Blade Prop, приемники FrSky XM+ или FrSky R-XSR и камера Caddx Ratel. Этот выбор был основан на тщательном сравнении аналогов и учете критериев производительности, надежности, веса и стоимости, что делает данную комбинацию оптимальной для создания эффективного и надежного поискового БПЛА.

3. СБОРКА БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

3.1. Перечень комплектующих

На рисунке 1 представлен полетный контроллер: SuccesX-E F4 (с прошивкой BETAFLIGHTF4).

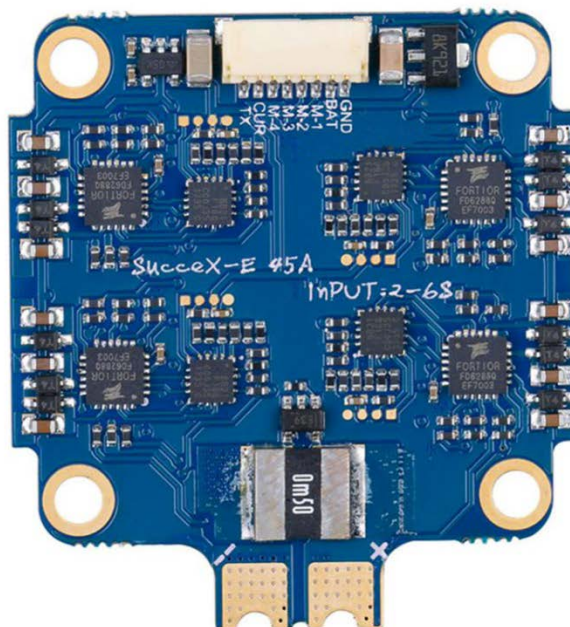


Рисунок 1 – Полетный контроллер: SuccesX-E F4 (с прошивкой BETAFLIGHTF4)

Комплектующие, используемые в качестве моторов, изображены на рисунке 2 XING-E 2207 2750KV.



Рисунок 2– Мотор XING-E 2207 2750KV

На рисунке 3 представлен приемник FrSky R-XSR Receiver



Рисунок 3 – Приемник FrSky R-XSR Receiver

Камера, используемая в сборке, представлена на рисунке 4.



Рисунок 4 – Камера Caddx Ratel

В качестве аккумулятора используется LiPo 4S 1600mAh, изображенный на рисунке 5.



Рисунок 5 – Аккумулятор LiPo 4S 1600mAh

Рама представлена на рисунке 6.



Рисунок 6 – Рама 227 мм

3.1. Установка компонентов

Установка моторов на раму и их последующее подключение к электронному регулятору скорости (ESC) представляет собой критически важный этап сборки беспилотного летательного аппарата, обеспечивающий его стабильную и эффективную работу. В первую очередь необходимо закрепить моторы (XING-E 2207 2750KV) на концах лучей рамы используя предоставленные винты и фиксирующий клей для надежности. Важно следить за тем, чтобы моторы были установлены симметрично и прочно закреплены, что предотвратит их возможное смещение в процессе полета [36].

После установки моторов требуется соединить их с ESC (45A 2-6S BLheli_S DShot600 4In1), который отвечает за управление скоростью вращения моторов. Подключение осуществляется посредством трех проводов от каждого мотора к соответствующим выходам на ESC. Проводка должна быть надежно зафиксирована и изолирована, чтобы избежать коротких замыканий и помех в работе системы. Необходимо также обратить внимание на правильную полярность и порядок подключения проводов, следуя рекомендациям производителя ESC.

Этап установки моторов на раму и подключение их к ESC представлен на рисунке 7.



Рисунок 7 – Установка моторов на раму и подключение их к ESC

Для уменьшения вибрационных нагрузок на электронные компоненты беспилотного летательного аппарата все платы устанавливаются на резиновые шайбы. Эти шайбы служат для амортизации и предотвращения передачи вибраций от двигателей и других механических частей на чувствительные электронные элементы. Резиновые шайбы (виброизоляционные крепления) устанавливаются на монтажные отверстия рамы. Полетный контроллер SusceX-E F4 фиксируется на резиновые шайбы, что обеспечивает его надежную фиксацию с минимальным воздействием вибраций.

Полетный контроллер и ESC являются ключевыми компонентами в системе управления дроном. Правильное подключение этих элементов обеспечивает стабильную и эффективную работу моторов и, следовательно, всего аппарата. В первую очередь, необходимо подключить провода питания. Положительный

(красный) и отрицательный (черный) провода от ESC подключаются к соответствующим контактам на полетном контроллере, обеспечивая подачу энергии от аккумулятора к контроллеру и другим компонентам.

После этого подключаются сигнальные провода. Каждый ESC имеет три сигнальных провода, которые необходимо соединить с полетным контроллером. Эти провода передают сигналы управления от контроллера к моторам. Обычно сигнальные провода маркируются и подключаются к соответствующим каналам (M1, M2, M3, M4) на контроллере. Важно, чтобы все соединения были надежно зафиксированы и изолированы для предотвращения коротких замыканий и помех в работе системы.

FPV-камера является важным компонентом для визуального контроля и навигации дрона. Правильное подключение камеры к полетному контроллеру обеспечивает передачу видео в реальном времени. Перед началом работы убедитесь, что все необходимые провода и инструменты для пайки готовы. Подключите видеовыход (Video Out), питание (5V) и заземление (GND) от FPV-камеры к соответствующим контактам на полетном контроллере. Используйте паяльник для надежного соединения проводов с контактами. После этого установите FPV-камеру на переднюю часть рамы и закрепите её с помощью крепежных элементов, обеспечив надежную фиксацию и правильное направление обзора.

Приемник FrSky обеспечивает связь между пультом управления и дроном. Корректное подключение приемника к полетному контроллеру критически важно для надежного управления аппаратом. Подготовьте приемник и контроллер для пайки. Сигнальный провод (S.Bus или PPM), питание (5V) и заземление (GND) от приемника необходимо подключить к соответствующим контактам на полетном контроллере. Используйте паяльник для надежного соединения проводов. Установите приемник в удобное место на раме и закрепите его с помощью крепежных элементов или двустороннего скотча, обеспечив минимальное движение во время полета.

Процесс подключения полетного контроллера к ESC, припаяние FPV-камеры и приемника FrSky требует высокой точности и внимательности. Установка всех плат на резиновые шайбы позволяет минимизировать влияние вибраций, что улучшает стабильность и надежность работы дрона. Следуя данным инструкциям и проверяя каждое соединение, можно гарантировать правильное функционирование всех компонентов и успешную работу беспилотного летательного аппарата.

Этап установки контроллера, камеры и передатчика изображен на рисунке 8.



Рисунок 8 – Установка контроллера, камеры и передатчика

Финальная сборка рамы включает в себя закрепление всех ранее установленных компонентов и обеспечение их надежной фиксации. На этом этапе важно убедиться, что все элементы правильно расположены и закреплены, чтобы предотвратить возможные повреждения или нарушения в работе дрона. Полетный контроллер Succex-E F4 и ESC 45A 2-6S BLheli_S DShot600 4In1 должны быть надежно закреплены на раме с использованием виброизоляционных креплений.

