

Минобрнауки России  
Юго-Западный государственный университет

Кафедра программной инженерии

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**ПО ПРОГРАММЕ БАКАЛАВРИАТА**

09.03.04 Программная инженерия

(код, наименование ОПОП ВО: направление подготовки, направленность (профиль))

«Разработка программно-информационных систем»

Интеллектуальная система распознавания и классификации возгораний,  
полученных с БПЛА

(название темы)

Дипломный проект

(вид ВКР: дипломная работа или дипломный проект)

Автор ВКР

Д. А. Каракчиев

(подпись, дата)

(инициалы, фамилия)

Группа ПО-026

Руководитель ВКР

Р. А. Томакова

(подпись, дата)

(инициалы, фамилия)

Нормоконтроль

А. А. Чаплыгин

(подпись, дата)

(инициалы, фамилия)

ВКР допущена к защите:

Заведующий кафедрой

А. В. Малышев

(подпись, дата)

(инициалы, фамилия)

Курск 2024 г.

**Минобрнауки России**  
**Юго-Западный государственный университет**

Кафедра программной инженерии

УТВЕРЖДАЮ:  
Заведующий кафедрой

---

(подпись, инициалы, фамилия)

«\_\_\_\_\_» 20\_\_\_\_ г.

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ  
РАБОТУ ПО ПРОГРАММЕ БАКАЛАВРИАТА**

Студента Каракчиева Д.А., шифр 20-06-0391, группа ПО-026

1. Тема «Интеллектуальная система распознавания и классификации возгораний, полученных с БПЛА» утверждена приказом ректора ЮЗГУ от «04» апреля 2024 г. № 1616-с.

2. Срок предоставления работы к защите «11» июня 2024 г.

3. Исходные данные для создания программной системы:

3.1. Перечень решаемых задач:

1) определить функциональные и технические требования разрабатываемого приложения;

2) разработать концептуальную модель системы поиска и классификации возгораний на данных, полученных с БПЛА;

3) спроектировать программную систему поиска и классификации возгораний;

4) сконструировать и протестировать программную систему поиска и классификации возгораний.

3.2. Входные данные и требуемые результаты для программы:

1) Входными данными для программной системы являются: данные справочников комплектующих, конфигураций, ПО ; данные изображения,

полученные с БПЛА, на которых предположительно могут быть зафиксированы участки возгорания; данные изображения, отформатированные в отдельных приложениях; модель нейронной сети по локализации возгораний.

2) Выходными данными для программной системы являются: изображения, с выделенными рамками вокруг участков возгорания и вероятность их наличия; сведения о выполненных анализах изображений, включая вероятность и точность обнаружения возгораний; выходные отчёты (инфографика) – по результатам анализа изображений, по эффективности обнаружения возгораний, по работе системы в целом; отчет с информацией о изображении; сведения о точности ошибки в тензоре нейронной сети.

#### 4. Содержание работы (по разделам):

##### 4.1. Введение

##### 4.1. Анализ предметной области

4.2. Техническое задание: основание для разработки, назначение разработки, требования к программной системе, требования к оформлению документации.

4.3. Технический проект: общие сведения о программной системе, проект данных программной системы, проектирование архитектуры программной системы, проектирование пользовательского интерфейса программной системы.

4.4. Рабочий проект: спецификация компонентов и классов программной системы, тестирование программной системы, сборка компонентов программной системы.

##### 4.5. Заключение

##### 4.6. Список использованных источников

#### 5. Перечень графического материала:

Лист 1. Сведения о ВКРБ

Лист 2. Цель и задачи разработки

Лист 3. Концептуальная модель сайта

Лист 4. Еще плакат

Руководитель ВКР

Р. А. Томакова

(подпись, дата)

(инициалы, фамилия)

Задание принял к исполнению

Д. А. Каракчиев

(подпись, дата)

(инициалы, фамилия)

## РЕФЕРАТ

Объем работы равен 77 страницам. Работа содержит 13 иллюстраций, 6 таблиц, 11 библиографических источников и 4 листа графического материала. Количество приложений – 2. Графический материал представлен в приложении А. Фрагменты исходного кода представлены в приложении Б.

Перечень ключевых слов: коммерческий сайт, Система, CMS, Битрикс, Joomla, аддитивные технологии, 3D-принтеры, услуги, сервисы, информатизация, автоматизация, информационные технологии, веб-форма, Apache, классы, база данных, средства защиты информации, подсистема, компонент, модуль, сущность, информационный блок, метод, контент-редактор, администратор, пользователь, web-сайт.

Объектом разработки является web-сайт компании, занимающейся производством 3D-принтеров, выпуском оборудования для создания порошков, разработкой программного обеспечения и организацией центров аддитивного производства.

Целью выпускной квалификационной работы является привлечение клиентов, увеличение заказов, информирование о продукции и услугах путем создания сайта компании.

В процессе создания сайта были выделены основные сущности путем создания информационных блоков, использованы классы и методы модулей, обеспечивающие работу с сущностями предметной области, а также корректную работу web-сайта, разработаны разделы, содержащие информацию о компании, ее деятельности, производимой продукции и услугах, разработан сервис по заказу 3D-деталей.

При разработке сайта использовалась система управления контентом «1С-Битрикс: Управление сайтом».

Разработанный сайт был успешно внедрен в компании.

## ABSTRACT

The volume of work is 77 pages. The work contains 13 illustrations, 6 tables, 11 bibliographic sources and 4 sheets of graphic material. The number of applications is 2. The graphic material is presented in annex A. The layout of the site, including the connection of components, is presented in annex B.

List of keywords: commercial website, System, CMS, Bitrix, Joomla, additive technologies, 3D printers, services, services, informatization, automation, information technology, web form, Apache, classes, database, component, module, entity, information block, method, content editor, administrator, user, web site.

The object of the research is the analysis of information technologies for the development of a production company's website.

The object of the development is the website of a company engaged in the production of 3D printers, the production of equipment for the creation of powders, software development and the organization of additive manufacturing centers.

The purpose of the final qualifying work is to attract customers, increase orders, inform about products and services by creating a company website.

In the process of creating the site, the main entities were identified by creating information blocks, classes and methods of modules were used to ensure work with the entities of the subject area, as well as the correct operation of the website, sections containing information about the company, its activities, products and services were developed, a service for ordering 3D parts was developed.

When developing the site, the content management system «1C – Bitrix: Site Management» was used.

The developed website was successfully implemented in the company.

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>11</b>
1 Анализ предметной области	14
1.1 Структура и Технические Характеристики БПЛА	14
1.1.1 Техническое устройство БПЛА	14
1.1.2 Компоненты и подсистемы БПЛА	15
1.1.3 Разнообразие моделей БПЛА и их применение в обнаружении возгораний	15
1.1.4 Работа камер и сенсоров на БПЛА для обнаружения возгораний	17
1.1.4.1 Типы камер и их характеристики	17
1.1.4.2 Использование датчиков тепла и газов	17
1.1.4.3 Обработка данных в реальном времени	18
1.2 Виды Пожаров и Методы Обнаружения	18
1.2.1 Типы возгораний и их особенности	18
1.2.2 Сравнительный анализ методов обнаружения пожаров	19
1.2.3 Технические проблемы и сложности при обнаружении возгораний с помощью БПЛА	20
1.3 Применение технологий для предотвращения пожаров	21
1.3.1 Использование данных об обнаруженных пожарах для оперативного реагирования и предотвращения катастроф.	21
1.3.2 Роль БПЛА в операциях по тушению пожаров и организации спасательных мероприятий	21
1.3.3 Влияние автоматизированных систем обнаружения на эффективность противопожарных операций	22
1.4 Инновации в Технологиях Пожарного Дронирования	23
1.4.1 Прогрессивные методы классификации и локализации возгораний с применением нейронных сетей	23
1.4.2 Перспективы применения беспилотных массивов дронов для комплексного контроля за пожарами и оценки ущерба	23
1.4.3 Непосредственное использование бпла для тушения пожаров	25

2 Техническое задание	27
2.1 Основание для разработки	27
2.2 Цель и назначение разработки	27
2.3 Требования пользователя к интерфейсу приложения	27
2.4 Моделирование вариантов использования	29
2.5 Требования к оформлению документации	31
3 Технический проект	32
3.1 Общая характеристика организации решения задачи	32
3.2 Обоснование выбора технологии проектирования	32
3.2.1 Описание используемых технологий и языков программирования	32
3.2.2 Сверточные нейронные сети	32
3.2.3 Машинное обучение	33
3.2.4 Язык программирования Python	33
3.2.5 Библиотеки Python	34
3.2.6 Архитектура сверточной нейронной сети	35
3.2.7 Функция активации ReLU	36
3.2.8 Функция IoU Loss	37
3.3 Диаграмма компонентов	39
3.3.1 Взаимодействие компонентов	40
3.3.2 Диаграмма программных классов	40
3.4 Проектирование пользовательского интерфейса	42
4 Рабочий проект	44
4.1 Классы, используемые при разработке приложения	44
4.1.1 Класс main	44
4.1.2 Класс dataprocessing	49
4.1.3 Класс creatingtfrecordclassifier	51
4.1.4 Класс creatingtfrecordlocalizer	52
4.1.5 Класс classifier	55
4.1.6 Класс training	56
4.2 Модульное тестирование разработанного web-сайта	57

4.3 Системное тестирование разработанного web-сайта	57
<b>ЗАКЛЮЧЕНИЕ</b>	60
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	60
ПРИЛОЖЕНИЕ А Представление графического материала	63
ПРИЛОЖЕНИЕ Б Фрагменты исходного кода программы	68
На отдельных листах (CD-RW в прикрепленном конверте)	77
Сведения о ВКРБ (Графический материал / Сведения о ВКРБ.png)	Лист 1
Цель и задачи разработки (Графический материал / Цель и задачи разработки.png)	Лист 2
Концептуальная модель сайта (Графический материал / Концептуальная модель сайта.png)	Лист 3
Еще плакат (Графический материал / Еще плакат.png)	Лист 4

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

НС – нейронная сеть.

ИС – информационная система.

ИТ – информационные технологии.

КТС – комплекс технических средств.

ПО – программное обеспечение.

РП – рабочий проект.

БПЛА – беспилотный летательный аппарат.

ТЗ – техническое задание.

ТП – технический проект.

UML (Unified Modelling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения.

## ВВЕДЕНИЕ

Аддитивные технологии (АТ) начали активно развиваться со времени получения первых трехмерных изображений изделий на дисплеях компьютеров. Начало положила стереолитография, затем довольно многочисленные новые принципы стали называть технологиями быстрого прототипирования, затем укоренилось название «Аддитивные технологии». Интенсивность развития данных технологий не имеет аналогов. АТ изменили процессы проектирования и конструирования изделий, превратив их в процессы непрерывного создания изделий. Современные проектирование и производство изделий невозможно представить без данного рода технологий. 3D-принтеры стали такими же распространенными, как и персональные компьютеры. С помощью 3D-принтеров получают ткани, обувь, продукты питания, а также выращивают человеческие органы. Во многих отраслях, например, в космической отрасли, альтернативы аддитивным технологиям нет.

АТ предполагают изготовление детали методом послойного нанесения материала, в отличие от традиционных методов формирования детали, за счёт удаления материала из массива заготовки.

При использовании АТ все стадии реализации проекта от идеи до материализации находятся в единой технологической цепи, в которой каждая технологическая операция выполняется в цифровой CAD/CAM/CAE-системе.

Современные компании, видя, как развиваются информационные технологии, пытаются использовать их выгодно для своего бизнеса, поэтому запускают свой web-сайт. С его помощью предприятие может заявить о себе, проинформировать потенциального заказчика об услугах или продуктах, которые предоставляет, а также позволяет пользователям сделать с помощью сайта онлайн-заказ, произвести покупку или оплатить счета.

Сайт считается лицом компании и может существенно повысить ее имидж. Любой пользователь сети Интернет сможет получить необходимую информацию о компании в любой момент, появляется возможность найти контактные телефоны, адрес и e-mail, чтобы связаться с компанией. Сейчас

большинство клиентов узнают о ее существовании именно через сайт. Поэтому сайт можно назвать самой лучшей рекламой.

Главной задачей профессионально построенного сайта является превращение посетителя, зашедшего на сайт, в потенциального клиента.

*Цель настоящей работы* – разработка web-сайта компании для привлечения новой аудитории, увеличения заказов, рекламы продукции и услуг компаний. Для достижения поставленной цели необходимо решить *следующие задачи*:

- провести анализ предметной области;
- разработать концептуальную модель web-сайта;
- спроектировать web-сайт;
- реализовать сайт средствами web-технологий.

*Структура и объем работы.* Отчет состоит из введения, 4 разделов основной части, заключения, списка использованных источников, 2 приложений. Текст выпускной квалификационной работы равен 12 страницам.

*Во введении* сформулирована цель работы, поставлены задачи разработки, описана структура работы, приведено краткое содержание каждого из разделов.

*В первом разделе* на стадии описания технической характеристики предметной области приводится сбор информации о деятельности компании, для которой осуществляется разработка сайта.

*В втором разделе* на стадии технического задания приводятся требования к разрабатываемому сайту.

*В третьем разделе* на стадии технического проектирования представлены проектные решения для web-сайта.

*В четвертом разделе* приводится список классов и их методов, использованных при разработке сайта, производится тестирование разработанного сайта.

В заключении излагаются основные результаты работы, полученные в ходе разработки.

В приложении А представлен графический материал. В приложении Б представлены фрагменты исходного кода.

# **1 Анализ предметной области**

## **1.1 Структура и Технические Характеристики БПЛА**

### **1.1.1 Техническое устройство БПЛА**

Беспилотные летательные аппараты (БПЛА), известные также как дроны, представляют собой высокотехнологичные устройства, которые находят разнообразное применение от военных операций до сельскохозяйственного мониторинга. Они могут быть сконструированы в различных формах, включая фиксированные крылья для дальних и высотных полетов или многороторные системы для более гибкого управления и вертикального взлета и посадки.

Корпус БПЛА обычно изготавливается из легких, но прочных материалов, таких как углеродное волокно, что обеспечивает оптимальное сочетание прочности и веса. Аэродинамический дизайн корпуса способствует уменьшению сопротивления воздуха, что увеличивает эффективность полета.

Силовая установка БПЛА может варьироваться от электрических моторов, которые обеспечивают тихий и экологичный полет, до бензиновых двигателей, предлагающих большую мощность и продолжительность полета. Современные БПЛА оснащены сложными системами управления, которые включают в себя автопилот, различные датчики для навигации и стабилизации, а также GPS для точного позиционирования.

Полезная нагрузка БПЛА может включать высококачественные камеры для фотографии и видеосъемки, различные датчики для сбора данных и специализированное оборудование для выполнения конкретных задач. Источники питания, такие как аккумуляторы или солнечные панели, обеспечивают энергией все системы БПЛА.

Коммуникационные системы играют ключевую роль в безопасности и эффективности полетов БПЛА, обеспечивая надежную связь между дроном и оператором. Системы безопасности, включая аварийное возвращение на

базу и парашюты, гарантируют, что БПЛА может быть безопасно возвращен в случае нештатных ситуаций.

### **1.1.2 Компоненты и подсистемы БПЛА**

Компоненты БПЛА играют критически важную роль в их функционировании и эффективности. Здесь представлены наиболее значимые компоненты дронов:

- Автопилот и системы управления: Эти системы являются мозгом БПЛА, обеспечивая автоматическое управление полетом. Они включают в себя микропроцессоры, программное обеспечение для управления полетом, а также датчики для стабилизации и навигации. Автопилот может выполнять задачи, такие как взлет, полет по заданным точкам, обход препятствий и посадка.