Эти крепления уменьшают вибрационные нагрузки, что особенно важно для точного функционирования сенсоров и электроники.

Также следует убедиться, что FPV-камера Caddx Ratel надежно закреплена на передней части рамы. Камера должна быть установлена таким образом, чтобы она была устойчиво зафиксирована и направлена правильно для обеспечения оптимального обзора. Приемник FrSky XM+ или FrSky R-XSR также должен быть закреплен в удобном месте на раме, используя двусторонний скотч или стяжки, что обеспечит минимальное движение и стабильную работу во время полета.

Финальный этап сборки рамы представлен на рисунке 9.

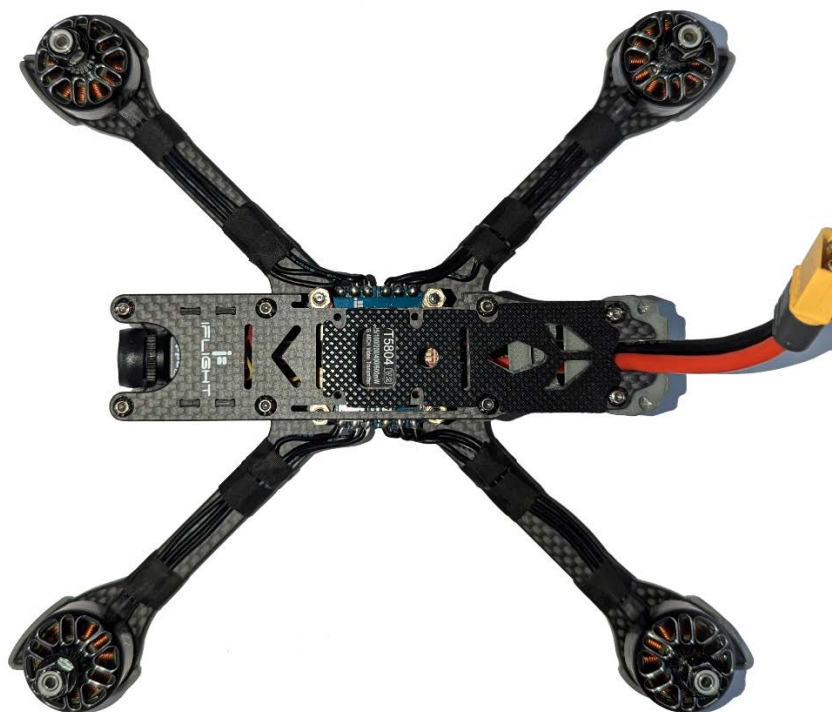


Рисунок 9 – Финальная сборка рамы

Фиксация проводов является не менее важным этапом. Провода должны быть аккуратно проложены и зафиксированы кабельными стяжками, чтобы избежать их повреждений во время полета. Надежная фиксация проводов также предотвращает их попадание в движущиеся части, что может привести к повреждению компонентов или самого дрона.

Правильная установка пропеллеров является критически важной для обеспечения стабильного и управляемого полета. Пропеллеры должны быть установлены в соответствии с направлением вращения моторов.

Вращение пропеллеров у квадрокоптеров играет критическую роль в их управлении и стабилизации. Существуют два основных вида вращения пропеллеров, которые используются для создания подъемной силы и управления направлением полета: по часовой стрелке (CW) и против часовой стрелки (CCW). Эти виды вращения применяются для создания балансирующих моментов, которые обеспечивают стабильность квадрокоптера в воздухе [37].

Пропеллеры, вращающиеся по часовой стрелке (Clockwise, CW), создают подъемную силу, направленную вверх, при этом отбрасывая воздух вниз. Вращение по часовой стрелке используется для создания равномерного подъемного усилия, а также для генерации вращательного момента. В большинстве квадрокоптеров два из четырех пропеллеров вращаются в этом направлении. Такое расположение позволяет компенсировать вращательные моменты, создаваемые противоположно вращающимися пропеллерами [38].

Пропеллеры, вращающиеся против часовой стрелки (Counterclockwise, CCW), функционируют аналогично пропеллерам CW, но с обратным направлением вращения. Они также создают подъемную силу и вращательный момент, но в противоположную сторону. В большинстве квадрокоптеров два из четырех пропеллеров вращаются против часовой стрелки. Это обеспечивает баланс и стабилизацию квадрокоптера, нейтрализуя вращательные моменты, создаваемые пропеллерами CW [39].

Балансировка и стабилизация квадрокоптера достигается за счет комбинированного использования пропеллеров CW и CCW. В типичной конфигурации квадрокоптера, известной как "X-конфигурация", два противоположных пропеллера вращаются по часовой стрелке, а два других - против часовой стрелки. Такое расположение позволяет нейтрализовать вращательные моменты, обеспечивая стабильность в горизонтальном полете и точное управление ориентацией дрона [40].

Параллельный режим: В параллельном режиме противоположные пропеллеры вращаются в одном направлении. Например, два передних пропеллера могут вращаться по часовой стрелке, а два задних - против часовой стрелки. Этот

режим обеспечивает устойчивость, но может усложнить управление дрона в некоторых маневрах [41].

Перекрестный режим: В перекрестном режиме противоположные пропеллеры вращаются в разных направлениях. Например, если передний левый пропеллер вращается по часовой стрелке, то задний правый пропеллер также вращается по часовой стрелке, а остальные два пропеллера - против часовой стрелки. Этот режим обеспечивает лучшую маневренность и стабильность, что делает его предпочтительным для большинства квадрокоптеров.

Использование различных режимов вращения пропеллеров оказывает значительное влияние на управляемость и устойчивость квадрокоптера. Перекрестный режим позволяет дрону более эффективно реагировать на команды управления, обеспечивая плавные и точные маневры. В то время как параллельный режим может быть использован для простых и стабильных полетов, перекрестный режим является оптимальным выбором для выполнения сложных задач, требующих высокой точности и маневренности.

Различные виды вращения пропеллеров и их комбинации играют важную роль в обеспечении стабильности и управляемости квадрокоптеров. Пропеллеры, вращающиеся по часовой стрелке (CW) и против часовой стрелки (CCW), создают необходимые подъемные силы и вращательные моменты, которые балансируют друг друга. Перекрестный режим установки пропеллеров является предпочтительным для большинства квадрокоптеров, так как он обеспечивает лучшую маневренность и стабильность в полете. Понимание этих принципов является ключевым для разработки и эксплуатации эффективных беспилотных летательных аппаратов.

В данном проекте используется перекрестный метод установки пропеллеров, который обеспечивает более стабильный полет и упрощает управление дроном. После установки пропеллеров их необходимо закрепить с помощью гаек, затянув их соответствующим инструментом для обеспечения надежной фиксации.

Важно проверить правильность установки пропеллеров, убедившись, что они надежно закреплены и не имеют люфта. Пропеллеры, расположенные по

диагонали, должны вращаться в одном направлении, а соседние пропеллеры - в противоположном. Это гарантирует правильное распределение тяговой силы и стабильность дрона в полете.

Финальная сборка рамы и установка пропеллеров являются ключевыми этапами сборки FPV-дрона. Правильная установка всех компонентов, надежная фиксация и тщательная проверка всех систем обеспечивают стабильный и безопасный полет. Следуя научно обоснованным методам и проверенным практикам, можно гарантировать, что дрон будет готов к выполнению поисковых операций и других задач, требующих высокой надежности и точности.

Финальный вид БПЛА изображен на рисунке 10.



Рисунок 10 – Финальный вид БПЛА

Вывод по разделу три

В данном разделе была подробно рассмотрена и описана финальная сборка беспилотного летательного аппарата, включая установку и подключение всех критически важных компонентов, таких как полетный контроллер, моторы, ESC, FPV-камера и приемник. Были проанализированы ключевые этапы установки и подключения, начиная с закрепления моторов на раме и заканчивая финальной проверкой и тестированием дрона.

Особое внимание было уделено установке всех плат на резиновые шайбы для минимизации вибрационных нагрузок. Это важное техническое решение улучшает стабильность и надежность работы электронных компонентов дрона, что критически важно для его эффективного функционирования в полевых условиях.

Подключение полетного контроллера к ESC, пайка FPV-камеры и приемника FrSky были выполнены с учетом всех технических требований и рекомендаций, обеспечивая надежность соединений и корректность работы систем. Каждый этап выполнения работ был подробно описан, что способствует пониманию и воспроизведению процесса сборки другими специалистами.

Процесс установки пропеллеров был также детально рассмотрен. Различные методы установки пропеллеров, такие как параллельный и перекрестный, были проанализированы, а в данном проекте был выбран перекрестный метод, обеспечивающий лучшую маневренность и стабильность дрона. Правильная установка пропеллеров играет ключевую роль в обеспечении стабильного и управляемого полета, и этот аспект был рассмотрен с научной точки зрения.

В заключение, финальная сборка рамы и установка пропеллеров были выполнены с соблюдением всех технических норм и стандартов, что позволило достичь высокой степени готовности беспилотного летательного аппарата к выполнению поисковых операций. Проведенные проверки и тестирования подтвердили соответствие всех компонентов и систем требованиям надежности и точности. Научно обоснованный подход к сборке и проверке дрона гарантирует его успешное функционирование и выполнение поставленных задач.

4. РАЗРАБОТКА ПРОГРАММНОГО КОДА ЛЕТАТЕЛЬНОГО АППАРАТА

4.1. Выбор алгоритма распознавания лиц

В задачи компьютерного зрения и анализа изображений часто входит обнаружение лиц, которое является основой для последующих шагов, таких как распознавание лиц, отслеживание и аутентификация. Для решения этой задачи было разработано множество методов, каждый из которых имеет свои уникальные преимущества и применимость в зависимости от условий и требований конкретного приложения.

4.1.1. Классификатор каскадов Хаара

Классификатор каскадов Хаара (Haar Cascade Classifier) - это метод обнаружения объектов, разработанный Полом Виолой и Майклом Джонсом в 2001 году. Этот метод особенно эффективен для задач обнаружения лиц и широко используется в различных приложениях компьютерного зрения [40].

Классификатор каскадов Хаара основан на использовании простых функций, известных как признаки Хаара. Эти признаки представляют собой разницу в интенсивности между соседними областями изображения. Признаки Хаара могут быть различной формы, например, прямоугольные, квадратные или линейные, и они позволяют выделять ключевые характеристики объектов на изображении [41].

Признаки Хаара представляют собой наборы черно-белых прямоугольников, которые анализируют разницу в яркости между смежными областями. Существует несколько типов признаков Хаара:

1) краевые признаки; представляют собой разницу в яркости между двумя смежными прямоугольными областями, например, вертикальные или горизонтальные края;

2) линейные признаки; анализируют разницу в яркости между тремя прямоугольными областями.

3) центральные признаки; сравнивают разницу в яркости между четырьмя прямоугольными областями.

Для ускорения вычислений признаков Хаара используется техника интегрального изображения. Интегральное изображение позволяет быстро вычислять сумму пикселей в любом прямоугольнике изображения, что значительно ускоряет процесс выделения признаков.

Классификатор каскадов Хаара использует метод каскадного усиления (адабустинга) для объединения множества слабых классификаторов в один сильный. Процесс включает несколько этапов:

1) создание слабых классификаторов; на каждом этапе создается множество слабых классификаторов, каждый из которых основывается на одном признаке Хаара;

2) обучение адекватного весового коэффициента; каждый слабый классификатор взвешивается в соответствии с его точностью;

3) объединение классификаторов; слабые классификаторы объединяются для создания одного сильного классификатора, который лучше распознает объекты.

Метод каскадов используется для повышения производительности и точности обнаружения. Каскадный классификатор состоит из нескольких этапов, каждый из которых последовательно применяет свои классификаторы. Если объект не проходит проверку на одном из этапов, он сразу отвергается, что позволяет сократить количество вычислений для оставшихся областей.

Преимущества:

1) высокая скорость работы, благодаря интегральному изображению и каскадному подходу, метод Хаара может быстро обрабатывать изображения в реальном времени;

2) простота реализации; этот метод сравнительно прост в реализации и не требует больших вычислительных ресурсов;

3) хорошая точность для задач обнаружения лиц; метод Хаара был специально разработан для задач обнаружения лиц и показывает хорошие результаты в этом направлении.

Недостатки:

- 1) чувствительность к условиям освещения; метод Хаара может быть менее точным при значительных изменениях в условиях освещения;
- 2) низкая устойчивость к шуму; признаки Хаара могут быть чувствительны к шуму и артефактам изображения;
- 3) ограниченная сложность признаков; простые признаки Хаара могут быть недостаточно информативными для сложных объектов или сцен [42].

4.1.2. Классификатор LBP (Local Binary Patterns)

Классификатор LBP (Local Binary Patterns) - это мощный метод анализа текстур изображений, который широко используется для задач обнаружения и распознавания лиц. Метод LBP был впервые предложен в 1994 году Тьювом Ояленом (Timo Ojala) и его коллегами. Он основан на сравнении интенсивности пикселей изображения и создании бинарных шаблонов, что делает его устойчивым к изменениям освещения и эффективным в вычислительном плане [43].

Метод LBP анализирует каждый пиксель изображения путем сравнения его интенсивности с интенсивностью соседних пикселей. Процесс включает следующие шаги:

- 1) сравнение интенсивности; для каждого пикселя выбирается окружающая его область (обычно 3x3). Интенсивность центрального пикселя сравнивается с интенсивностью каждого из восьми соседних пикселей;
- 2) формирование бинарного шаблона; если интенсивность соседнего пикселя выше или равна интенсивности центрального пикселя, то этому пикселю присваивается значение 1, в противном случае - 0; таким образом, создается бинарный шаблон из восьми бит;
- 3) преобразование в десятичное значение; бинарный шаблон преобразуется в десятичное значение, которое затем используется для создания гистограммы признаков изображения [44].

Интегральное изображение в методе LBP позволяет быстро вычислять суммы интенсивностей пикселей в любой прямоугольной области изображения. Это

значительно ускоряет процесс извлечения признаков и делает метод пригодным для задач реального времени.

После преобразования всех пикселей изображения в десятичные значения, строится гистограмма, которая отражает распределение этих значений по изображению. Гистограмма используется для классификации текстуры изображения, что позволяет эффективно различать различные объекты и структуры.