- Системы наблюдения и датчики: Эти системы собирают данные с помощью различных датчиков, таких как камеры, инфракрасные датчики, лидары и радары. Они используются для картографирования, сбора геоданных, наблюдения и других задач, требующих визуализации или измерения

- Коммуникационное оборудование: Включает в себя радиопередатчики и приемники для обмена данными между БПЛА и оператором. Это обеспечивает передачу телеметрии, видео и управляющих команд в реальном времени. Также могут использоваться спутниковые системы связи для дальних полетов и обеспечения связи вне зоны прямой видимости.

### **1.1.3 Разнообразие моделей БПЛА и их применение в обнаружении возгораний**

Беспилотные аппараты представляют собой широкий спектр аeronавтических систем, каждая из которых обладает уникальными характеристиками и способностями, делающими их подходящими для различных задач и миссий. В контексте обнаружения возгораний, БПЛА становятся неоценимым инструментом благодаря их способности быстро и эффективно собирать данные с высокой точностью.

Модели БПЛА варьируются от небольших квадрокоптеров до крупных фиксированных крыльев, каждая из которых имеет свои преимущества. Например, квадрокоптеры могут зависать в воздухе и маневрировать в ограниченных пространствах, что делает их идеальными для детального изучения определенных участков. С другой стороны, БПЛА с фиксированными крыльями способны на длительные полеты на большие расстояния, что позволяет им покрывать обширные территории при поиске признаков возгорания.

Примеры разных видов БПЛА, широко используемых для тушения пожаров и возгораний:

- DJI Phantom 4 Pro: Этот квадрокоптер широко используется для фотографии и видеосъемки благодаря своей стабильности в полете и высококачественной камере. Он также может быть адаптирован для мониторинга и обнаружения пожаров с помощью дополнительных датчиков.

- Parrot Anafi: Этот компактный дрон оснащен тепловизионной камерой, что делает его подходящим для поиска тепловых подписей в рамках задач по обнаружению пожаров. Его легкость и мобильность позволяют быстро развертывать его на местности.

- General Atomics MQ-9 Reaper: Это беспилотное летательное средство с фиксированным крылом, которое первоначально разрабатывалось для выполнения военных задач, но также может использоваться для длительного мониторинга больших территорий, что делает его полезным для обнаружения пожаров.

- Insitu ScanEagle: Этот небольшой, но мощный БПЛА с фиксированным крылом способен проводить полеты продолжительностью до 24 часов, что делает его идеальным для непрерывного наблюдения за территориями в целях предотвращения пожаров.

- Firefly6 Pro: Этот БПЛА оснащен инфракрасными камерами для обнаружения очагов возгорания и системами для доставки огнетушащих средств. Он способен к вертикальному взлету и посадке, что позволяет ему оперативно действовать в сложных условиях и на ограниченных площадках, где традиционные средства не могут быть использованы.

Использование БПЛА в обнаружении возгораний включает в себя не только непосредственный поиск огня, но и анализ температурных аномалий, изменений в растительности и других индикаторов, которые могут указывать на потенциальную угрозу пожара. Современные БПЛА оснащены различными датчиками, включая тепловизионные камеры и датчики для анализа спектральных данных, что позволяет им обнаруживать возгорания на ранних стадиях, когда они еще могут быть локализованы и потушены с минимальными усилиями.

#### **1.1.4 Работа камер и сенсоров на БПЛА для обнаружения возгораний**

##### **1.1.4.1 Типы камер и их характеристики**

- Оптические камеры: Обеспечивают высокое разрешение и детализацию изображений, что позволяет операторам видеть мелкие детали на земле. Они могут быть оснащены зумом для увеличения участков интереса.
- Инфракрасные камеры: Позволяют обнаруживать тепло, исходящее от объектов, что особенно полезно в условиях низкой видимости или ночью. Они могут выявлять тепловые подписи возгораний, даже если они не видны в оптическом диапазоне.
- Мультиспектральные камеры: Сочетают в себе несколько типов датчиков для сбора данных в различных диапазонах спектра. Это позволяет анализировать растительность и обнаруживать изменения, которые могут указывать на риск возгорания.

##### **1.1.4.2 Использование датчиков тепла и газов**

- Тепловые датчики: Обнаруживают повышенные температуры, что может быть признаком начидающегося пожара. Они могут быть настроены на определенные пороговые значения для автоматического оповещения.

- Газовые датчики: Способны обнаруживать наличие газов, таких как углекислый газ или метан, которые могут выделяться при горении. Это помогает в раннем обнаружении пожаров.

#### **1.1.4.3 Обработка данных в реальном времени**

- Аналитическое программное обеспечение: Интегрировано с БПЛА для анализа собранных данных на лету. Это позволяет операторам быстро реагировать на изменения и принимать решения.
- Коммуникационные системы: Обеспечивают передачу данных с БПЛА на землю в реальном времени, что позволяет командам на земле координировать действия по борьбе с пожарами.

### **1.2 Виды Пожаров и Методы Обнаружения**

#### **1.2.1 Типы возгораний и их особенности**

Лесные пожары, пожары на промышленных объектах и городские пожары – это три основных типа возгораний, каждый из которых имеет свои уникальные характеристики и требует специфического подхода к тушению и предотвращению.

Лесные пожары часто возникают в результате естественных процессов, таких как удары молний, но также могут быть вызваны человеческой деятельностью. Они распространяются быстро, усугубляемые сухой растительностью, ветром и топографическими условиями. Лесные пожары могут охватывать огромные территории и вызывать значительный экологический и экономический ущерб. Они также могут привести к потере биоразнообразия и эрозии почвы.

Пожары на промышленных объектах представляют собой особую опасность из-за наличия взрывоопасных и токсичных материалов. Такие пожары могут возникать в результате технологических нарушений, несоблюдения правил безопасности или аварий. Они требуют быстрого и профессионального реагирования, поскольку последствия могут включать не только уничто-

жение имущества, но и серьезные риски для здоровья и безопасности людей, а также для окружающей среды.

Городские пожары могут возникать в жилых и коммерческих зданиях и часто связаны с неисправной электропроводкой, неосторожным обращением с огнем или умышленными поджогами. Они могут быстро распространяться между зданиями, особенно в плотно застроенных районах, и требуют немедленного вмешательства пожарных служб. Городские пожары также представляют угрозу для жизни людей и могут привести к значительным материальным потерям.

### **1.2.2 Сравнительный анализ методов обнаружения пожаров**

Сравнительный анализ методов обнаружения пожаров включает в себя оценку различных технологий и подходов, используемых для раннего выявления и предупреждения о пожарах. Основные методы включают использование дымовых датчиков, тепловых датчиков, инфракрасных камер и систем видеонаблюдения.

Дымовые датчики являются наиболее распространенным и доступным средством, обнаружающим частицы дыма в воздухе. Тепловые датчики реагируют на повышение температуры, что может указывать на наличие пожара. Инфракрасные камеры и системы видеонаблюдения позволяют оперативно обнаруживать источники тепла и пламени, особенно в условиях плохой видимости.

Каждый из этих методов имеет свои преимущества и недостатки. Например, дымовые датчики могут быстро срабатывать на дым, но они не всегда эффективны в открытых или хорошо проветриваемых пространствах. Тепловые датчики могут не сработать, если пожар возник вне их диапазона действия. Инфракрасные камеры и системы видеонаблюдения требуют сложной калибровки и могут быть дорогими в установке и обслуживании.

В итоге, выбор метода обнаружения пожаров зависит от конкретных условий и требований к безопасности. Важно провести тщательный анализ потенциальных рисков и определить наиболее подходящую систему для каж-

дого конкретного случая. Современные технологии также предлагают интегрированные решения, сочетающие различные методы обнаружения для повышения надежности и эффективности системы предупреждения о пожарах.

### **1.2.3 Технические проблемы и сложности при обнаружении возгораний с помощью БПЛА**

Одной из основных задач является обеспечение стабильности и точности полета БПЛА в различных погодных условиях. Сильный ветер, дождь и другие атмосферные явления могут существенно повлиять на управляемость и эффективность работы БПЛА.

Кроме того, необходимо точно калибровать оптические и инфракрасные камеры, чтобы они могли эффективно обнаруживать признаки возгорания на больших расстояниях и в различных условиях освещенности. Это требует сложных алгоритмов обработки изображений и может быть затруднено в случае, если на местности присутствуют другие источники тепла.

Дальность и время полета БПЛА также ограничены их энергетическими возможностями. Необходимо регулярно подзаряжать или менять аккумуляторы, что может быть проблематично в удаленных или труднодоступных районах.

Обеспечение безопасности полетов БПЛА и предотвращение столкновений с другими летательными аппаратами является еще одной важной задачей. Для этого требуется интеграция с системами воздушного контроля и соблюдение строгих правил использования воздушного пространства.

Наконец, обработка и анализ большого объема данных, собранных БПЛА, требует мощных вычислительных ресурсов и специализированного программного обеспечения, что также может быть сложностью, особенно в условиях реального времени.

## **1.3 Применение технологий для предотвращения пожаров**

### **1.3.1 Использование данных об обнаруженных пожарах для оперативного реагирования и предотвращения катастроф.**

С помощью беспилотных летательных аппаратов возможно быстро оценить масштабы возгорания, определить его точное местоположение и направление распространения огня. Это позволяет спасательным службам эффективно распределять ресурсы и направлять пожарные команды туда, где они наиболее нужны.

Инфракрасные камеры на БПЛА способны выявлять очаги возгорания, которые невидимы для человеческого глаза, особенно в условиях сильного задымления или ночью. Это дает возможность предотвратить распространение огня на ранней стадии и снизить вероятность возникновения крупномасштабных катастроф.

Кроме того, данные с БПЛА используются для создания точных карт распространения огня, что необходимо для планирования эвакуации населения и определения безопасных маршрутов. Также они помогают в координации действий различных служб, участвующих в тушении пожаров и оказании помощи пострадавшим.

Важным аспектом является и использование данных для анализа причин возникновения пожаров и разработки мер по их предотвращению в будущем. Анализируя информацию о прошлых пожарах, можно выявить наиболее уязвимые участки территории и принять необходимые меры для уменьшения риска возгорания.

### **1.3.2 Роль**

#### **БПЛА в операциях по тушению пожаров и организации спасательных мероприятий**

Благодаря возможности сбора информации в реальном времени, БПЛА обеспечивают командам быстрый доступ к актуальным данным о ситуации

на месте пожара. Это позволяет оперативно принимать решения и адаптировать стратегии тушения в соответствии с меняющимися условиями.

Координация действий различных служб спасения является ключевым элементом успешного тушения пожаров. БПЛА предоставляют командирам на местах и центрам управления операциями точные данные о распространении огня, плотности дыма и возможных опасностях. Это позволяет спасательным службам эффективно распределять ресурсы, направлять пожарные бригады в наиболее нужные точки и обеспечивать безопасность персонала.

Предоставление данных для составления планов тушения также является важной функцией БПЛА. С их помощью можно создавать детализированные карты местности, отслеживать изменения в распространении огня и определять оптимальные маршруты для подхода к очагам возгорания. Эти данные необходимы для разработки стратегий тушения, которые максимально сокращают время на борьбу с огнем и минимизируют риски для жизни и здоровья людей.

### **1.3.3 Влияние автоматизированных систем обнаружения на эффективность противопожарных операций**

Автоматизированные системы обнаружения пожаров оказывают значительное влияние на эффективность противопожарных операций. Они сокращают время, необходимое для обнаружения пожаров, что критически важно для предотвращения их распространения. Благодаря быстрому реагированию на возгорания, возможности для локализации огня и предотвращения его распространения значительно увеличиваются.

Системы автоматического обнаружения обеспечивают увеличение точности определения местоположения пожара, что позволяет спасательным службам быстрее и точнее реагировать на чрезвычайные ситуации. Это приводит к более оперативному принятию решений и эффективному распределению ресурсов, что способствует снижению ущерба от пожаров и сохранению жизней.

## **1.4 Инновации в Технологиях Пожарного Дронирования**

### **1.4.1 Прогрессивные методы классификации и локализации возгораний с применением нейронных сетей**

Современные системы используют сложные алгоритмы для анализа данных с дронов и спутников, что позволяет с высокой точностью определять местоположение и характеристики возгораний. Нейронные сети, обученные на больших объемах данных, способны распознавать различные типы пожаров и предсказывать их поведение.

Как нейронные сети могут быть использованы в этой области:

- Сбор данных: Нейронные сети начинают с анализа больших объемов данных о пожарах, включая изображения и видео, полученные с дронов и спутников.
- Обучение модели: Данные используются для обучения нейронных сетей распознавать различные типы пожаров и их характеристики.
- Классификация пожаров: Обученные модели способны классифицировать пожары по типу, размеру и интенсивности.
- Локализация пожаров: Нейронные сети анализируют геопространственные данные для точного определения местоположения пожаров.
- Прогнозирование поведения огня: С помощью алгоритмов нейронные сети могут предсказывать направление и скорость распространения огня.
- Оценка ущерба: Нейронные сети могут анализировать потенциальный ущерб от пожара, помогая планировать эвакуацию и ресурсное обеспечение.

### **1.4.2 Перспективы применения беспилотных массивов дронов для комплексного контроля за пожарами и оценки ущерба**

Беспилотные массивы дронов открывают новые горизонты в области контроля за пожарами и оценки ущерба. Эти технологии предлагают револю-

ционный подход к мониторингу и реагированию на чрезвычайные ситуации, обеспечивая беспрецедентную оперативность и точность.

Системы дронов способны оперативно собирать данные с различных уголков зоны бедствия, предоставляя операторам полную картину происходящего. Использование множества дронов одновременно позволяет получать объемные данные о температуре, скорости ветра и влажности воздуха, что критически важно для оценки ситуации и принятия решений.

В будущем можно ожидать следующие инновации:

- Автономные дроны-пожарные: Разработка дронов, способных не только обнаруживать пожары, но и самостоятельно проводить первичное тушение, например, с помощью воды или огнетушащих веществ.
- Интеграция с Интернетом вещей (IoT): Соединение дронов с датчиками на зданиях и в лесах для создания единой сети раннего реагирования на пожары.
- Усовершенствованные алгоритмы прогнозирования: Использование глубокого обучения для анализа данных и создания более точных моделей поведения огня, что позволит предсказывать пожары за дни и недели до их возникновения.
- Роботизированные пожарные станции: Автоматизация пожарных станций с помощью ИИ, которые будут координировать действия дронов и наземных роботов-пожарных.
- Системы виртуальной и дополненной реальности: Обучение пожарных с помощью VR и AR, позволяющее имитировать различные сценарии пожаров для повышения эффективности и безопасности тренировок.
- Сетевые операции: Разработка протоколов для координации множества дронов и роботов, работающих вместе в условиях пожара, для оптимизации процесса тушения и снижения рисков для человеческих пожарных.

Эти инновации не только улучшат реагирование на пожары, но и помогут в предотвращении их возникновения, а также в минимизации ущерба и ускорении процесса восстановления после пожаров.

### **1.4.3 Непосредственное использование БПЛА для тушения пожаров**

БПЛА могут быть оснащены датчиками для обнаружения пожаров и системами доставки огнетушащих средств, таких как вода или пена. Они могут быстро достигать труднодоступных мест и выполнять тушение на ранних стадиях пожара, что снижает риски для пожарных и повышает шансы на предотвращение распространения огня.