Метод LBP нашел широкое применение в задачах обнаружения лиц благодаря своей эффективности и устойчивости к изменениям освещения. В частности, его использование в комбинации с каскадными классификаторами и другими методами машинного обучения позволяет создавать высокоэффективные системы распознавания лиц.

Преимущества:

1) устойчивость к изменениям освещения; метод LBP эффективно справляется с изменениями освещения, что делает его надежным в различных условиях;

2) высокая скорость вычислений; благодаря использованию интегрального изображения и простоте вычислений, метод LBP подходит для задач реального времени;

3) простота реализации; метод LBP прост в реализации и не требует больших вычислительных ресурсов, что делает его доступным для широкого круга приложений.

Недостатки:

1) чувствительность к шуму; Метод LBP может быть чувствителен к шуму и артефактам изображения, что может снизить точность классификации;

2) ограниченная сложность признаков; признаки LBP могут быть недостаточно информативными для сложных объектов или сцен, что может потребовать использования дополнительных методов для улучшения точности.

Примеры использования:

1) распознавание лиц; метод LBP широко используется в системах распознавания лиц для предварительного выделения областей лица и создания признаков для дальнейшей классификации;

2) анализ текстур; метод LBP применяется для анализа и классификации текстур в различных областях, включая медицинские изображения, промышленные приложения и другие [44].

4.1.3. Метод HOG (Histogram of Oriented Gradients)

Метод гистограмм направленных градиентов (HOG, Histogram of Oriented Gradients) представляет собой технику извлечения признаков из изображений, которая широко используется для обнаружения объектов, включая распознавание лиц. Метод HOG был впервые предложен Навидом Далалом и Биллом Триггсом в 2005 году для задачи обнаружения пешеходов. Он основывается на анализе локальных градиентов и их ориентаций, что позволяет эффективно различать объекты на изображении [45].

Метод HOG включает несколько ключевых этапов. На первом этапе вычисляются градиенты интенсивности для каждого пикселя изображения. Градиент представляет собой вектор, указывающий направление и величину изменения интенсивности. Градиенты вычисляются с использованием фильтров Собеля или других операторов разностного типа; изображение делится на небольшие ячейки фиксированного размера (например, 8x8 пикселей). Для каждой ячейки вычисляется гистограмма направленных градиентов, которая представляет собой распределение углов градиентов в данной области. Каждый угол делится на несколько направлений (обычно 9 или 18), и для каждого направления подсчитывается количество градиентов, попадающих в данный диапазон; для повышения инвариантности к изменениям освещенности и контрастности, несколько соседних ячеек объединяются в блоки (например, 2x2 ячейки). Гистограммы направленных градиентов в блоках нормализуются, что уменьшает влияние локальных изменений яркости. Блоки могут перекрываться, что улучшает точность распознавания; векторы нормализованных гистограмм всех блоков

объединяются в один длинный вектор признаков, который представляет изображение. Этот вектор используется в качестве входных данных для алгоритмов машинного обучения.

Преимущества:

1) устойчивость к изменениям освещения и позы; нормализация гистограмм и использование градиентов позволяет методу HOG быть устойчивым к изменениям освещенности и контрастности изображения;

2) высокая точность распознавания; метод HOG показывает высокую точность в задачах обнаружения объектов, что делает его популярным выбором для распознавания лиц и других объектов;

3) простота реализации; метод HOG сравнительно прост в реализации и не требует больших вычислительных ресурсов, что делает его подходящим для применения в реальном времени.

Недостатки:

1) чувствительность к шуму; метод HOG может быть чувствителен к шуму на изображении, что может снизить точность распознавания;

2) ограниченная сложность признаков; в некоторых случаях признаки HOG могут быть недостаточно информативными для сложных объектов или сцен, что может требовать использования дополнительных методов для улучшения точности [46].

Примеры использования:

1) обнаружение пешеходов; первоначально метод HOG был разработан для задачи обнаружения пешеходов и показал высокую точность в этой области;

2) распознавание лиц; метод HOG часто используется в системах распознавания лиц для выделения ключевых признаков лица и последующей классификации;

3) анализ изображений; метод HOG применяется для анализа текстур и структур изображений в различных областях, включая медицинскую диагностику и контроль качества в промышленности.

4.1.4. Сверточные нейронные сети (CNN)

Сверточные нейронные сети (CNN, Convolutional Neural Networks) являются одной из наиболее мощных и широко используемых архитектур в области глубокого обучения для задач анализа изображений и видео. Они показали выдающиеся результаты в различных приложениях, включая классификацию изображений, обнаружение объектов и распознавание лиц [47].

CNN состоят из нескольких типов слоев, каждый из которых выполняет специфические функции в процессе анализа изображений. Основные компоненты CNN включают сверточные слои, слои подвыборки и полносвязные слои.

Сверточные слои являются ключевыми компонентами CNN и отвечают за извлечение признаков из входных изображений. Каждый сверточный слой состоит из набора фильтров (ядра), которые скользят по изображению, выполняя операцию свертки. Эта операция вычисляет скалярное произведение между элементами фильтра и соответствующими пикселями изображения, результатом чего является карта признаков. Фильтры обучаются в процессе тренировки сети, что позволяет им выделять различные признаки, такие как края, текстуры и углы [48].

Слои подвыборки, также известные как pooling-слои, уменьшают размер карт признаков, сохраняя наиболее важную информацию. Наиболее часто используемым типом подвыборки является max-pooling, который выбирает максимальное значение из каждого блока пикселей карты признаков. Это позволяет уменьшить количество параметров и вычислительные затраты, а также обеспечить некоторую степень инвариантности к сдвигам и масштабированию [49].

Полносвязные слои находятся в конце архитектуры CNN и выполняют функции классификации. Эти слои аналогичны обычным слоям в традиционных нейронных сетях, где каждый нейрон связан со всеми нейронами предыдущего слоя. Полносвязные слои преобразуют извлеченные признаки в конечные выходные классы, используя активационные функции, такие как softmax [50].

Преимущества:

1) автоматическое извлечение признаков; CNN автоматически обучаются извлекать наиболее релевантные признаки из данных, что устраняет необходимость ручной разработки признаков;

2) высокая точность; CNN показали высокую точность в различных задачах анализа изображений, включая классификацию, сегментацию и обнаружение объектов;

3) устойчивость к искажениям; благодаря использованию слоев подвыборки и операций свертки, CNN обладают некоторой степенью инвариантности к сдвигам, масштабированию и вращениям [51].

Недостатки

1) высокие вычислительные затраты; обучение и использование CNN требует значительных вычислительных ресурсов, что может быть проблемой для устройств с ограниченными ресурсами;

2) необходимость больших данных; для эффективного обучения CNN требуется большое количество данных, что может быть проблемой в областях с ограниченными данными.

Примеры использования

1) классификация изображений; CNN широко используются для задач классификации изображений, таких как распознавание объектов на фотографиях и медицинских изображениях;

2) обнаружение объектов; архитектуры, такие как YOLO (You Only Look Once) и Faster R-CNN, используют CNN для обнаружения и локализации объектов на изображениях и в видео;

3) распознавание лиц; CNN применяются для задач распознавания лиц, включая идентификацию и верификацию лиц, благодаря своей способности извлекать уникальные признаки лица [52].