Преимущества:

- Быстрый отклик: БПЛА могут быть запущены немедленно и достигать места пожара быстрее, чем наземные команды.
- Доступ в труднодоступные места: Они могут летать в районы, недоступные для пожарных машин, например, в горных или заболоченных районах.
- Безопасность персонала: Снижение риска для жизни пожарных, поскольку БПЛА могут выполнять опасные задачи.
- Сбор данных: Возможность сбора ценной информации о пожаре для анализа и планирования тушения.

Недостатки:

- Ограниченнная грузоподъемность: БПЛА могут нести только ограниченное количество огнетушащего средства.
- Время полета: Ограничено время полета из-за емкости аккумуляторов.
- Зависимость от погоды: Плохие погодные условия могут ограничивать использование БПЛА.
- Регулирование: Необходимость соблюдения авиационных правил и регуляций.

Перспективы: Развитие технологий может привести к увеличению грузоподъемности и времени полета БПЛА, а также к улучшению их устойчивости к погодным условиям. Интеграция с искусственным интеллектом может

улучшить способность БПЛА к самостоятельному обнаружению пожаров и принятию решений о тушении.

Выгодно ли это: Определенно, использование БПЛА выгодно, особенно в регионах с частыми лесными пожарами и труднодоступными территориями. Они могут сократить время реагирования и уменьшить ущерб от пожаров, что в долгосрочной перспективе может быть экономически оправданным, несмотря на начальные затраты на разработку и внедрение системы в ту или иную область.

## **2 Техническое задание**

### **2.1 Основание для разработки**

Основанием для разработки является задание на выпускную квалификационную работу бакалавра «Интеллектуальная система распознавания и классификации возгораний, полученных с БПЛА».

### **2.2 Цель и назначение разработки**

Программно-информационная система предназначена для автоматизации процесса обнаружения и классификации возгораний, повышения эффективности мониторинга пожароопасных зон и поддержки принятия информированных решений при реагировании на потенциальные угрозы.

Посредством создания интеллектуальной системы мы стремимся революционизировать процесс обнаружения и реагирования на пожары, а также позволить специалистам быстро и эффективно определять возгорания без лишних затрат.

Задачами данной разработки являются:

- создание информационной базы для выбора нескольких изображений для последовательной классификации;
- предоставление предварительной обработки изображения для распознавания;
- реализация классификации возгораний по типу;
- выявление оценки степени опасности возгорания;
- обучение нейронной сети на подготовленных данных;
- оптимизация параметров сети для достижения максимальной точности распознавания;
- создание удобного и эффективного пользовательского интерфейса.

### **2.3 Требования пользователя к интерфейсу приложения**

Приложение должно включать в себя:

- графический интерфейс пользователя;

- возможность загрузки изображений с БПЛА для их распознавания;
- отображение результата распознавания с указанием класса возгорания и степень опасности, а также уверенность в распознавании;
- возможность загрузки данных обучения для улучшения качества распознавания;
- возможность вывода полученных изображений с распознанным возгоранием для дальнейшего использования.

Композиции шаблонов программы представлены на рисунках 2.1 и 2.2.

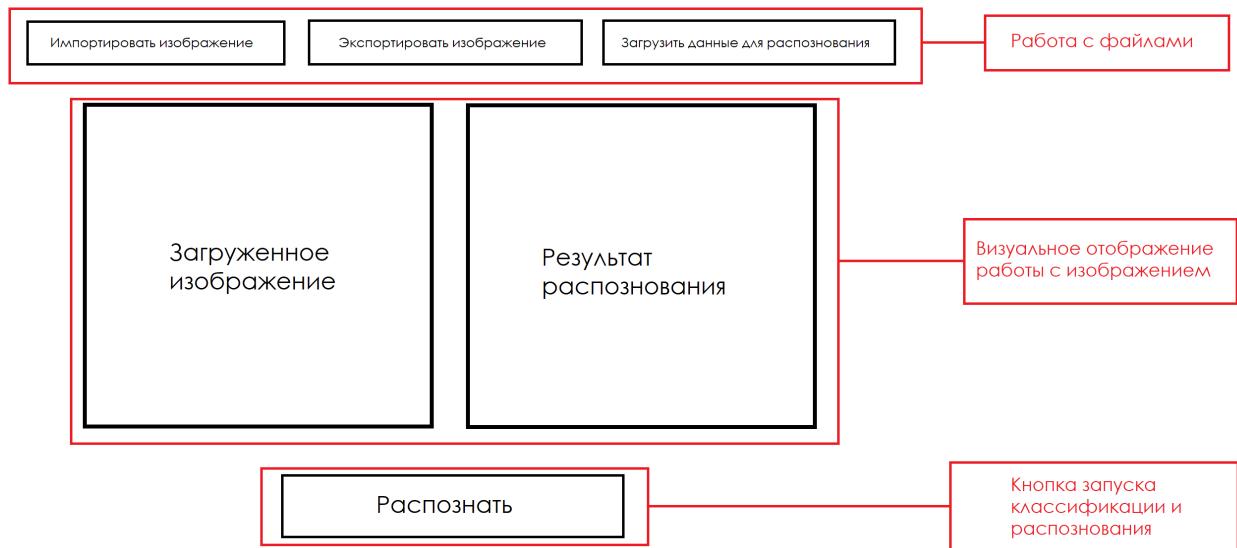


Рисунок 2.1 – Композиция шаблона программы. Окно №1.

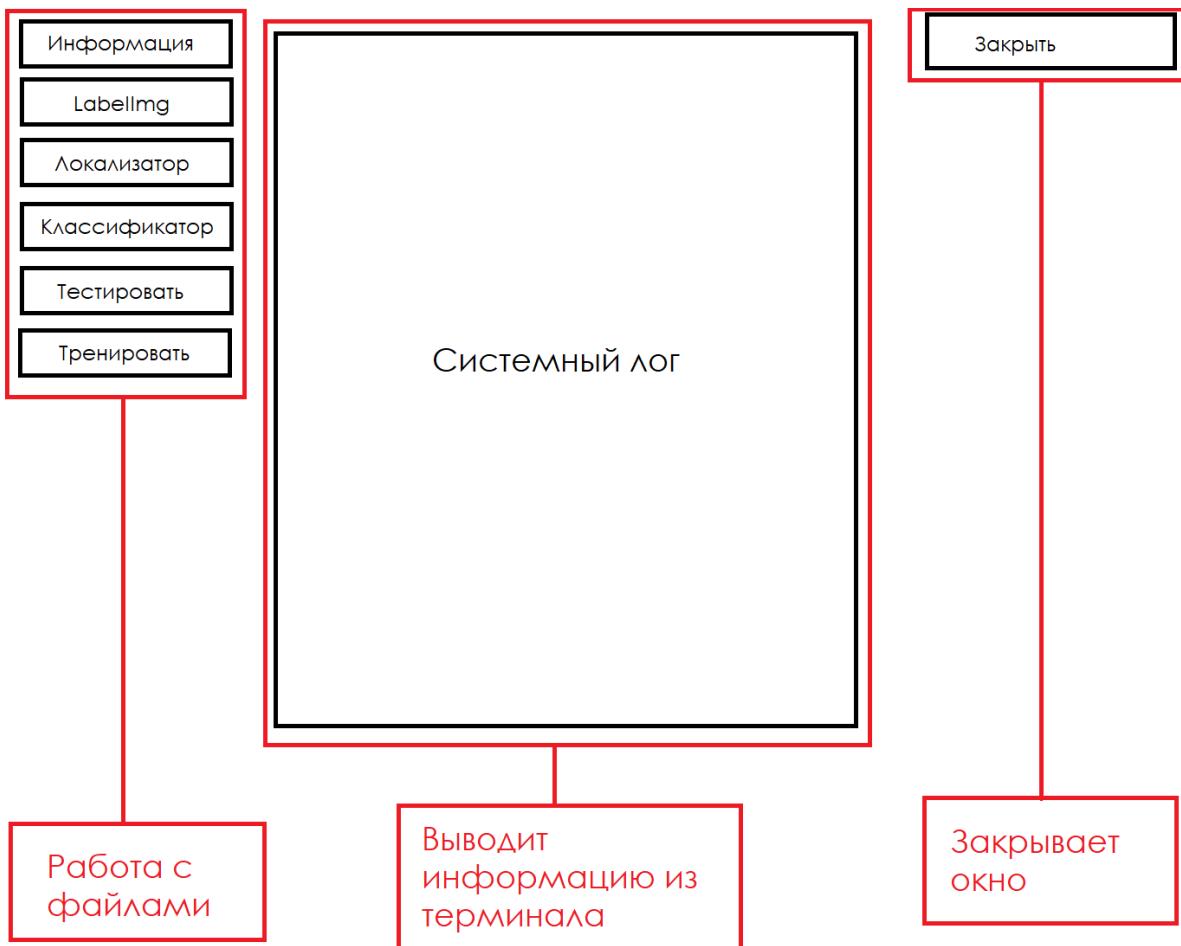


Рисунок 2.2 – Композиция шаблона программы. Окно №2.

## 2.4 Моделирование вариантов использования

Для разрабатываемого сайта была реализована модель, которая обеспечивает наглядное представление вариантов использования приложения.

Она помогает в физической разработке и детальном анализе взаимосвязей объектов. При построении диаграммы вариантов использования применяется унифицированный язык визуального моделирования UML.

Диаграмма вариантов использования описывает функциональность разрабатываемой системы. Она отражает взаимодействие системы с актерами, такими как операторы БПЛА, службы реагирования и специалисты. Каждый прецедент на диаграмме описывает действия системы для актеров: загрузка изображения, загрузка данных для распознавания, само распознавание и вывод результирующего изображения. Диаграмма обеспечивает по-

нимание целей системы, выявляя пробелы и соответствие потребностям заинтересованных сторон. Прецедент служит для описания набора действий, которые система предоставляет пользователю.

Диаграмма представлена на рисунке 2.3



Рисунок 2.3 – Диаграмма вариантов использования

На основании анализа предметной области в программе должны быть реализованы следующие прецеденты:

1. Распознание объекта (возгорания) на изображении с помощью нейронной сети.
2. Сохранение результатов обучения для дальнейшего использования.
3. Загрузка результатов обучения для дальнейшего использования.
4. Обучение и переобучение нейронной сети.
5. Сохранение записей для обучения нейронной сети.

## **2.5 Требования к оформлению документации**

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

### **3 Технический проект**

#### **3.1 Общая характеристика организации решения задачи**

Целью этого проекта является спроектировать и разработать приложение, которое поможет специализированным службам по тушению всевозможных возгораний вести более эффективную и проработанную деятельность.

Это приложение представляет собой интеллектуальную систему, предназначенную для автоматического обнаружения и классификации очагов возгораний. Эта система способна, при помощи нейронной сети, распознавать объекты, такие как возгорания и пожары, и определять их класс и уровень опасности на основе цветовых характеристик и наличия задымленности на изображении, а также обучаться при помощи пользователя.

#### **3.2 Обоснование выбора технологии проектирования**

Для задачи классификации возгораний, полученных с БПЛА, используется две сверточные нейронные сети. Такие сети хорошо подходят для задачи обработки изображений и классификации объектов, что делает их эффективным выбором для анализа данных с камер БПЛА. Она способна извлекать характерные признаки из изображений, таких как формы, текстуры и цвета, что позволяет эффективно классифицировать возгорания.

##### **3.2.1 Описание используемых технологий и языков программирования**

В процессе разработки приложения используются программные средства и языки программирования. Каждое программное средство и каждый язык программирования применяется для круга задач, при решении которых они необходимы.

##### **3.2.2 Сверточные нейронные сети**

Convolutional neural network(CNN) или Сверточная нейронная сеть - это тип искусственной нейронной сети, который широко используется для задач

обработки изображений и компьютерного зрения. Ключевой особенностью CNN является использование сверточных слоев, которые извлекают характерные признаки из входных изображений. Эти признаки могут включать в себя формы, текстуры, края или цвета.

Основная идея CNN заключается в применении сверточных фильтров к входному изображению, что позволяет выявить повторяющиеся шаблоны или признаки. Эти фильтры скользят по изображению, извлекая информацию на разных уровнях абстракции. Затем эта информация проходит через дополнительные слои, такие как подвыборка или пулинг, которые уменьшают пространственные размеры данных, и полностью подключенные слои, которые выполняют классификацию или регрессию.

CNN показали впечатляющие результаты в задачах классификации изображений, обнаружения объектов, сегментации и даже в анализе медицинских изображений. Они эффективно обрабатывают большие объемы данных, обучаясь выявлять сложные зависимости и характерные признаки.

### **3.2.3 Машинное обучение**

Машинное обучение - это раздел искусственного интеллекта, который фокусируется на разработке алгоритмов и моделей, позволяющих компьютерам обучаться и улучшать свои задачи без явного программирования.

Применение алгоритмов машинного обучения лежит в основе нашей системы анализа и классификации данных. Мы обучаем нашу модель на обширной базе изображений, позволяя системе эффективно распознавать и классифицировать объекты в реальном времени. Этот процесс включает в себя использование сложных алгоритмов, которые могут извлекать и интерпретировать характерные особенности из данных изображений.

### **3.2.4 Язык программирования Python**

Python является одним из самых популярных и широко используемых языков программирования для разработки приложений искусственного ин-

теллекта и машинного обучения. Он имеет простой и понятный синтаксис, что ускоряет процесс разработки и делает код более читаемым.

### 3.2.5 Библиотеки Python

- NumPy - фундаментальная библиотека для научных расчетов в Python. Она обеспечивает эффективную работу с многомерными массивами и матричными вычислениями, что критически важно для обработки и манипуляции данными.
- Matplotlib - библиотека для визуализации данных. Она позволяет создавать настраиваемые и интуитивно понятные графики, диаграммы и изображения, что облегчает визуальный анализ данных и представление результатов.
- OpenCV (Open Source Computer Vision Library) - библиотека компьютерного зрения, которая предлагает широкий спектр алгоритмов для обработки изображений и видео. Она идеально подходит для задач обработки изображений, обнаружения объектов и анализа видео, полученных с камер БПЛА.
- Pandas - библиотека для анализа и манипуляции данными. Она предоставляет удобные структуры данных, такие как DataFrame, и богатый набор инструментов для обработки, фильтрации и агрегации данных, упрощая подготовку и анализ больших наборов данных.
- TensorFlow - это открытая платформа машинного обучения с масштабируемыми инструментами для обучения и развертывания моделей. Она обеспечивает гибкую и эффективную инфраструктуру для создания сложных нейронных сетей. Keras - это высокоуровневый API, построенный на TensorFlow, который упрощает процесс создания и обучения нейронных сетей. Он предлагает простой и интуитивно понятный интерфейс, позволяя быстро разрабатывать и экспериментировать с различными архитектурами моделей.

### 3.2.6 Архитектура сверточной нейронной сети

Архитектура сверточной нейронной сети включает в себя 14 слоев с различными функциями.

Схема архитектуры нейронной сети представлена на рисунке 3.1.

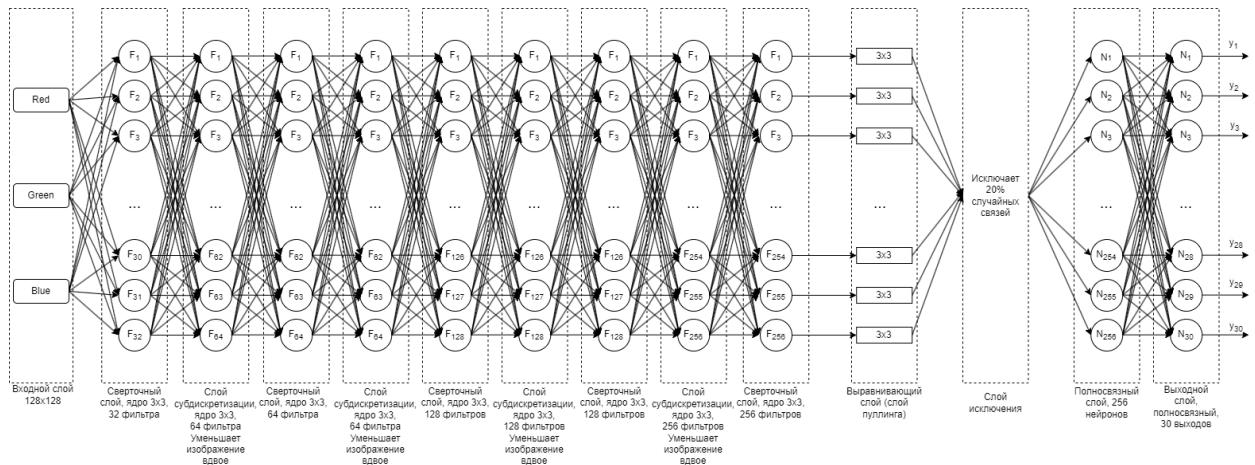


Рисунок 3.1 – Архитектура нейронной сети

Слои нейронной сети классификации описаны далее:

#### 1. Input Layer (Входной слой)

Функция Input определяет входной слой с размером изображения  $128 \times 128$  пикселей и 3 цветовыми каналами (RGB).

#### 2. Convolutional Layers (Сверточные слои, 9)

Первый свёрточный слой Conv2D имеет 32 фильтра размером  $3 \times 3$ , функцию активации ReLU и параметр padding='same', который сохраняет размерность входного изображения.

Второй свёрточный слой Conv2D увеличивает количество фильтров до 64 и применяет шаг (strides) равный 2, что уменьшает размерность изображения вдвое.

Следующие два слоя Conv2D также имеют 64 фильтра и один из них снова применяет шаг 2 для уменьшения размерности.

Пятый и шестой свёрточные слои Conv2D содержат 128 фильтров каждый, с шагом 2 на шестом слое.

Седьмой свёрточный слой Conv2D сохраняет 128 фильтров и размерность.

Восьмой и девятый свёрточные слои Conv2D увеличивают количество фильтров до 256, с шагом 2 на восьмом слое.

### 3. Flatten Layer (Выравнивающий слой)

Слой Flatten преобразует многомерный тензор свёрточных слоёв в одномерный, чтобы его можно было подать на полно связные слои.

### 4. Dropout Layer (Слой исключения)

Слой Dropout с параметром 0.2 предотвращает переобучение, случайным образом "выключая" 20

### 5. Dense Layers (Полно связные слои, 2)

Первый полно связный слой Dense имеет 256 нейронов и функцию активации ReLU.

Второй полно связный слой Dense формирует выходной слой с 30 нейронами, количество которых соответствует количеству выходных значений, которые должна предсказывать модель.

#### 3.2.7 Функция активации ReLU

ReLU, или Rectified Linear Unit, — это функция активации, которая используется в нейронных сетях для увеличения нелинейности. Формула ReLU представлена формулой 1.

$$f(x) = \max(0, x) \quad (1)$$

Это означает, что если вход x положительный, то функция возвращает x, а если x отрицательный, то функция возвращает 0. ReLU популярна потому, что она ускоряет обучение нейронных сетей без значительной потери точности. Она также помогает решить проблему исчезающего градиента, так как производная для положительных значений всегда равна 1, что обеспечивает более быстрое и эффективное обучение.

График функции ReLu предоставлен на изображении 3.2.

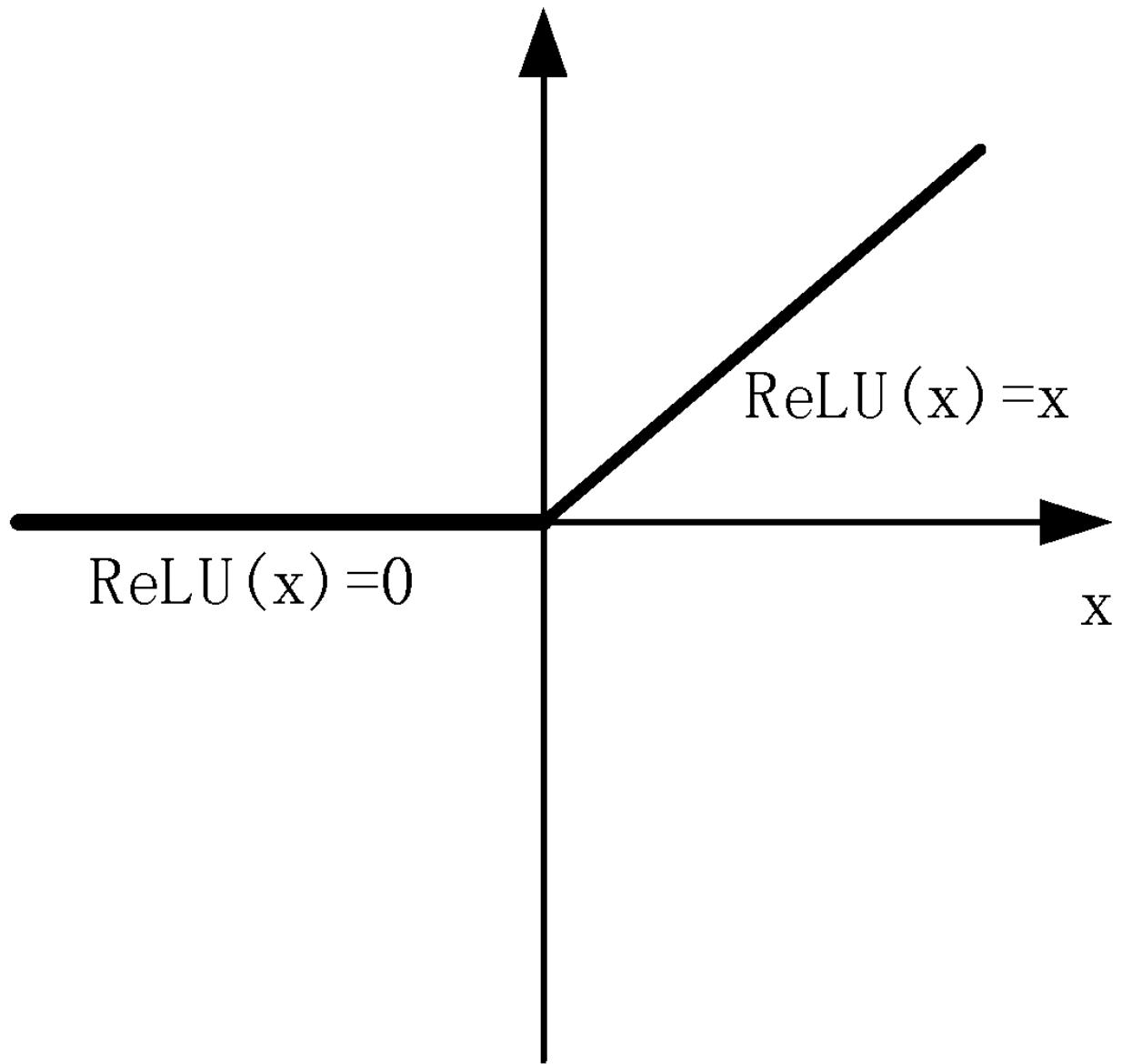


Рисунок 3.2 – График функции ReLU

### 3.2.8 Функция IoU Loss

Intersection over Union (IoU) является метрикой, используемой для измерения точности объектного детектора на определенном наборе данных. Если мы работаем с задачами компьютерного зрения, такими как сегментация изображений или обнаружение объектов, IoU может помочь оценить, насколько предсказанные границы объекта соответствуют истинным границам.

Наглядным образом функцию можно увидеть на изображении 3.3.

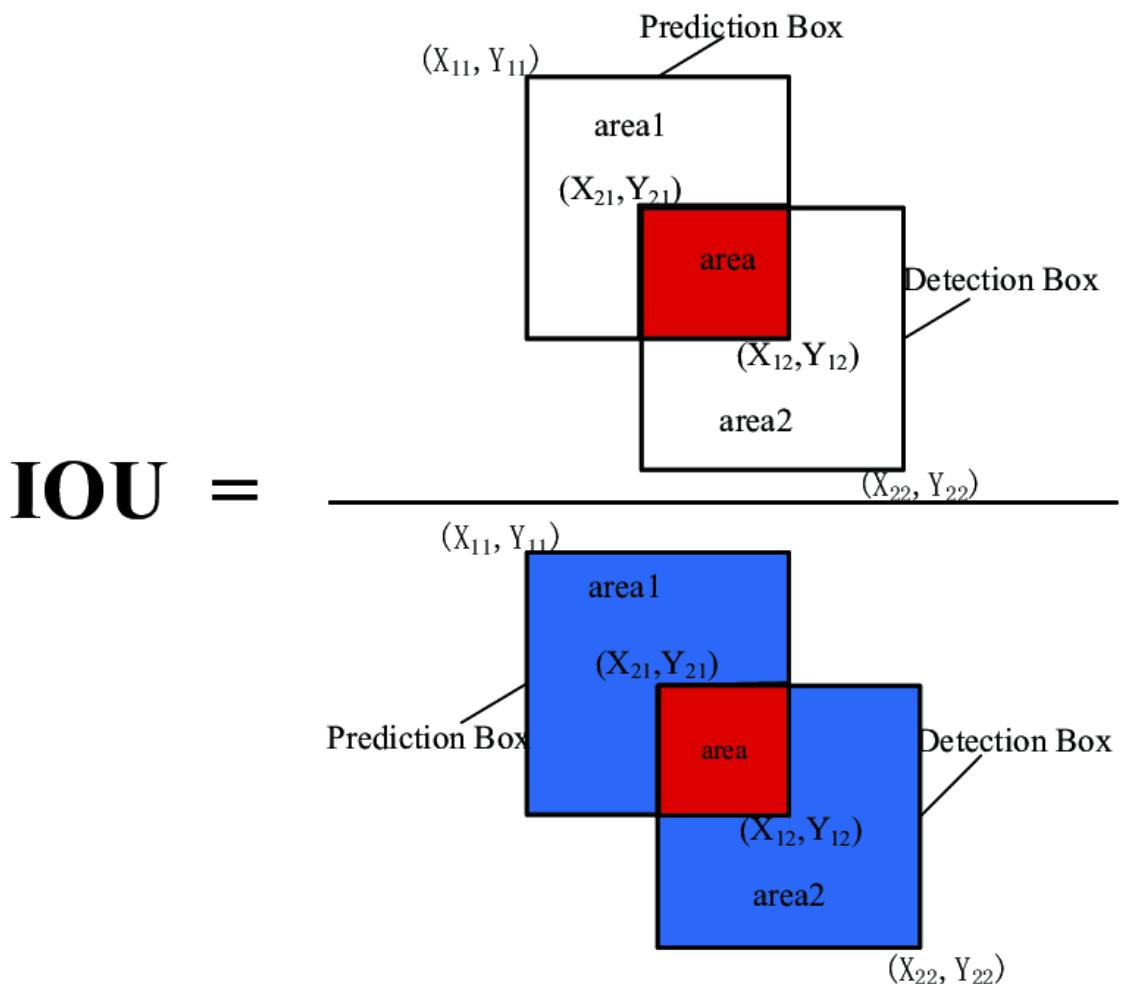


Рисунок 3.3 – График функции IoU

IoU рассчитывается по формуле 2.

$$IoU = \frac{\text{площадь объединения}}{\text{площадь пересечения}} \quad (2)$$

Где:

- Площадь пересечения - это область, где предсказанная граница и истинная граница объекта перекрываются.
- Площадь объединения - это область, покрытая как предсказанный границей, так и истинной границей, вместе взятых.

IoU loss - это функция потерь, которая используется для обучения моделей, выполняющих задачи, связанные с локализацией объектов. Вместо того чтобы использовать стандартные функции потерь, такие как кросс-энтропия, которые могут не полностью отражать точность локализации, IoU loss направ-

мую оптимизирует метрику IoU, стремясь увеличить площадь пересечения и уменьшить площадь объединения.

Функция потерь IoU показана в формуле 3.

$$IoULoss = 1 - IoU \quad (3)$$

Таким образом, минимизация IoU loss в процессе обучения приводит к увеличению IoU между предсказанными и истинными границами, что помогает в задачах локализации объектов нашего проекта.

### 3.3 Диаграмма компонентов

Диаграмма компонентов представляет структуру системы в виде набора компонентов и их взаимосвязей. Каждый компонент отвечает за определенную функцию в рамках системы и может включать в себя подсистемы или модули.

- Графический интерфейс: отвечает за создание интерфейса, в котором пользователь наглядно видит результат распознавания в сравнении с оригиналом;
- Обработка изображения: включает в себя методы улучшения качества изображений, такие как коррекция освещения, удаление шумов и извлечение важных признаков;
- Сверточная нейронная сеть (CNN): Ядро этой системы, реализующее алгоритмы обучения и распознавания объектов;
- Данные параметров нейронной сети: представляют собой обученную модель, которая хранит в себе веса и параметры, извлеченные из данных во время процесса обучения;
- Классификация объекта: этот модуль используется для классификации возгорания;
- Генерация отчета: отвечает за создание отчета для выводения класса возгорания и оценки его опасности.

### 3.3.1 Взаимодействие компонентов

1. Пользователь загружает изображение через графический интерфейс пользователя.
2. Интерфейс передает изображение в модуль обработки изображения.
3. После обработки данные передаются в модуль сверточной нейронной сети для распознавания объекта.
4. Модуль данных параметров передает веса и параметры и корректирует работу нейронной сети.
5. Результаты распознавания классифицируются в модуле классификации объекта.
6. Модуль генерации отчета выводит данные о возгорании в графическом интерфейсе.

Диаграмма компонентов представлена на рисунке 3.4.

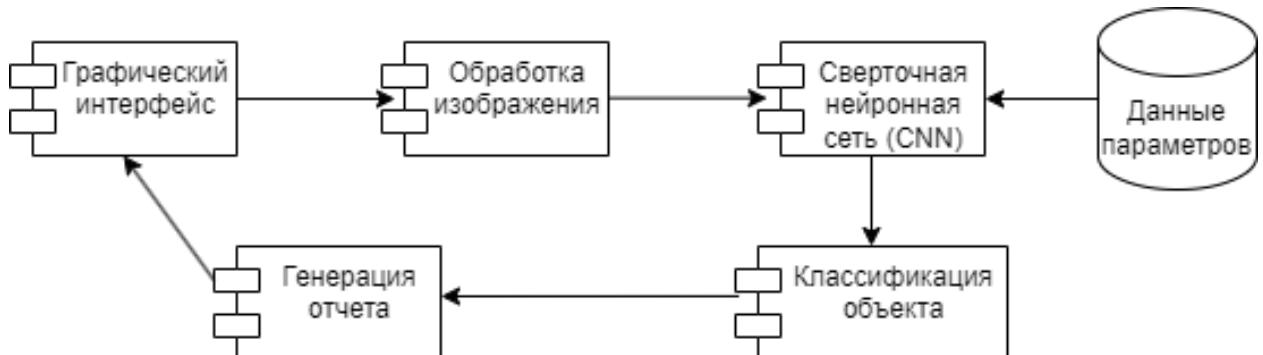


Рисунок 3.4 – Диаграмма компонентов

### 3.3.2 Диаграмма программных классов

Точка входа в программу является класс main. В этом классе осуществляется запуск и инициализация основных компонентов программы:

- dataprocessing - компонент, отвечающий за обработку данных и изображений.
- creatingtfrecordclassifier - компонент, отвечающий за создание и загрузку изображений и создания их записи в формате .tfrecord

- `creatingtfrecordlocalizer` - компонент, отвечающий за создание и загрузку изображений и создания их записи в формате `.tfrecord`
- `classifier` - компонент, отвечающий за работу с нейронной сетью по классификации данных, её обучение и тестирование.
- `training` - компонент, отвечающий за работу с нейронной сетью по распознанию данных, её обучение и тестирование.