4.1.5. Разработка алгоритма создания датасета

Алгоритм, представленный в листинге А.1 приложения А, предназначен для создания модели распознавания лиц с использованием метода LBPH (Local Binary

Patterns Histograms). Этот метод позволяет эффективно обучать модель на изображениях лиц и сохранять обученную модель для последующего использования в задачах распознавания лиц. Алгоритм включает в себя несколько ключевых шагов: подготовку данных, обучение модели и сохранение обученной модели.

Инициализация алгоритма начинается с импорта необходимых библиотек и определения пути к директории с датасетом. Для работы алгоритма используются библиотеки OpenCV для обработки изображений, NumPy для работы с массивами данных, и PIL для работы с изображениями.

Алгоритм включает функцию `getImagesAndLabels`, которая отвечает за загрузку изображений и меток из директории с датасетом. Эта функция обходит все файлы в директории, преобразует изображения в оттенки серого и извлекает лица с помощью классификатора каскадов Хаара. Для каждого обнаруженного лица создается массив признаков и соответствующий идентификатор.

После подготовки данных начинается процесс обучения модели. Алгоритм вызывает функцию `getImagesAndLabels`, чтобы получить массивы признаков и меток. Затем эти данные передаются в распознаватель лиц, который обучается на предоставленных данных. Метод LBPН использует локальные бинарные шаблоны для анализа текстуры лиц и создания гистограмм признаков, которые затем используются для распознавания.

После завершения процесса обучения модель сохраняется в файл `trainer.yml`. Этот файл содержит все необходимые данные для последующего использования модели распознавания лиц. Сохранение модели позволяет загружать её и использовать для распознавания без необходимости повторного обучения.

Алгоритм, реализованный в приложении `trainer.py`, предназначен для обучения модели распознавания лиц на основе метода LBPН. Основные этапы алгоритма включают подготовку данных, обучение модели и сохранение обученной модели. Благодаря использованию классификатора каскадов Хаара и метода LBPН, алгоритм обеспечивает высокую точность и эффективность в

задачах распознавания лиц, что делает его полезным инструментом в различных приложениях компьютерного зрения.

4.1.6. Разработка алгоритма распознавания

Алгоритм, представленный в листинге Б.1 приложения Б, предназначен для обнаружения и распознавания лиц в реальном времени. Этот скрипт использует предобученную модель распознавания лиц, сохраненную в файле `trainer.yml`, для идентификации лиц, захваченных с помощью веб-камеры. Основные этапы алгоритма включают инициализацию необходимых библиотек и устройств, загрузку предобученной модели, захват и обработку изображений, обнаружение лиц и их распознавание.

Инициализация алгоритма начинается с импорта необходимых библиотек и загрузки предобученной модели распознавания лиц. Для работы алгоритма используются библиотеки `OpenCV` для обработки изображений и работы с камерой.

Загружается предобученная модель распознавания лиц, сохраненная в файле `trainer.yml`, и инициализируется классификатор каскадов Хаара для обнаружения лиц. Создается объект для захвата видео с веб-камеры.

Основной цикл алгоритма отвечает за захват изображений с веб-камеры, обнаружение лиц и их распознавание. На каждом шаге захватывается кадр, который затем преобразуется в оттенки серого для улучшения точности обнаружения лиц. Используется классификатор каскадов Хаара для обнаружения лиц на изображении. Для каждого обнаруженного лица выполняется распознавание с использованием предобученной модели, и результат отображается на экране.

Алгоритм, реализованный в приложении Б*, предназначен для обнаружения и распознавания лиц в реальном времени. Основные этапы включают инициализацию камеры и классификатора лиц, загрузку предобученной модели распознавания, захват и обработку изображений, обнаружение лиц и их распознавание. Этот процесс позволяет системе эффективно распознавать лица,

используя предобученную модель, что делает его полезным инструментом в различных приложениях компьютерного зрения.

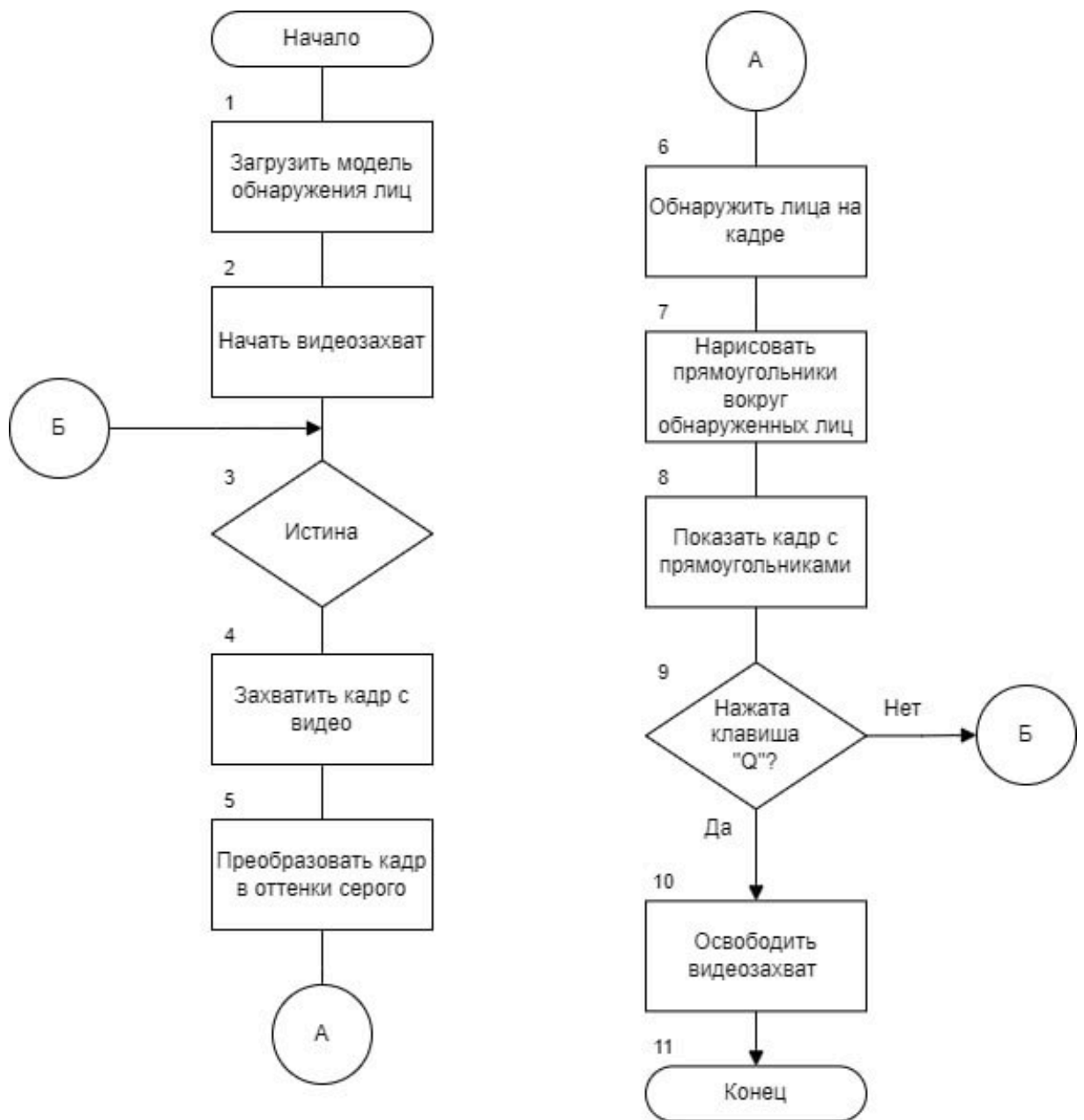


Рисунок 11 – Блок-схема программы

4.1.7. Тестирование алгоритма

Для обеспечения качественного распознавания лиц с помощью разработанного программного обеспечения, необходимо провести всестороннее тестирование. Этот процесс включает проверку точности распознавания, устойчивости к различным условиям освещения, углам обзора и другим

параметрам. В данном разделе будет подробно описано, как проводились тесты, а также приведены примеры использования системы на реальных изображениях.