Диаграмма классов предоставлена на рисунке ??.

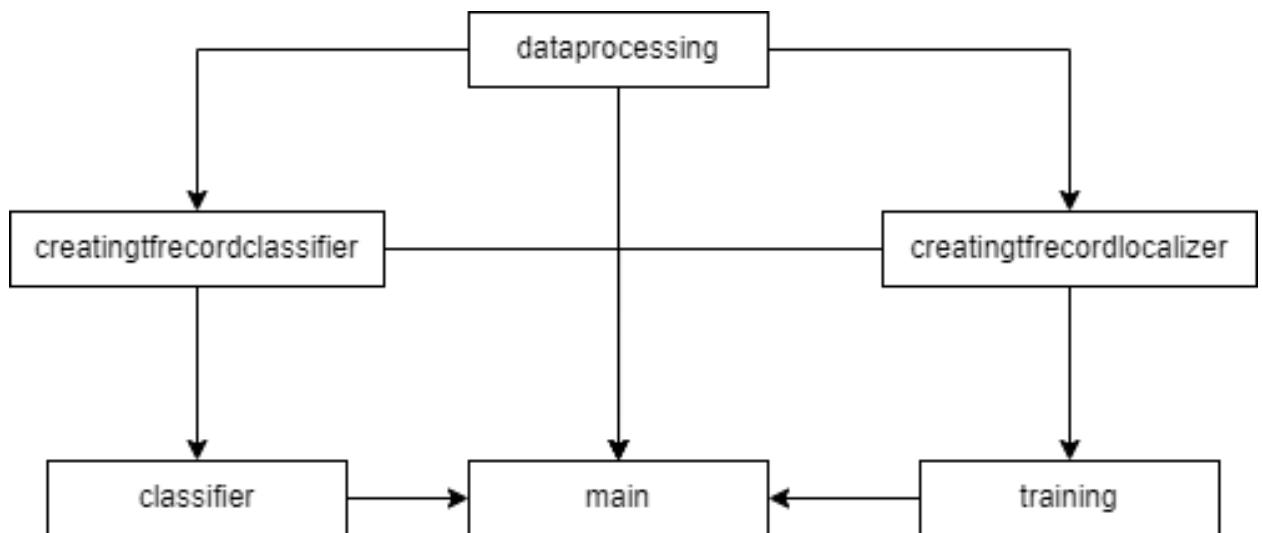


Рисунок 3.5 – Диаграмма классов и их связей

На диаграмме показаны данные связи:

- Связь `dataprocessing` - `main`. Класс `main` использует основные функции класса `dataprocessing`, такие как отображение информации о изображении, загрузка изображения и нормализация координат изображения.
- Связь `dataprocessing` - `creatingtfrecordclassifier`. Класс `creatingtfrecordclassifier` использует основные функции класса `dataprocessing`, такие как загрузка изображения и нормализация координат изображения. Для создания записи `tfrecord` необходимо создать запись со всеми изображениями и файлами формата `.xml`.
- Связь `dataprocessing` - `creatingtfrecordlocalizer`. Класс `creatingtfrecordlocalizer` использует основные функции класса `dataprocessing`, такие как загрузка изображения и нормализация координат изображения.

ния. Для создания записи tfrecord необходимо создать запись со всеми изображениями и файлами формата .xml.

- Связь creatingtfrecordclassifier - classifier. Для работы классу classifier нужен созданный в классе creatingtfrecordclassifier файл tfrecord.
- Связь creatingtfrecordclassifier - main. Класс main использует основные функции класса creatingtfrecordclassifier, такие как создание файла tfrecord для классификатора.
- Связь creatingtfrecordlocalizer - training. Для работы классу training нужен созданный в классе creatingtfrecordlocalizer файл tfrecord.
- Связь creatingtfrecordlocalizer - main. Класс main использует основные функции класса creatingtfrecordlocalizer, такие как создание файла tfrecord для локализатора.
- Связь classifier - main. Класс main использует основные функции класса classifier, такие как тестирование, обучение, загрузка и сохранение сети.
- Связь localizer - main. Класс main использует основные функции класса localizer, такие как тестирование, обучение, загрузка и сохранение сети.

### **3.4 Проектирование пользовательского интерфейса**

На основании требований к пользовательскому интерфейсу, представленных в пункте 2.3 технического задания, был разработан графический интерфейс приложения. Для создания пользовательского интерфейса используется библиотека tkinter и matplotlib.

На рисунке 3.6 представлен макет интерфейса окон для распознания и обучения нейронной сети. Данный макет содержит следующие элементы:

1. Загрузка изображения из системы.
2. Загрузка своей модели нейронной сети.
3. Запустить распознание изображения
4. Открыть окно тренировки модели нейронной сети.
5. Поле загруженного изображения.
6. Поле изображения с распознанным возгоранием.

7. Закрывает окно тренировки модели нейронной сети.
8. Показывает информацию о первой картинке в папке с изображениями для обучения.
9. Запускает стороннюю программу labelimg.
10. Показывает в поле для вывода информацию о наличии файлов.
11. Создает новую запись tfrecord для локализатора.
12. Создает новую запись tfrecord для классификатора.
13. Запускает функцию тестирования классификатора.
14. Запускает другую функцию тестирования классификатора с точными значениями.
15. Запускает функцию обучения классификатора.
16. Сохраняет модель нейронной сети классификатора.
17. Сохраняет модель нейронной сети локализатора.
18. Загружает модель нейронной сети локализатора.
19. Запускает функцию тестирования нейронной сети локализатора.
20. Запускает функции для обучения нейронной сети локализатора.
21. Поле вывода информации из терминала после выполнения любых операций.

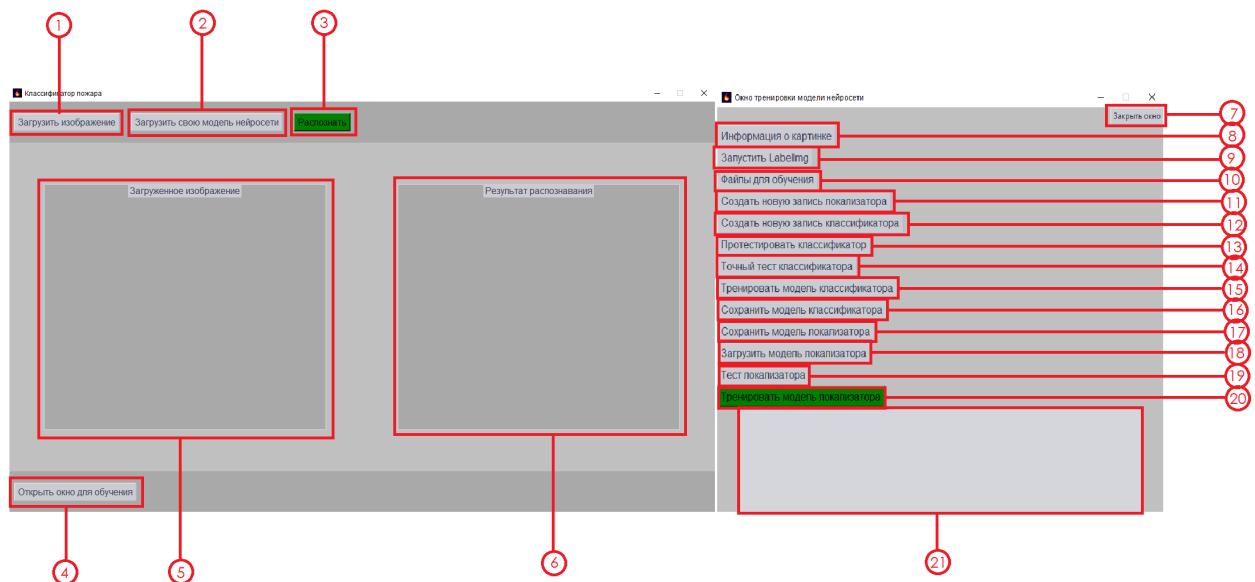


Рисунок 3.6 – Макет интерфейса окна распознавания и окна обучения нейронной сети

## 4 Рабочий проект

### 4.1 Классы, используемые при разработке приложения

Список классов и методов, которые были использованы при создании приложения представлены далее.

#### 4.1.1 Класс main

Класс `main` является точкой входа в приложение и используется для реализации идентификации объектов с помощью нейронной сети. Здесь происходит инициализация основных компонентов программы в качестве кнопок на интерфейсе приложения. Описание полей и методов данного класса представлено в таблице 4.1.

Таблица 4.1 – Спецификация полей класса «`main`»

Наименование	Метод доступа	Тип данных	Описание
1	2	3	4
path_for_one	public	String	Хранит путь к выбранному изображению
localizer	public	tf.keras.Model	Модель локализатора, загруженная из файла. Файл хранится в корневой папке проекта и создается в классе <code>training</code> .
classifier	public	tf.keras.Model	Модель классификатора, загруженная из файла. Файл хранится в корневой папке проекта и создается в классе <code>classifier</code> .

Продолжение таблицы 4.1

1	2	3	4
window	private	Tk	Главное окно приложения. Окно настроено и имеет несколько вложений соответствующие визуальной составляющей. Содержит в себе кнопки и поля Canvas.
image_field_raw	private	Canvas	Холст для отображения исходного изображения. Содержится в окне window.
image_field_ready	private	Canvas	Холст для отображения результата распознавания. Содержится в окне window.
ConsoleRedirector	public	Class	Класс для перенаправления вывода консоли в текстовое поле text с помощью функции sys.stdout = ConsoleRedirector(text).
image_field_ready	private	Canvas	Холст для отображения результата распознавания. Содержится в окне window.

## Продолжение таблицы 4.1

1	2	3	4
detect_objects	private	Function	Функция для обнаружения объектов на изображении с использованием модели локализатора. Здесь происходит подготовка изображения, локализация, нормализация, разделение на элементы, нарезание изображений и сбор в массив (10, 32, 32, 3). Далее происходит классификация, счет метки класса и сбор координат в нормальный вид.
namespace	public	Dictionary	Словарь для сопоставления индексов классов с их названиями. В нашем варианте 0 - Ничего (nothing), 1 - Возгорание (fire).
visualize	private	Function	Функция для визуализации результатов детекции с помощью OpenCV. Рисует текст и квадраты на изображении в соответствии с распознанными объектами (возгораниями)

## Продолжение таблицы 4.1

1	2	3	4
prettify	private	Function	Функция для улучшения результатов детекции путём объединения перекрывающихся рамок. Здесь вычисляем IoU, самый надежный способ определить совпадение. И если ни с чем не объединили, так и оставляем результат.
detect_fire_in_image	private	Function	Функция для обнаружения огня на изображении и его визуализации. Считывает изображение, переводит его в matplotlib и выводит на Canvas поле результата.
loadimage	public	Function	Функция для загрузки изображения через диалоговое окно. Также помещает его в Canvas поле загруженного изображения.
detect	private	Function	Функция для запуска процесса обнаружения огня на загруженном изображении. Вызывает detect_fire_in_image().

Продолжение таблицы 4.1

1	2	3	4
check_and_install_packages	private	Function	Функция для проверки и установки необходимых пакетов PyQt5 и sip. Вызывается для предварительного устранения ошибок с пакетами при запуске labelimg.
run_labelimg	private	Function	Функция для запуска приложения LabelImg для аннотации изображений. Labelimg установлен заранее, но требует для запуска дополнительные пакеты в системе.
create_tfrec_classifier	private	Function	Функция для создания TFRecord для классификатора. Вызывает функцию из другого класса.
start_train_localizer	private	Function	Функция для начала обучения модели локализатора. Вызывает функцию train() и loadmodel() из другого класса.
train_window_open	private	Function	Открывает новое окно для обучения нейронной сети. Здесь реализованы функции для удобного обучения, тестирования и работы с файлами нейронной сети.
load_model_data	private	Function	Функция для загрузки пользовательской модели нейросети.

#### 4.1.2 Класс dataprocessing

Класс dataprocessing отвечает за обработку данных и изображений. Класс предназначен для обработки данных из XML файла, который содержит аннотации для изображений, используемых в наших задачах. В классе парсится XML файл, извлекается информацию о размеченных объектах и их координатах, загружается соответствующее изображение и визуализирует его. Описание полей и методов данного класса представлено в таблице 4.3.

Таблица 4.3 – Спецификация полей класса «dataprocessing»

Наименование	Метод доступа	Тип данных	Описание
1	2	3	4
tree	public	xml.etree.ElementTree	Объект дерева XML, содержащий структурированные данные из файла XML. Значением является адрес файла для информирования.
root	public	xml.etree.ElementTree	Корневой элемент XML файла, представляющий доступ к дочерним элементам. Парсинг.
num_objects	public	int	Количество объектов, размещенных в XML файле. Количество размещенных объектов (6 - кол-во служебных элементов, таких как размер, название и т.д)
load_img	public	Function	Функция для загрузки и предобработки изображения с использованием TensorFlow.

Продолжение таблицы 4.3

1	2	3	4
cords	public	list	Список для хранения нормализованных координат объектов.
w	public	int	Ширина изображения, извлеченная из XML файла.
h	public	int	Высота изображения, извлеченная из XML файла.
object_cords	public	list	Список для хранения нормализованных координат одного объекта. Тут мы также нормализуем координаты от -1 до 1, опираясь на исходные координаты.
img	public	Tensor	Тензор изображения после загрузки и предобработки.
plt.figure	public	Function	Функция для создания новой фигуры в Matplotlib.
plt.subplot	public	Function	Функция для добавления подграфика в текущую фигуру.
plt.imshow	public	Function	Функция для отображения изображения в подграфике.
plt.axis	public	Function	Функция для управления отображением осей графика.
plt.show	public	Function	Функция для отображения всей фигуры с подграфиками. Открывает окно с исходным изображением.

#### 4.1.3 Класс creatingtfrecordclassifier

Класс creatingtfrecordclassifier отвечает за создание и загрузку изображений и создания их записи в формате .tfrecord. В этом классе подготавливаются данные для дальнейшего использования в других классах и главное - создание файла classifier\_dataset.tfrecord . Описание полей и методов данного класса представлено в таблице 4.5.

Таблица 4.5 – Спецификация полей класса «creatingtfrecordclassifie»

Наименование	Метод доступа	Тип данных	Описание
1	2	3	4
fn	public	str	Путь к папке с изображениями. Значением является адрес папки с нашими изображениями и xml файлами.
check_xml_list	public	function	Функция для создания списка XML файлов в указанной папке. Формируем список всех xml файлов в папке.
load_img	public	function	Функция для загрузки и предобработки изображения.
create_load_tfrec_for_classifier	public	function	Функция для создания TFRecord для классификатора. Здесь мы преобразуем папку в tfrecord для классификатора

## Продолжение таблицы 4.5

1	2	3	4
namespace	public	dictionary	Словарь для сопоставления названий с метками классов. В нашем случае 0 - Ничего (nothing), 1- Возгорание (fire).
saveinrecord	public	function	Внутренняя функция для сохранения обработанного изображения и метки в TFRecord. Создается запись writer, данные предоставляются в байтовом виде и собирается экземпляр, который потом записывается в запись writer.
parse_record	public	function	Функция для разбора записей TFRecord. Имена элементов остаются как при записи

### 4.1.4 Класс creatingtfrecordlocalizer

Класс creatingtfrecordlocalizer отвечает за создание и загрузку изображений и создания их записи в формате .tfrecord. В этом классе подготавливаются данные для дальнейшего использования в других классах и главное - создание файла localizer\_dataset.tfrecord . Описание полей и методов данного класса представлено в таблице 4.7.

Таблица 4.7 – Спецификация полей класса «creatingtfrecordlocalizer»

Наименование	Метод доступа	Тип данных	Описание
1	2	3	4
fn	public	string	Путь к папке с изображениями. Значением является адрес папки с нашими изображениями и xml файлами.
load_img	private	function	Загружает изображение, декодирует, нормализует и изменяет размер.
create_load_tfrec_for_localizer	private	function	Создает TFRecord для локализатора из XML файлов.
p	private	list	Формируем список всех xml файлов в папке.
writer	private	TFRecordWriter	Записывает данные в TFRecord. Создание самой записи
tree	private	ElementTree	Адрес файла
root	private	Element	Парсит XML файлы.
num_objects	private	int	Количество объектов в XML файле.
cords	private	list	Список координат объектов.
w	private	int	Ширина изображения.
h	private	int	Высота изображения.

## Продолжение таблицы 4.7

1	2	3	4
object_cords	private	list	Координаты одного объекта. Тут мы также нормализуем координаты от -1 до 1, опираясь на исходные координаты.
img	private	Tensor	Тензор изображения.
serialized_img	private	bytes	Сериализованное изображение. Готовим данные, представляем в байтовом виде.
serialized_cords	private	bytes	Сериализованные координаты. Готовим данные, представляем в байтовом виде.
example	private	Example	Пример данных для TFRecord. Собираем экземпляр
dataset	private	TFRecordDataset	Набор данных из TFRecord. Сразу после создания проверка чтения записи.
parse_record	private	function	Разбирает запись из TFRecord. Имена элементов как при записи
feature_description	private	dict	Описание приходящего экземпляра.
parsed_record	private	dict	Разобранный экземпляр.

#### 4.1.5 Класс classifier

Класс classifier отвечает за работу с нейронной сетью по классификации данных, её обучение и тестирование. В этом классе реализован парсинг элементов из tfrecord, загрузка и создание модели нейросети, а также построение её архитектуры. Описание полей и методов данного класса представлено в таблице 4.9.

Таблица 4.9 – Спецификация полей класса «classifier»

Наименование	Метод доступа	Тип данных	Описание
1	2	3	4
parse_record	public	function	Разбирает запись TFRecord, возвращая изображение и имя.
shuffle, cache, prefetch, batch	public	method	Подготавливает датасет к обучению: перемешивание, кэширование, предварительная загрузка, батчинг.
test_classifier	public	function	Визуализирует примеры из датасета и их метки.
Model	public	class	Определяет модель нейросети с методами для обучения.
training_step	public	method	Выполняет шаг обучения, возвращая среднее значение потерь.
trainclass	public	function	Обучает классификатор и визуализирует диаграмму потерь.
imshow_and_pred	public	function	Визуализирует изображения и предсказания модели с помощью matplotlib.

## Продолжение таблицы 4.9

1	2	3	4
saveclassifier	public	function	Сохраняет обученную модель классификатора.

### 4.1.6 Класс training

Класс `training` отвечает за работу с нейронной сетью по локализации данных, её обучение и тестирование. В этом классе реализован парсинг элементов из `tfrecord`, загрузка и создание модели нейросети, а также построение её архитектуры и явное обучение. Описание полей и методов данного класса представлено в таблице 4.11.

Таблица 4.11 – Спецификация полей класса «`training`»

Наименование	Метод доступа	Тип данных	Описание
1	2	3	4
parse_record	public	function	Разбирает запись <code>TFRecord</code> , возвращая изображение и координаты.
IoU_Loss	public	function	Вычисляет потери IoU между истинными и предсказанными рамками.
Model	public	class	Определяет модель нейросети с методами для обучения.
training_step	public	method	Выполняет шаг обучения, возвращая значение потерь.
savemodel	public	function	Сохраняет обученную модель.
loadmodel	public	function	Загружает веса модели.

## Продолжение таблицы 4.11

1	2	3	4
testing	public	function	Тестирует модель, визуализируя результаты.
train	public	function	Обучает модель и визуализирует историю потерь.

## 4.2 Модульное тестирование разработанного web-сайта

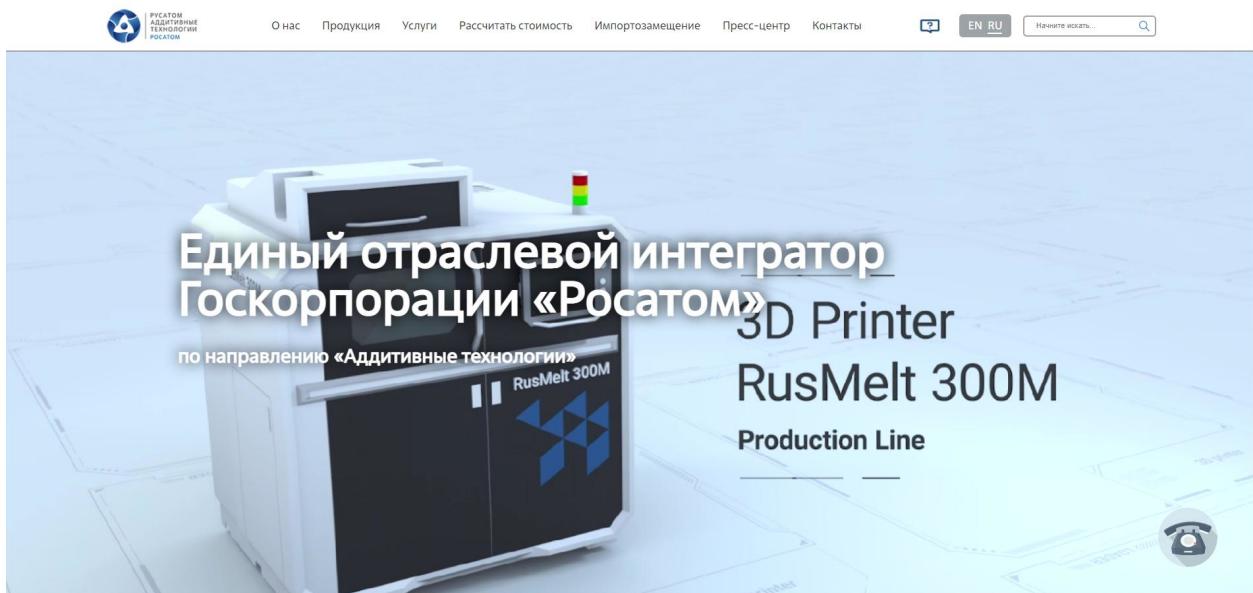
Модульный тест для класса User из модели данных представлен на рисунке 4.1.

```
1 from django.test import TestCase
2 from .models import *
3 User = get_user_model()
4
5
6 class ShpoTestCases(TestCase):
7
8     def setUp(self) -> None:
9         self.user = User.objects.create(username='testtestovich', password='
10             testtestovich', first_name='Sad', last_name='')
11
12     def test_2(self):
13
14         self.assertEqual(self.user.first_name, 'Sad')
15         self.assertEqual(self.user.last_name, 'Cat')
16         print((self.user))
17         print((self.user.first_name))
18         print((self.user.last_name))
```

Рисунок 4.1 – Модульный тест класса User

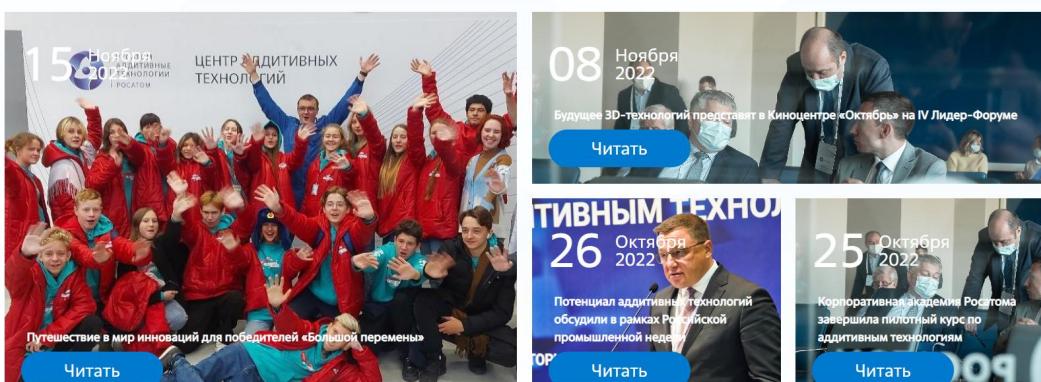
## 4.3 Системное тестирование разработанного web-сайта

На рисунке 4.2 представлена главная страница сайта «Русатом – Аддитивные технологии».



## Новости

Новости



## Области применения

### Гражданская авиация

3D-печать топливных форсунок, турбинных колес, изделий с применением бионического дизайна.

#### Преимущества:

- Снижение веса деталей. Скорость. Быстрая скорость создания деталей.
- Экологичность. Во время работы на 3D-принтере отходов практически нет.
- Конфигурация изделия. Создание форм, которые невозможно воспроизвести другими способами.

Рисунок 4.2 – Главная страница сайта «Русатом – Аддитивные технологии»

На рисунке 4.3 представлен динамический вывод заголовков, включающий в себя искомые фразы при поиске фраз.

На рисунке 4.4 представлен ввод данных для публикации новести.

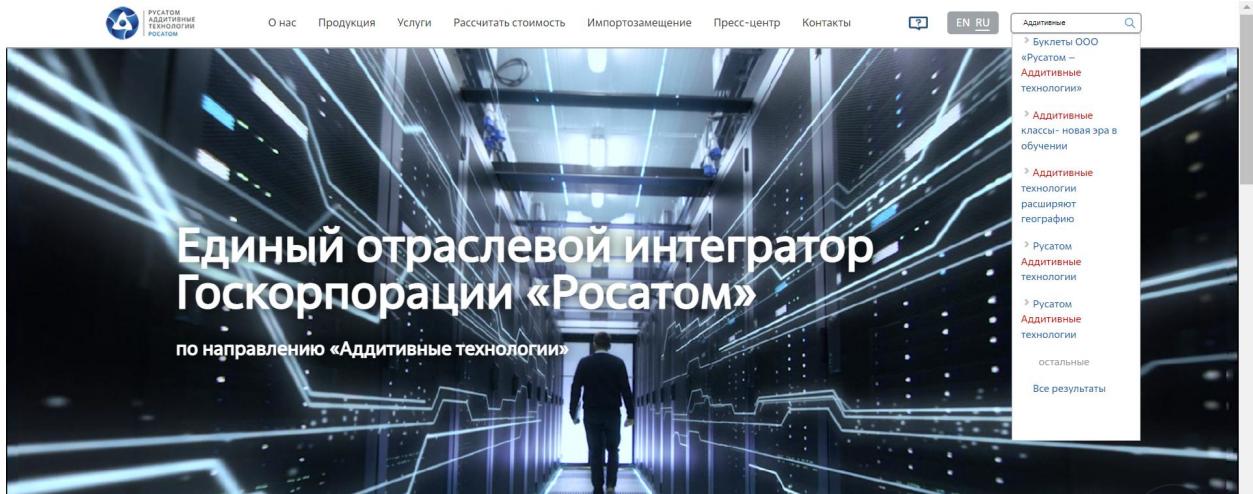


Рисунок 4.3 – Динамический вывод заголовков

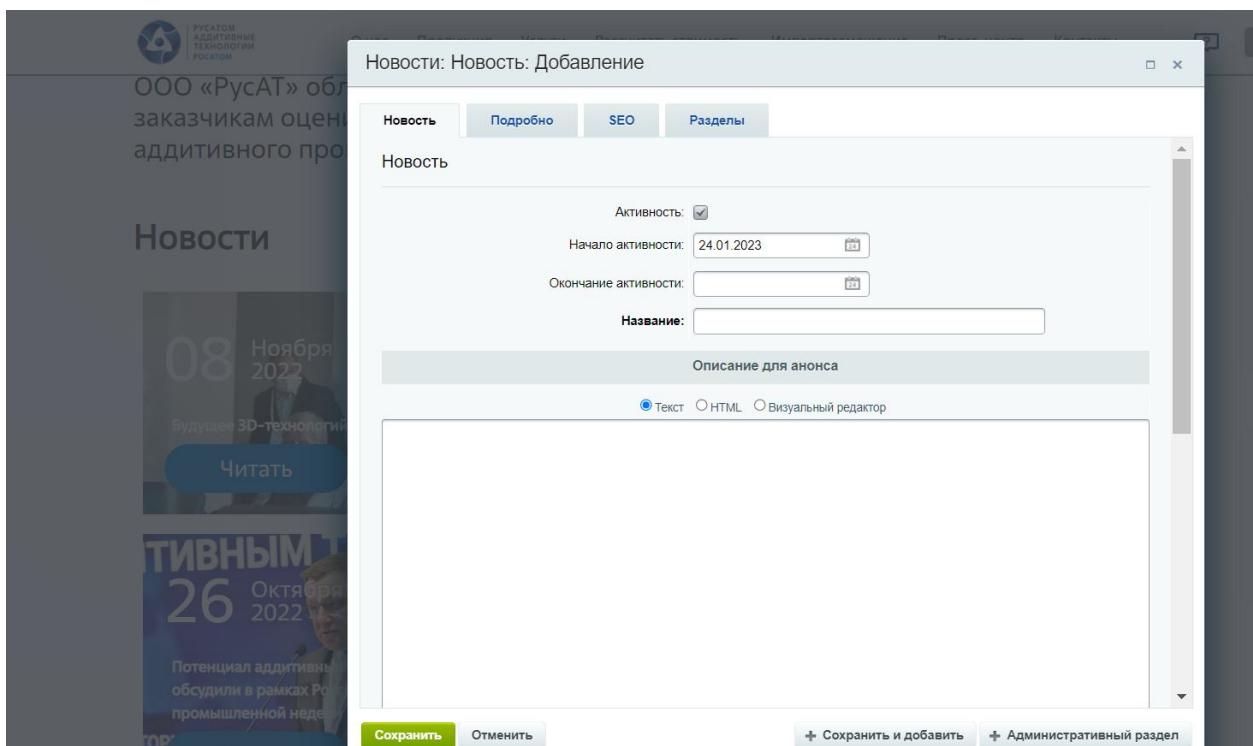


Рисунок 4.4 – Ввод данных для публикации очень-очень длинной, интересной и полезной новости

## ЗАКЛЮЧЕНИЕ

Преимущества аддитивных технологий заключается в разнообразии процессов, позволяющих применять их в различных областях производства. Существенным ограничением же является и экономическая составляющая, которая не позволит внедрить аддитивное производство повсеместно.