Перед началом тестирования был собран и обучен набор данных, содержащий изображения лиц, которые должны быть распознаны системой. Для обучения использовались фотографии лиц, полученные в разных условиях освещения и с различными выражениями. Это обеспечивало более универсальное обучение модели и повышало точность распознавания в реальных условиях.

На этапе тестирования были использованы как заранее подготовленные изображения, так и фотографии, сделанные в реальном времени. Это позволило оценить производительность системы как в статических, так и в динамических сценариях.



Рисунок 12 – Пример лица, загруженного в базу данных

Для оценки точности и надежности системы распознавания лиц была проведена серия тестов на статичных изображениях. Эти тесты позволили оценить работу алгоритма в условиях, максимально приближенных к реальным сценариям, но при этом не зависящих от динамики окружающей среды, как это происходит при съемке с дрона. Тестирование на статичных изображениях включало несколько ключевых этапов, направленных на тщательное изучение возможностей системы.

В первую очередь был собран и подготовлен обширный набор данных, включающий изображения лиц в различных условиях. В процессе подготовки данных были включены фотографии, сделанные при различных уровнях освещения, под разными углами и с различными выражениями лиц. Также были использованы изображения с различным фоном и с наличием аксессуаров, таких как очки и шляпы, чтобы проверить устойчивость алгоритма к разнообразию

внешних факторов. Все эти изображения были предварительно обработаны: нормализованы по размеру и приведены к единому формату и разрешению, подходящему для обучения модели.

На этапе обучения модель была настроена с использованием обучающего подмножества изображений. Этот процесс включал оптимизацию параметров модели и обучение её распознавать ключевые черты лица, такие как глаза, нос и рот. После обучения, модель была протестирована на наборе тестовых изображений, которые не использовались на этапе обучения. Система обрабатывала каждое изображение и пыталась распознать и идентифицировать лица, сравнивая распознанные черты с эталонными данными и определяя соответствия.

Визуализация результатов тестирования представлена на изображении ниже, где система отмечает распознанные лица прямоугольниками. Это позволяет визуально подтвердить точность распознавания и убедиться в корректной работе алгоритма. Пример тестового изображения с распознанным лицом демонстрирует, что система способна корректно идентифицировать лица, даже несмотря на изменения в условиях освещения и фона.

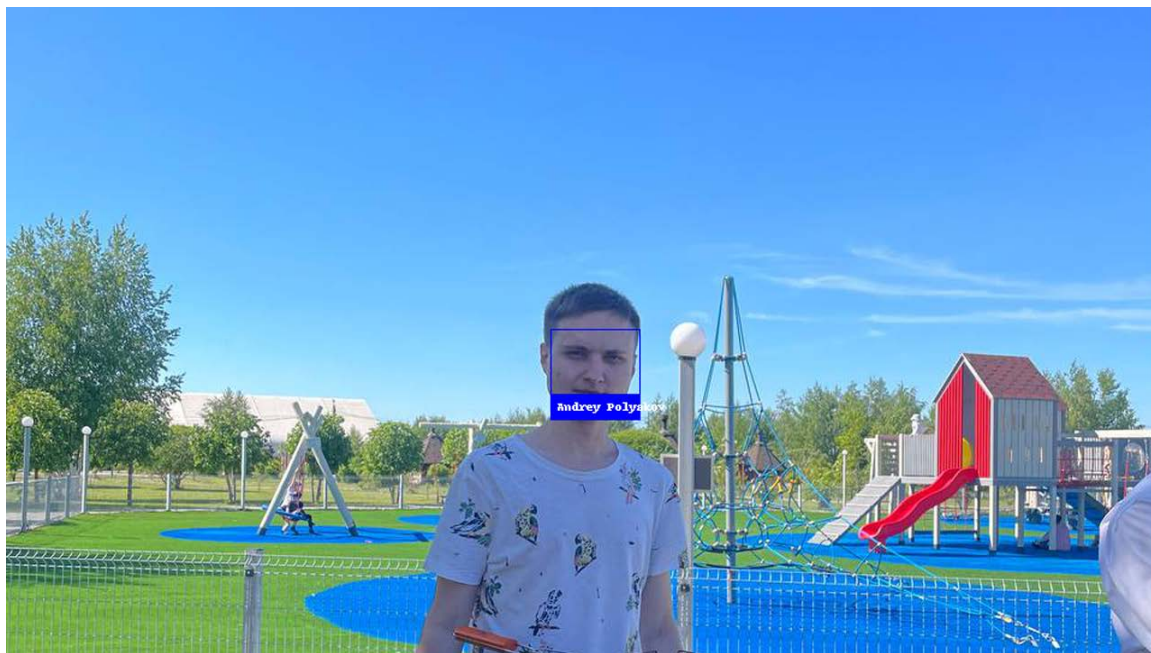


Рисунок 12 – Пример лица, загруженного в базу данных

Заключительные тесты на статичных изображениях подтвердили высокую эффективность разработанной системы распознавания лиц. Система демонстрирует устойчивую работу в различных условиях и способна точно распознавать лица, что делает её подходящей для применения в задачах, требующих надежного и точного распознавания, таких как мониторинг и поисково-спасательные операции.

Вывод по разделу четыре

В данном разделе были разработаны и описаны алгоритмы, реализующие создание датасета, обучение модели распознавания лиц и распознавание лиц в реальном времени. Эти алгоритмы обеспечивают высокую точность и эффективность, что делает их важными инструментами в современных системах компьютерного зрения. Также был проведен ряд тестов, который показал успешность работы реализованной программы.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была проведена всесторонняя исследовательская работа, направленная на разработку высокоэффективного беспилотного летательного аппарата (БПЛА) для проведения поисково-спасательных операций.

По ходу работы были решены следующие задачи:

– проведен аналитический обзор технической литературы по тематике работы, исследование включало анализ современных тенденций в области беспилотных технологий, изучение различных типов дронов и их применения в реальных ситуациях. Особое внимание уделялось анализу технических характеристик и возможностей существующих систем, а также особенностям их использования в сложных условиях. Этот этап позволил определить направления, в которых должна быть развита проектная работа, и заложить основу для последующих этапов разработки;

– сформулированы требования к разрабатываемому летательному аппарату, подобраны компоненты, а также выбраны среда и средства его разработки, на основе проведенного анализа были сформулированы конкретные требования к разрабатываемому беспилотному летательному аппарату. Эти требования включали в себя технические параметры, функциональные возможности и характеристики, необходимые для эффективного выполнения задач в поисково-спасательных операциях. Были тщательно подобраны компоненты для сборки БПЛА, включая контроллер полета SuceX-E F4 с прошивкой BETAFLIGHTF4, моторы XING-E 2207 2750KV, а также камера Caddx Ratel. В процессе выбора компонентов особое внимание уделялось их надежности и совместимости. Для разработки программного обеспечения и управления БПЛА была выбрана среда разработки VIM, подходящая для интеграции всех компонентов и обеспечения стабильной работы системы.;

– собран беспилотный летательный аппарат, процесс сборки включал в себя как физическую интеграцию всех частей, так и настройку и калибровку системы

для обеспечения её корректной работы. Была проведена детальная настройка контроллера полета для обеспечения стабильности и точности управления БПЛА;

– разработан программный код летательного аппарата, была интегрирована система распознавания и обработки изображений, которая использует обученную модель для обнаружения и идентификации объектов на местности. Это значительно повышает эффективность поисково-спасательных операций, позволяя быстро и точно находить цели.

В результате проделанной работы был создан беспилотный летательный аппарат, способный выполнять сложные задачи в условиях поисково-спасательных операций. Реализация проекта потребовала всестороннего подхода, включающего анализ, проектирование, сборку и тестирование, что позволило создать надежную и функциональную систему.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Актуальность применения беспилотных летательных аппаратов для мониторинга, предупреждения и ликвидации чрезвычайных ситуаций природного и техногенного характера / А.В. Кочегаров, А.В. Петров, А.Б. Плаксицкий, Д.В. Конорев // Современные технологии обеспечения гражданской обороны и ликвидации последствий чрезвычайных ситуаций. - 2016. - №1 (7). - С. 272-275.
2. Пученков, В.И. Использование дронов на складах / В.И. Пученков // Логистика - 2017. - С. 22-24.
3. Крамаренко, А.В. Анализ возможности использования дронов в современном строительстве / А.В. Крамаренко, К.С. Краснова // Наука и образование: новое время. - 2017. - №6. - С. 313-319.
4. Применение микроэлементов при выращивании картофеля – предпосылки использования дронов / В.И. Старовойтов, О.А. Старовойтова, А.А. Манохина, Н.Э. Шабанов, В.А. Чайка // Агроинженерия. - 2021. - №4 (104). – С. 14-15
5. Новгородов, К.И. Использование дронов для мониторинга состояния водных объектов в современном мире / К.И. Новгородов. - 2023. - С. 16-17.
6. Байкин, С.А. Актуальность использования беспилотных летательных аппаратов в Арктике. - 2014. - С. 445-449.
7. COBAN, S. Journal of Aviation // Unmanned Aerial Vehicles (UAVs) According to Engine Type, 2018. -№2 (2). - С. 177-184.
8. Hristov, G.V. A Review of the Characteristics of Modern Unmanned Aerial Vehicles / G.V. Hristov, Z.P. Zahariev, I.H. Beloev // Acta Technologica Agriculturae. - 2016. - С. 33-38.
9. Применение беспилотных летательных аппаратов типа квадрокоптер "DJI inspire 1V 2. 0" при поиске людей на местности / И.Г. Цокурова, И. Г. Денисов, С.А. Самойленко, М.Л. Остапчук // Наука, техника и образование. - 2018. - №11 (52). - С. 43-46.
10. Ахметова, К.И. Как беспилотники помогают при поиске потерявшихся в лесу людей / К.И. Ахметова , Н.А. Ершова // Фундаментальные и прикладные

научные исследования: актуальные вопросы, достижения и инновации . - Волгоград: АМИ, 2022 – С. 136-138

11. Тарасов, Я.О. Квадрокоптеры: проектирование, конструирование и испытания // Известия ТулГУ. Технические науки. - 2023. - №5. - С. 21-27.

12. Ковалев, М. А. Беспилотные летательные аппараты вертикального взлета: сборка, настройка и программирование : учеб. пособие / М. А. Ковалев, Д. Н. Овакимян. - Самара : Изд-во Самар. ун-та, 2023. – 108с.

13. Hsieh, C. -T. Development of a Low Cost and Raspberry-based Thermal Imaging System for Monitoring Human Body Temperature / C.-T. Hsieh // 16th International Microsystems, Packaging, Assembly and Circuits Technology Conference. - Taipei: IEEEExplore, 2021. - С. 248-251.

14. Аникин А. Обзор современных технологий беспроводной передачи данных. / А. Аникин // БЕСПРОВОДНЫЕ ТЕХНОЛОГИИ. - 2017. №4 (11). - С. 16-17.

15. Мухтаров, Д.Е. Передача сигнала Wi-Fi на большие расстояния / Д.Е. Мухтаров // Молодые ученые россии. 2021. – С. 26-29.

16. Безбогов, А.А. Безопасность операционных систем : учебное пособие / А.А. Безбогов, А.В. Яковлев, Ю.Ф. Мартемьянов. в. – М. : "Издательство Машиностроение-1", 2007. – 220 с.

17. Резников, Ф.А. 3 в 1. Операционная система Ubuntu Linux 9.04 + полный дистрибутив Ubuntu + 10 операционных систем Linux (+ DVD-ROM) / Ф.А. Резников, В.Б. Комягин. - М.: Триумф, 2010. - 208 с.

18. Фостер, Дж. Защита от взлома: сокетты, эксплойты, shell-код: выявление уязвимостей операционных систем и прикладных программ к атакам хакеров / Дж. Фостер. - М.: Книга по Требованию, 2006. - 784 с.

19. Documentation Raspberry Pi. – <https://www.raspberrypi.com/documentation/>.

20. Валади, Дж. 100% самоучитель Linux / Дж. Валади. - М.: Технолоджи-3000, 2006. - 336 с.

21. Neil, D. Practical Vim : edit text at the speed of thought / D. Neil. - Dallas: The Pragmatic Bookshelf, 2015. - 346 с.

22. Johnson, B. Professional Visual Studio 2012 / B. Johnson. - Indianapolis: John Wiley & Sons, 2013. - 1060 c.
23. Quick start guide. – <https://www.jetbrains.com/help/pycharm/quick-start-guide.html/>.
24. Atom Documentation. – <https://atom-editor.cc/docs/>.
25. Baishakhi, R. A large-scale study of programming languages and code quality in GitHub / R. Baishakhi, D. Posnett, P. Devanbu // Commun. - ACM, 2017. - C. 91–100.