Компании, видя, как развиваются информационные технологии, пытаются использовать их выгодно для своего бизнеса, запуская свой сайт для того, чтобы заявить о своем существовании, проинформировать потенциального клиента об услугах или продуктах, которые предоставляет. Для продвижения компании «Русатом – Аддитивные технологии» был разработан веб-сайт на основе системы «1С-Битрикс: Управление сайтом».

Основные результаты работы:

1. Проведен анализ предметной области. Выявлена необходимость использовать 1С-Битрикс.
2. Разработана концептуальная модель web-сайта. Разработана модель данных системы. Определены требования к системе.
3. Осуществлено проектирование web-сайта. Разработана архитектура серверной части. Разработан пользовательский интерфейс web-сайта.
4. Реализован и протестирован web-сайт. Проведено модульное и системное тестирование.

Все требования, объявленные в техническом задании, были полностью реализованы, все задачи, поставленные в начале разработки проекта, были также решены.

Готовый рабочий проект представлен адаптивной версткой сайта. Сайт находится в публичном доступе, поскольку опубликован в сети Интернет.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Хайкин, С. Нейронные сети: полный курс, 2-е издание / С. Хайкин. – Москва : Вильямс, 2018. – 1104 с. – ISBN 978-5-8459-2101-0. – Текст : непосредственный.
2. Лутц, М. Изучаем Python, 5-е издание / М. Лутц. – Санкт-Петербург : Питер, 2019. – 1584 с. – ISBN 978-5-4461-0705-9. – Текст : непосредственный.
3. Гудфеллоу, И., Бенджио, Ю., Курвилль, А. Глубокое обучение / И. Гудфеллоу, Ю. Бенджио, А. Курвилль. – Москва : ДМК Пресс, 2017. – 652 с. – ISBN 978-5-97060-487-9. – Текст : непосредственный.
4. Мерфи, К. Машинное обучение: вероятностный подход / К. Мерфи. – Москва : ДМК Пресс, 2018. – 704 с. – ISBN 978-5-97060-212-7. – Текст : непосредственный.
5. Рашка, С., Мирджалили, В. Python и машинное обучение / С. Рашка, В. Мирджалили. – Москва : ДМК Пресс, 2018. – 418 с. – ISBN 978-5-97060-310-0. – Текст : непосредственный.
6. Жолковский, Е. К. TensorFlow для профессионалов / Е. К. Жолковский. – Москва : ДМК Пресс, 2019. – 480 с. – ISBN 978-5-97060-746-7. – Текст : непосредственный.
7. Чоллет, Ф. Глубокое обучение на Python / Ф. Чоллет. – Москва : ДМК Пресс, 2018. – 304 с. – ISBN 978-5-97060-409-1. – Текст : непосредственный.
8. Клейн, Р. Нечеткие системы в Python / Р. Клейн. – Москва : ДМК Пресс, 2020. – 320 с. – ISBN 978-5-97060-758-0. – Текст : непосредственный.
9. Байдер, Д. Python Tricks: A Buffet of Awesome Python Features / Д. Байдер. – Москва : ДМК Пресс, 2021. – 300 с. – ISBN 978-5-97060-999-7. – Текст : непосредственный.

10. Герон, О. Практическое машинное обучение с Scikit-Learn и TensorFlow / О. Герон. – Москва : ДМК Пресс, 2019. – 572 с. – ISBN 978-5-97060-524-1. – Текст : непосредственный.

11. Нильсен, М. Нейронные сети и глубокое обучение / М. Нильсен. – Москва : ДМК Пресс, 2021. – 250 с. – ISBN 978-5-97060-777-1. – Текст : непосредственный.

## **ПРИЛОЖЕНИЕ А**

### **Представление графического материала**

Графический материал, выполненный на отдельных листах, изображен на рисунках А.1–А.4.

БОРГ 2068443.09.03.04.23008	1																																																					
<h2>Сведения о ВКРБ</h2> <p><b>Минобрнауки России Юго-Западный государственный университет</b></p> <p>Кафедра программной инженерии</p> <p><b>ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА ПО ПРОГРАММЕ БАКАЛАВРИАТА</b></p> <p>«Разработка web-сайта «Русатом – Аддитивные технологии» на платформе 1С – Битрикс»</p>																																																						
<p>Руководитель ВКР к.т.н, доцент Малышев Александр Васильевич</p>	<p>Автор ВКР студентка группы ПО-81з Мягкая Ирина Витальевна</p>																																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">Логотип работы</td> <td style="width: 10%;">Мягкая И.В.</td> <td style="width: 10%;">Подпись/Дата</td> </tr> <tr> <td>Руководитель</td> <td>Малышев А.В.</td> <td></td> </tr> <tr> <td>Вице-руководитель</td> <td>Нагибина А.А.</td> <td></td> </tr> <tr> <td>Научный консультант</td> <td></td> <td></td> </tr> <tr> <td>Научный руководитель</td> <td></td> <td></td> </tr> <tr> <td>Координатор</td> <td></td> <td></td> </tr> <tr> <td>Компьютерный специалист</td> <td></td> <td></td> </tr> <tr> <td>Контактный телефон</td> <td></td> <td></td> </tr> <tr> <td>Адрес электронной почты</td> <td></td> <td></td> </tr> <tr> <td>Факс</td> <td></td> <td></td> </tr> <tr> <td>Сайт</td> <td></td> <td></td> </tr> </table> <p style="margin-top: 10px;">Сведения о ВКРБ</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">Логотип</td> <td style="width: 10%;">Имя</td> <td style="width: 10%;">Номер</td> <td style="width: 10%;">Московский</td> </tr> <tr> <td>Борисов</td> <td>Лисова</td> <td>1</td> <td>Гусев</td> </tr> <tr> <td>Малышев</td> <td>Мягкая</td> <td>23</td> <td>Нагибина</td> </tr> <tr> <td>Малышева</td> <td>Ирина</td> <td></td> <td>ЮЗГУ</td> </tr> <tr> <td>Малышев</td> <td>Витальевна</td> <td></td> <td>ПО-81з</td> </tr> </table>		Логотип работы	Мягкая И.В.	Подпись/Дата	Руководитель	Малышев А.В.		Вице-руководитель	Нагибина А.А.		Научный консультант			Научный руководитель			Координатор			Компьютерный специалист			Контактный телефон			Адрес электронной почты			Факс			Сайт			Логотип	Имя	Номер	Московский	Борисов	Лисова	1	Гусев	Малышев	Мягкая	23	Нагибина	Малышева	Ирина		ЮЗГУ	Малышев	Витальевна		ПО-81з
Логотип работы	Мягкая И.В.	Подпись/Дата																																																				
Руководитель	Малышев А.В.																																																					
Вице-руководитель	Нагибина А.А.																																																					
Научный консультант																																																						
Научный руководитель																																																						
Координатор																																																						
Компьютерный специалист																																																						
Контактный телефон																																																						
Адрес электронной почты																																																						
Факс																																																						
Сайт																																																						
Логотип	Имя	Номер	Московский																																																			
Борисов	Лисова	1	Гусев																																																			
Малышев	Мягкая	23	Нагибина																																																			
Малышева	Ирина		ЮЗГУ																																																			
Малышев	Витальевна		ПО-81з																																																			

Рисунок А.1 – Сведения о ВКРБ

0001СМОГО601999992348	<h2>Цель и задачи разработки</h2>	2																																				
<p>Цель настоящей работы – разработка и внедрение web-сайта для продвижения компании ООО «Русатом – Аддитивные технологии».</p> <p>Для достижения поставленной цели необходимо решить следующие задачи:</p> <ol style="list-style-type: none"> <li>1. Создание информационных разделов сайта «О компании», «Продукция», «Услуги», «Рассчитать стоимость изготовления детали», «Пресс-центр», «Импортозамещение», «Контакты».</li> <li>2. Реализация формы для обратной связи.</li> <li>3. Реализация калькулятора расчета стоимости изготовления деталей.</li> <li>4. Реализация формы заявки на изготовление деталей.</li> <li>5. Создание удобного поиска по сайту.</li> </ol>																																						
<table border="1" style="margin-left: auto; margin-right: 0;"> <tr> <td colspan="3" style="text-align: center;">БКРБ 2068443.09.03.04.23008</td> </tr> <tr> <td style="width: 33%;">Автор работы:</td> <td style="width: 33%;">Именин И.В.</td> <td style="width: 33%;">Подпись / Дата</td> </tr> <tr> <td>Руководитель:</td> <td>Молчанов А.В.</td> <td></td> </tr> <tr> <td>Нормоконтроль:</td> <td>Чагинян А.А.</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td colspan="3" style="text-align: right;">Цель и задачи разработки</td> </tr> <tr> <td colspan="3" style="text-align: right;">Лим. Просса. Писцово.</td> </tr> <tr> <td colspan="3" style="text-align: right;">Лист 2 / Листов 23</td> </tr> <tr> <td colspan="3" style="text-align: right;">Выпускная квалификационная работа бакалавра</td> </tr> <tr> <td colspan="3" style="text-align: right;">ЮЗГУ ПО-81з</td> </tr> </table>			БКРБ 2068443.09.03.04.23008			Автор работы:	Именин И.В.	Подпись / Дата	Руководитель:	Молчанов А.В.		Нормоконтроль:	Чагинян А.А.											Цель и задачи разработки			Лим. Просса. Писцово.			Лист 2 / Листов 23			Выпускная квалификационная работа бакалавра			ЮЗГУ ПО-81з		
БКРБ 2068443.09.03.04.23008																																						
Автор работы:	Именин И.В.	Подпись / Дата																																				
Руководитель:	Молчанов А.В.																																					
Нормоконтроль:	Чагинян А.А.																																					
Цель и задачи разработки																																						
Лим. Просса. Писцово.																																						
Лист 2 / Листов 23																																						
Выпускная квалификационная работа бакалавра																																						
ЮЗГУ ПО-81з																																						

Рисунок А.2 – Цель и задачи разработки



Рисунок А.3 – Концептуальная модель сайта



Рисунок А.4 – Еще плакат

## ПРИЛОЖЕНИЕ Б

### Фрагменты исходного кода программы

#### main.tex

```
1 \input{setup.tex}
2
3 % Режим шаблона (должен быть включен один из трех)
4 \BKPtrue
5 \%Практикtrue
6 \%Курсоваяtrue
7
8 \newcommand{\Дисциплина}{<<Проектирование и архитектура программных систем>>}
9   % для курсовой
10 \newcommand{\КодСпециальности}{09.03.04} % Курсовая
11 \newcommand{\Специальность}{Программная инженерия} % Курсовая
12 \newcommand{\Тема}{Интеллектуальная система распознавания и классификации
13   возгораний,} % ВКР Курсовая
14 \newcommand{\ТемаВтораяСтрока}{ полученных с БПЛА}
15 \newcommand{\ГдеПроводитсяПрактика}{ООО «МЦОБ. Онлайн-сервисы»} % для
16   практики
17 \newcommand{\РуководительПрактПредпр}{} % для практики
18 \newcommand{\ДолжнРуководительПрактПредпр}{} % для практики
19 \newcommand{\РуководительПрактУнивер}{Чаплыгин А. А.} % для практики
20 \newcommand{\ДолжнРуководительПрактУнивер}{к.т.н. доцент} % для практики
21 \newcommand{\Автор}{Д. А. Каракчиев}
22 \newcommand{\АвторРод}{Каракчиева Д.А.}
23 \newcommand{\АвторПолностьюРод}{Каракчиева Даниила Александровича} % для
24   практики
25 \newcommand{\Шифр}{20-06-0391}
26 \newcommand{\Курс}{4} % для практики
27 \newcommand{\Группа}{ПО-026}
28 \newcommand{\Руководитель}{Р. А. Томакова} % для ВКР и курсовой
29 \newcommand{\Нормоконтроль}{А. А. Чаплыгин} % для ВКР
30 \newcommand{\ЗавКаф}{А. В. Малышев} % для ВКР
31 \newcommand{\ДатаПриказа}{«04» апреля 2024~г.} % для ВКР
32 \newcommand{\НомерПриказа}{1616-с} % для ВКР
33 \newcommand{\СрокПредоставления}{«11» июня 2024~г.} % для ВКР, курсового
34
35 \begin{document}
36 \maketitle
37 \ifПрактика{}\else{
38   \input{ЛистЗадания}
39   \input{Реферат}\fi
40 \tableofcontents
41 \input{Обозначения}
42 \ifПрактика{}\else{\input{Введение}}\fi
43 \input{Анализ}
44 \input{ТехЗадание}
45 \input{ТехПроект}
46 \ifПрактика{}\else{
47   \input{РабочийПроект}
48   \input{Заключение}
49 }\fi
```

```
46 \input{СписокИсточников}
47 \ifBKP{\input{Плакаты}}\fi
48 \ifПрактика{}\else{\input{Код}}\fi
49 \end{document}
```

## TexПроект.tex

```
1 \section{Технический проект}
2 \subsection{Общая характеристика организации решения задачи}
3
4 Целью этого проекта является спроектировать и разработать приложение, которое
    поможет специализированным службам по тушению всевозможных возгораний
    вести более эффективную и проработанную деятельность.
5
6 Это приложение представляет собой интеллектуальную систему, предназначенную
    для автоматического обнаружения и классификации очагов возгораний. Эта
    система способна, при помощи нейронной сети, распознавать объекты, такие
    как возгорания и пожары, и определять их класс и уровень опасности на
    основе цветовых характеристик и наличия задымленности на изображении, а
    также обучаться при помощи пользователя.
7
8 \subsection{Обоснование выбора технологии проектирования}
9
10 Для задачи классификации возгораний, полученных с БПЛА, используется две
    сверточные нейронные сети. Такие сети хорошо подходят для задачи обработки
    изображений и классификации объектов, что делает их эффективным выбором
    для анализа данных с камер БПЛА. Она способна извлекать характерные
    признаки из изображений, таких как формы, текстуры и цвета, что позволяет
    эффективно классифицировать возгорания.
11
12 \subsubsection{Описание используемых технологий и языков программирования}
13
14 В процессе разработки приложения используются программные средства и языки
    программирования. Каждое программное средство и каждый язык
    программирования применяется для круга задач, при решении которых они
    необходимы.
15
16 \subsubsection{Сверточные нейронные сети}
17
18 Convolutional neural network(CNN) или Сверточная нейронная сеть - это тип
    искусственной нейронной сети, который широко используется для задач
    обработки изображений и компьютерного зрения. Ключевой особенностью CNN
    является использование сверточных слоев, которые извлекают характерные
    признаки из входных изображений. Эти признаки могут включать в себя формы,
    текстуры, края или цвета.
19
20 Основная идея CNN заключается в применении сверточных фильтров к входному
    изображению, что позволяет выявить повторяющиеся шаблоны или признаки. Эти
    фильтры скользят по изображению, извлекая информацию на разных уровнях
    абстракции. Затем эта информация проходит через дополнительные слои, такие
    как подвыборка или пулинг, которые уменьшают пространственные размеры
    данных, и полностью подключенные слои, которые выполняют классификацию или
    регрессию.
21
```

22 CNN показали впечатляющие результаты в задачах классификации изображений, обнаружения объектов, сегментации и даже в анализе медицинских изображений . Они эффективно обрабатывают большие объемы данных, обучаясь выявлять сложные зависимости и характерные признаки.

23

24 \subsubsection{Машинное обучение}

25

26 Машинное обучение - это раздел искусственного интеллекта, который фокусируется на разработке алгоритмов и моделей, позволяющих компьютерам обучаться и улучшать свои задачи без явного программирования.

27

28 Применение алгоритмов машинного обучения лежит в основе нашей системы анализа и классификации данных. Мы обучаем нашу модель на обширной базе изображений, позволяя системе эффективно распознавать и классифицировать объекты в реальном времени. Этот процесс включает в себя использование сложных алгоритмов, которые могут извлекать и интерпретировать характерные особенности из данных изображений.

29

30 \subsubsection{Язык программирования Python}

31

32 Python является одним из самых популярных и широко используемых языков программирования для разработки приложений искусственного интеллекта и машинного обучения. Он имеет простой и понятный синтаксис, что ускоряет процесс разработки и делает код более читаемым.

33

34 \subsubsection{Библиотеки Python}

35

36 \begin{itemize}

37 \item NumPy - фундаментальная библиотека для научных расчетов в Python. Она обеспечивает эффективную работу с многомерными массивами и матричными вычислениями, что критически важно для обработки и манипуляции данными.

38 \item Matplotlib - библиотека для визуализации данных. Она позволяет создавать настраиваемые и интуитивно понятные графики, диаграммы и изображения, что облегчает визуальный анализ данных и представление результатов.

39 \item OpenCV (Open Source Computer Vision Library) - библиотека компьютерного зрения, которая предлагает широкий спектр алгоритмов для обработки изображений и видео. Она идеально подходит для задач обработки изображений , обнаружения объектов и анализа видео, полученных с камер БПЛА.

40 \item Pandas - библиотека для анализа и манипуляции данными. Она предоставляет удобные структуры данных, такие как DataFrame, и богатый набор инструментов для обработки, фильтрации и агрегации данных, упрощая подготовку и анализ больших наборов данных.

41 \item TensorFlow - это открытая платформа машинного обучения с масштабируемыми инструментами для обучения и развертывания моделей. Она обеспечивает гибкую и эффективную инфраструктуру для создания сложных нейронных сетей. Keras - это высокоуровневый API, построенный на TensorFlow, который упрощает процесс создания и обучения нейронных сетей . Он предлагает простой и интуитивно понятный интерфейс, позволяя быстро разрабатывать и экспериментировать с различными архитектурами моделей.

42 \end{itemize}

43

44 \subsubsection{Архитектура сверточной нейронной сети}

45

46 Архитектура сверточной нейронной сети включает в себя 14 слоев с различными функциями.  
 47  
 48 Схема архитектуры нейронной сети представлена на рисунке ~\ref{archNC:image}.  
 49  
 50 `\begin{figure}[H]`  
 51 `\center{\includegraphics[width=1\linewidth]{archNC}}`  
 52 `\caption{Архитектура нейронной сети}`  
 53 `\label{archNC:image}`  
 54 `\end{figure}`  
 55  
 56 Слои нейронной сети классификации описаны далее:  
 57  
 58 `\begin{enumerate}`  
 59 `\item Input Layer (Входной слой)`  
 60  
 61 Функция Input определяет входной слой с размером изображения 128×128 пикселей и 3 цветовыми каналами (RGB).  
 62 `\item Convolutional Layers (Сверточные слои, 9)`  
 63  
 64 Первый свёрточный слой Conv2D имеет 32 фильтра размером 3×3, функцию активации ReLU и параметр padding='same', который сохраняет размерность входного изображения.  
 65  
 66 Второй свёрточный слой Conv2D увеличивает количество фильтров до 64 и применяет шаг (strides) равный 2, что уменьшает размерность изображения вдвое.  
 67  
 68 Следующие два слоя Conv2D также имеют 64 фильтра и один из них снова применяет шаг 2 для уменьшения размерности.  
 69  
 70 Пятый и шестой свёрточные слои Conv2D содержат 128 фильтров каждый, с шагом 2 на шестом слое.  
 71  
 72 Седьмой свёрточный слой Conv2D сохраняет 128 фильтров и размерность.  
 73  
 74 Восьмой и девятый свёрточные слои Conv2D увеличивают количество фильтров до 256, с шагом 2 на восьмом слое.  
 75 `\item Flatten Layer (Выравнивающий слой)`  
 76  
 77 Слой Flatten преобразует многомерный тензор свёрточных слоёв в одномерный, чтобы его можно было подать на полно связные слои.  
 78 `\item Dropout Layer (Слой исключения)`  
 79  
 80 Слой Dropout с параметром 0.2 предотвращает переобучение, случайным образом "выключая" 20% нейронов на каждом шаге обучения.  
 81 `\item Dense Layers (Полно связные слои, 2)`  
 82  
 83 Первый полно связный слой Dense имеет 256 нейронов и функцию активации ReLU.  
 84  
 85 Второй полно связный слой Dense формирует выходной слой с 30 нейронами, количество которых соответствует количеству выходных значений, которые должна предсказывать модель.  
 86 `\end{enumerate}`

```

87
88 \subsubsection{Функция активации ReLU}
89
90 ReLU, или Rectified Linear Unit, — это функция активации, которая
    используется в нейронных сетях для увеличения нелинейности.
91 Формула ReLU представлена формулой ~\ref{form0:equation}.
92
93 \begin{equation}
94     \text{\footnotesize{17}}\text{\normalsize{20}}\{f(x)=\max(0,x)\}
95     \label{form0:equation}
96 \end{equation}
97
98 Это означает, что если вход  $x$  положительный, то функция возвращает  $x$ , а если
     $x$  отрицательный, то функция возвращает 0.
99 ReLU популярна потому, что она ускоряет обучение нейронных сетей без
    значительной потери точности. Она также помогает решить проблему
    исчезающего градиента, так как производная для положительных значений
    всегда равна 1, что обеспечивает более быстрое и эффективное обучение.
100
101 График функции ReLU предоставлен на изображении ~\ref{relu:image}.
102
103 \begin{figure}[H]
104 \center{\includegraphics[width=1\linewidth]{relu}}
105 \caption{График функции ReLU}
106 \label{relu:image}
107 \end{figure}
108
109 \subsubsection{Функция IoU Loss}
110 Intersection over Union (IoU) является метрикой, используемой для измерения
    точности объектного детектора на определенном наборе данных. Если мы
    работаем с задачами компьютерного зрения, такими как сегментация
    изображений или обнаружение объектов, IoU может помочь оценить, насколько
    предсказанные границы объекта соответствуют истинным границам.
111
112 Наглядным образом функцию можно увидеть на изображении ~\ref{iou:image}.
113
114 \begin{figure}[H]
115 \center{\includegraphics[width=1\linewidth]{iou}}
116 \caption{График функции IoU}
117 \label{iou:image}
118 \end{figure}
119
120 IoU рассчитывается по формуле ~\ref{form1:equation}.
121
122 \begin{equation}
123     \text{\footnotesize{17}}\text{\normalsize{20}}\{IoU= \frac{\text{площадь объединения}}{\text{площадь}}
        \text{пересечения}}\}
124     \label{form1:equation}
125 \end{equation}
126
127 Где:
128 \begin{itemize}
129     \item Площадь пересечения - это область, где предсказанная граница и истинная
        граница объекта перекрываются.

```

```

130 \item Площадь объединения - это область, покрытая как предсказанной границей,
      так и истинной границей, вместе взятых.
131 \end{itemize}
132
133 IoU loss - это функция потерь, которая используется для обучения моделей,
      выполняющих задачи, связанные с локализацией объектов. Вместо того чтобы
      использовать стандартные функции потерь, такие как кросс-энтропия, которые
      могут не полностью отражать точность локализации, IoU loss напрямую
      оптимизирует метрику IoU, стремясь увеличить площадь пересечения и
      уменьшить площадь объединения.
134
135 Функция потерь IoU показана в формуле ~\ref{form2:equation}.
136
137 \begin{equation}
138   \text{\footnotesize{17}}\text{\normalsize{20}}\text{IoU loss} = 1 \text{IoU}
139 \label{form2:equation}
140 \end{equation}
141
142 Таким образом, минимизация IoU loss в процессе обучения приводит к увеличению
      IoU между предсказанными и истинными границами, что помогает в задачах
      локализации объектов нашего проекта.
143
144 \subsection{Диаграмма компонентов}
145
146 Диаграмма компонентов представляет структуру системы в виде набора
      компонентов и их взаимосвязей. Каждый компонент отвечает за определенную
      функцию в рамках системы и может включать в себя подсистемы или модули.
147
148 \begin{itemize}
149 \item Графический интерфейс: отвечает за создание интерфейса, в котором
      пользователь наглядно видит результат распознавания в сравнении с
      оригиналом;
150 \item Обработка изображения: включает в себя методы улучшения качества
      изображений, такие как коррекция освещения, удаление шумов и извлечение
      важных признаков;
151 \item Сверточная нейронная сеть (CNN): Ядро этой системы, реализующее
      алгоритмы обучения и распознавания объектов;
152 \item Данные параметров нейронной сети: представляют собой обученную модель,
      которая хранит в себе веса и параметры, извлеченные из данных во время
      процесса обучения;
153 \item Классификация объекта: этот модуль используется для классификации
      возгорания;
154 \item Генерация отчета: отвечает за создание отчета для выведения класса
      возгорания и оценки его опасности.
155 \end{itemize}
156
157 \subsubsection{Взаимодействие компонентов}
158
159 \begin{enumerate}
160 \item Пользователь загружает изображение через графический интерфейс
      пользователя.
161 \item Интерфейс передает изображение в модуль обработки изображения.
162 \item После обработки данные передаются в модуль сверточной нейронной сети
      для распознавания объекта.

```

```

163 \item Модуль данных параметров передает веса и параметры и корректирует
      работу нейронной сети.
164 \item Результаты распознавания классифицируются в модуле классификации
      объекта.
165 \item Модуль генерации отчета выводит данные о возгорании в графическом
      интерфейсе.
166 \end{enumerate}

167
168 Диаграмма компонентов представлена на рисунке ~\ref{comp:image}.

169
170 \begin{figure}[H]
171 \center{\includegraphics[width=1\linewidth]{comp}}
172 \caption{Диаграмма компонентов}
173 \label{comp:image}
174 \end{figure}

175
176 \subsubsection{Диаграмма программных классов}

177
178 Точкой входа в программу является класс main. В этом классе осуществляется
      запуск и инициализация основных компонентов программы:
179 \begin{itemize}
180     \item dataprocessing - компонент, отвечающий за обработку данных и
          изображений.
181     \item creatingtfrecordclassifier - компонент, отвечающий за создание и
          загрузку изображений и создания их записи в формате .tfrecord
182     \item creatingtfrecordlocalizer - компонент, отвечающий за создание и
          загрузку изображений и создания их записи в формате .tfrecord
183     \item classifier - компонент, отвечающий за работу с нейронной сетью по
          классификации данных, её обучение и тестирование.
184     \item training - компонент, отвечающий за работу с нейронной сетью по
          распознанию данных, её обучение и тестирование.
185 \end{itemize}

186
187 Диаграмма классов представлена на рисунке ~\ref{classdiag:image}.

188
189 \begin{figure}[H]
190 \center{\includegraphics[width=1\linewidth]{classdiag}}
191 \caption{Диаграмма классов и их связей}
192 \label{classdiag:image}
193 \end{figure}

194
195 На диаграмме показаны данные связи:
196 \begin{itemize}
197     \item Связь dataprocessing - main. Класс main использует основные функции
          класса dataprocessing, такие как отображение информации о изображении,
          загрузка изображения и нормализация координат изображения.
198     \item Связь dataprocessing - creatingtfrecordclassifier. Класс
          creatingtfrecordclassifier использует основные функции класса
          dataprocessing, такие как загрузка изображения и нормализация координат
          изображения. Для создания записи tfrecord необходимо создать запись со
          всеми изображениями и файлами формата .xml.
199     \item Связь dataprocessing - creatingtfrecordlocalizer. Класс
          creatingtfrecordlocalizer использует основные функции класса
          dataprocessing, такие как загрузка изображения и нормализация координат

```

изображения. Для создания записи `tfrecord` необходимо создать запись со всеми изображениями и файлами формата `.xml`.

200 \item Связь `creatingtfrecordclassifier` - `classifier`. Для работы классу `classifier` нужен созданный в классе `creatingtfrecordclassifier` файл `tfrecord`.

201 \item Связь `creatingtfrecordclassifier` - `main`. Класс `main` использует основные функции класса `creatingtfrecordclassifier`, такие как создание файла `tfrecord` для классификатора.

202 \item Связь `creatingtfrecordlocalizer` - `training`. Для работы классу `training` нужен созданный в классе `creatingtfrecordlocalizer` файл `tfrecord`.

203 \item Связь `creatingtfrecordlocalizer` - `main`. Класс `main` использует основные функции класса `creatingtfrecordlocalizer`, такие как создание файла `tfrecord` для локализатора.

204 \item Связь `classifier` - `main`. Класс `main` использует основные функции класса `classifier`, такие как тестирование, обучение, загрузка и сохранение сети.

205 \item Связь `localizer` - `main`. Класс `main` использует основные функции класса `localizer`, такие как тестирование, обучение, загрузка и сохранение сети

206 \end{itemize}

207

208

209 \subsection{Проектирование пользовательского интерфейса}

210 \% накинуть ссылок на литературу

211 На основании требований к пользовательскому интерфейсу, представленных в пункте 2.3 технического задания, был разработан графический интерфейс приложения. Для создания пользовательского интерфейса используется библиотека `tkinter` и `matplotlib`.

212

213 На рисунке `\ref{maketinterface:image}` представлен макет интерфейса окон для распознания и обучения нейронной сети. Данный макет содержит следующие элементы:

214 \begin{enumerate}

215 \item Загрузка изображения из системы.

216 \item Загрузка своей модели нейронной сети.

217 \item Запустить распознание изображения

218 \item Открыть окно тренировки модели нейронной сети.

219 \item Поле загруженного изображения.

220 \item Поле изображения с распознанным возгоранием.

221 \item Закрывает окно тренировки модели нейронной сети.

222 \item Показывает информацию о первой картинке в папке с изображениями для обучения.

223 \item Запускает стороннюю программу `labelimg`.

224 \item Показывает в поле для вывода информацию о наличии файлов.

225 \item Создает новую запись `tfrecord` для локализатора.

226 \item Создает новую запись `tfrecord` для классификатора.

227 \item Запускает функцию тестирования классификатора.

228 \item Запускает другую функцию тестирования классификатора с точными значениями.

229 \item Запускает функцию обучения классификатора.

230 \item Сохраняет модель нейронной сети классификатора.

231 \item Сохраняет модель нейронной сети локализатора.

232 \item Загружает модель нейронной сети локализатора.

```
233 \item Запускает функцию тестирования нейронной сети локализатора.  
234 \item Запускает функции для обучения нейронной сети локализатора.  
235 \item Поле вывода информации из терминала после выполнения любых операций.  
236 \end{enumerate}  
237  
238 \begin{figure}[H]  
239 \center{\includegraphics[width=1\linewidth]{maketinterface}}  
240 \caption{Макет интерфейса окна распознавания и окна обучения нейронной сети}  
241 \label{maketinterface:image}  
242 \end{figure}
```

## **Место для диска**