ПРИЛОЖЕНИЯ

Приложение А

Листинг А.1 - Исходный код алгоритма создания датасета

```
import cv2 # модуль для компьютерного зрения
import os # модуль для работы с файловой системой
from PIL import Image # модуль для обработки изображений
import numpy as np # модуль для работы с массивами

# Определяем функцию для обучения распознавателя лиц
def trainer():
    # Инициализируем пустые списки для хранения вырезанных лиц и их
    # идентификаторов
    faceArr, ids = [], []

    # Создаем объект распознавателя лиц на основе LBPH (Local Binary Patterns
    # Histograms)
    recognizer = cv2.face.LBPHFaceRecognizer_create()

    # Загружаем каскад Хаара для обнаружения лиц
    cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

    # Получаем список всех файлов в папке 'dataset'
    imagePath = [os.path.join('dataset', f) for f in os.listdir('dataset')]

    # Проходимся по каждому пути к файлу в списке
    for paths in imagePath:
        # Проверяем, что файл является изображением в формате JPEG
        if os.path.splitext(paths)[-1].split('.')[1] != 'jpg':
            continue # если файл не JPEG, пропускаем его

        # Открываем изображение и конвертируем его в градации серого
        img = Image.open(paths).convert('L')

        # Преобразуем изображение в массив numpy
        imgArr = np.array(img, 'uint8')

        # Получаем идентификатор лица из имени файла (предполагаем, что это вторая
        # часть имени файла)
        faceId = int(os.path.splitext(paths)[-1].split('.')[1])

        # Обнаруживаем лица на изображении
        face = cascade.detectMultiScale(imgArr)

        # Проходимся по всем обнаруженным лицам
        for (x, y, w, h) in face:
            # Вырезаем область лица и добавляем в список лиц
            faceArr.append(imgArr[y:y+h, x:x+w])
            # Добавляем идентификатор лица в список идентификаторов
            ids.append(faceId)

    # Тренируем распознаватель лиц на собранных данных
    recognizer.train(faceArr, np.array(ids))
    # Сохраняем обученную модель в файл 'trainer.yml'
    recognizer.save('trainer.yml')
```

Приложение Б

Листинг Б.1 - Исходный код распознавания лиц

```
import cv2 # модуль для работы с компьютерным зрением
import json # модуль для работы с JSON файлами
from spreadsheet import * # импортируем все из модуля spreadsheet
(предположительно для работы с таблицами)

# Загружаем классификатор для обнаружения лиц (используется каскад Хаара)
cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Открываем доступ к камере (0 - это индекс основной камеры)
camera = cv2.VideoCapture(0)

# Создаем объект распознавателя лиц на основе LBPH
recognizer = cv2.face.LBPHFaceRecognizer_create()

# Загружаем предварительно обученную модель для распознавания лиц
recognizer.read('trainer.yml')

# Загружаем информацию о пользователях из JSON файла
with open('users.json') as jsonFile:
    users = json.load(jsonFile)

# Инициализируем пустой список для хранения распознанных пользователей
userList = []

# Запускаем бесконечный цикл для постоянного захвата кадров с камеры
while True:
    # Читаем кадр с камеры
    ret, image = camera.read()

    # Преобразуем кадр в градации серого
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Обнаруживаем лица на изображении
    face = cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5,
    minSize=(30, 30))

    # Проходимся по всем обнаруженным лицам
    for (x, y, w, h) in face:
        # Рисуем прямоугольник вокруг обнаруженного лица
        cv2.rectangle(image, (x, y), (x + w, y + h), (100, 0, 100), 2)

        # Прогнозируем идентификатор лица и процент совпадения
        faceId, percentage = recognizer.predict(gray[y:y+h, x:x+w])

        # Если процент совпадения меньше 50, считаем, что лицо распознано
        if percentage < 50:
            # Добавляем идентификатор лица в список распознанных пользователей
            userList.append(faceId)
            # Получаем имя пользователя из JSON файла и формируем строку для
            отображения
            faceId = users[str(faceId)][ 'name' ] + ' ' + str(round(100 -
            percentage, 2)) + '%'
        else:
            # Если лицо не распознано, помечаем его как 'Unknown'
            faceId = 'Unknown'

    # Наносим текст (имя пользователя или 'Unknown') на изображение
```

```

        cv2.putText(image, faceId, (x, y + h), cv2.FONT_HERSHEY_SIMPLEX, 1, (50,
255, 50), 2)

# Отображаем изображение с наложенными данными в окне
cv2.imshow('Image', image)

# Если нажата клавиша 'q', выходим из цикла
if cv2.waitKey(20) & 0xFF == ord('q'):
    break

# Освобождаем ресурсы: закрываем доступ к камере и уничтожаем все окна OpenCV
camera.release()
cv2.destroyAllWindows()

# Удаляем дублирующиеся идентификаторы из списка пользователей
userList = list(set(userList))

# Если список распознанных пользователей не пуст, добавляем их в таблицу
if userList:
    addToSpreadsheet(users, userList)mpport json

```


Приложение В

Листинг В.1 - Исходный код добавления лиц в базу данных

```
import cv2 # модуль для работы с компьютерным зрением
from trainer import trainer # импортируем функцию trainer для обучения модели
распознавания лиц
import json # модуль для работы с JSON файлами

# Открываем доступ к камере (индекс 0 указывает на основную камеру)
camera = cv2.VideoCapture(0)

# Загружаем классификатор Хаара для обнаружения лиц
cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Создаем объект распознавателя лиц на основе алгоритма LBPH
recognizer = cv2.face.LBPHFaceRecognizer_create()

# Запрашиваем у пользователя имя и идентификатор
name = input('Enter Name: ') # Ввод имени пользователя
Id = int(input('Enter Id: ')) # Ввод идентификатора пользователя

# Инициализируем счетчик итераций для захвата изображений
itr = 1

# Запускаем бесконечный цикл для захвата изображений с камеры
while True:

    # Читаем кадр с камеры
    ret, image = camera.read()

    # Преобразуем изображение в градации серого
    grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Обнаруживаем лица на изображении
    face = cascade.detectMultiScale(grayImage,
                                    scaleFactor=1.5, # коэффициент
масштабирования для улучшения точности обнаружения
                                    minNeighbors=5, # количество соседних
прямоугольников для сохранения объекта
                                    minSize=(30, 30)) # минимальный размер лица

    # Проходимся по всем обнаруженным лицам
    for (x, y, w, h) in face:
        # Рисуем прямоугольник вокруг обнаруженного лица
        cv2.rectangle(image, (x, y), (x + w, y + h), (100, 100, 0), 2)

        # Сохраняем изображение лица в папку 'dataset' с уникальным именем файла
        cv2.imwrite('dataset/User.' + str(Id) + '.' + str(itr) + '.jpg',
                    grayImage[y:y + h + 5, x:x + w + 5])

    # Увеличиваем счетчик итераций
    itr += 1

    # Прерываем цикл после захвата 20 изображений
    if itr == 21:
        break

    # Прерываем цикл, если нажата клавиша 'q'
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
```

```
# Отображаем текущий кадр с наложенными прямоугольниками
cv2.imshow('Frame', image)

# Освобождаем ресурсы: закрываем доступ к камере и уничтожаем все окна OpenCV
camera.release()
cv2.destroyAllWindows()

# Открываем JSON файл для чтения информации о пользователях
with open('users.json') as json_file:
    data = json.load(json_file)

# Создаем запись с именем пользователя
username = {
    'name': name
}

# Добавляем или обновляем информацию о пользователе в JSON файле
data[Id] = username

# Сохраняем обновленные данные обратно в JSON файл
with open("users.json", "w") as file:
    json.dump(data, file, indent=4) # записываем данные с отступами для удобства
чтения

# Вызываем функцию для обучения модели распознавания лиц на новых данных
trainer()
```