

**Оглавление**

[Введение 4](#_Toc199353059)

[1 Анализ предметной области 7](#_Toc199353060)

[1.1 Описание сферы применения компьютерного зрения при распознавании шахматных фигур 7](#_Toc199353061)

[1.2 Анализ структуры, регламентов подразделения ЗАО «ЕС-ЛИЗИНГ», регулирующих выполнение выбранного бизнес-процесса 8](#_Toc199353062)

[1.3 Роль и востребованность интеллектуальных систем распознавания шахматных фигур 8](#_Toc199353063)

[1.4 Моделирование бизнес-процесса «КАК ЕСТЬ» 9](#_Toc199353064)

[1.5 Анализ рынка программного продукта и анализ к своей системе 10](#_Toc199353065)

[1.6 Анализ стейкхолдеров и их требований к разрабатываемой системе 14](#_Toc199353066)

[1.7 Выбор средств разработки 16](#_Toc199353067)

[1.8 Техническое задание на разрабатываемое ПО представлено в Приложении 1. 18](#_Toc199353068)

[1.9 Выводы по разделу 18](#_Toc199353069)

[2 Проектная часть 21](#_Toc199353070)

[2.1 Разработка прототипа модели глубокого обучения 21](#_Toc199353071)

[2.2 Формирование набора данных 22](#_Toc199353072)

[2.3 Выбор модели 23](#_Toc199353073)

[2.4 Процесс обучения модели 24](#_Toc199353074)

[2.5 Выводы по разделу 25](#_Toc199353075)

[3 Frontend-разработка 28](#_Toc199353076)

[3.1 Проектирование и функциональная структура системы классификации шахматных фигур 28](#_Toc199353077)

[3.2 Разработка пользовательский приложений 31](#_Toc199353078)

[3.3 Веб-приложение 31](#_Toc199353079)

[3.4 Десктопное приложение 34](#_Toc199353080)

[3.5 Доступ к данным 37](#_Toc199353081)

[3.6 Выводы по разделу 38](#_Toc199353082)

[ЭКОНОМИЧЕСКАЯ ЧАСТЬ 40](#_Toc199353083)

[4 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИС 40](#_Toc199353084)

[4.1 Организация и планирование работ по теме 40](#_Toc199353085)

[4.2 Расчет стоимости проведения работ по теме 43](#_Toc199353086)

[4.3 Выводы по главе 3 46](#_Toc199353087)

[Заключение 47](#_Toc199353088)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 52](#_Toc199353089)

[5 Приложение 1. Техническое задание на тему разрботкаинтеллектуальной системы классификации шахматных фигур 53](#_Toc199353090)

[1 Цель и назначение создания автоматизированной системы 53](#_Toc199353091)

[2 Требования 54](#_Toc199353092)

[4 Требования к организации входных данных 54](#_Toc199353093)

[9 Этапы разработки 55](#_Toc199353094)

[Приложение 2 Исходный код приложения 56](#_Toc199353095)

# Введение

В настоящем мире информационные технологии активно внедряются в различные сферы деятельности, значительно упрощая выполнение задач и оптимизируя рутинные процессы. Одной из таких сфер является шахматы — игра, которая требует высокого уровня стратегического мышления, анализа и обработки данных. В условиях роста популярности шахмат, как образовательного инструмента и спортивной дисциплины возникает необходимость автоматизации процессов анализа профессиональных шахматных партий, обучения новичков.

Одной из ключевых задач в этой области является распознавание шахматных фигур на изображениях реальных шахматных досок. Это сложная техническая задача, требующая применения современных методов компьютерного зрения и машинного обучения. Традиционные подходы к анализу шахматных партий часто ограничиваются ручным вводом данных или использованием устаревших алгоритмов, которые не способны адаптироваться к сложным сценариям, таким как различие между похожими фигурами, обработка изображений с различным фоном, углами съемки и освещением.

Современные технологии позволяют решать такие задачи с высокой точностью. Например, использование нейронных сетей, свёрточные архитектур (CNN) и глубокого обучения открывает новые возможности для автоматизации анализа шахматных партий. Интеллектуальная система, основанная на этих технологиях, может автоматически определять тип фигуры (пешка, конь, слон, ладья, ферзь, король) и её цвет (белый или черный), что значительно упрощает работу аналитиков, тренеров и любителей шахмат.

Актуальность данной работы обусловлена несколькими факторами:

1. **Образовательная потребность**: Шахматы активно используются в образовательных программах для развития логического мышления, стратегического планирования и аналитических навыков. Автоматизация анализа партий позволяет создавать персонализированные учебные материалы для студентов.
2. **Спортивная сфера:** Профессиональные шахматисты и тренеры нуждаются в инструментах для анализа партий, выявления ошибок и разработки новых стратегий.
3. **Технологическая востребованность:** Разработка интеллектуальных систем распознавания шахматных фигур может быть интегрирована в мобильные приложения, онлайн-платформы и системы дополненной реальности, что расширяет их функциональность и удобство использования.

Кроме того, современные методы машинного обучения и компьютерного зрения находят широкое применение в различных областях, таких как медицина, транспорт, финансы и розничная торговля. Например, в медицине они используются для анализа рентгеновских снимков, в транспорте — для создания автономных транспортных средств, а в розничной торговле — для анализа покупательского поведения. Подобные технологии могут быть успешно применены и в шахматах, где требуется точное распознавание объектов на изображениях.

Одной из основных технологий, используемых в данной работе, являются сверточные нейронные сети (Convolutional Neural Networks, CNN). Эти сети особенно эффективны для задач, связанных с обработкой изображений, так как они способны автоматически извлекать признаки из данных и использовать их для классификации. В частности, CNN могут быть обучены распознавать шахматные фигуры на основе их визуальных характеристик, таких как форма, цвет и текстура.

Важно отметить, что разработка интеллектуальной системы распознавания шахматных фигур имеет не только теоретическую, но и практическую значимость. Например, такая система может быть использована в образовательных платформах для автоматизации анализа партий и создания интерактивных учебных материалов. Она также может быть интегрирована в мобильные приложения для любителей шахмат, что позволит им анализировать свои партии и улучшать навыки игры.

Целью данной выпускной квалификационной работы является разработка интеллектуальной системы распознавания шахматных фигур, способной автоматически определять тип фигуры (пешка, конь, слон, ладья, ферзь, король) и её цвет (белый или черный) с использованием современных методов машинного обучения и компьютерного зрения.

Для достижения поставленной цели необходимо решить следующие задачи

* Провести анализ предметной области для выявления основных требований к системе.
* Выбрать подходящие методы машинного обучения и компьютерного зрения.
* Разработать модель, способную работать с различными условиями освещения, углами съемки и фонами.
* Создать удобный пользовательский интерфейс для взаимодействия с системой.
* Оценить эффективность разработанной системы и её применимость в реальных сценариях.

Объект исследования: процессы автоматизированного анализа и классификации шахматных фигур на изображениях.  
Предмет исследования: программная система распознавания шахматных фигур, основанная на методах глубокого обучения.

# Анализ предметной области

Ссылка на GitHub: <https://github.com/Yed1x/Diplom>

## Описание сферы применения компьютерного зрения при распознавании шахматных фигур

Компьютерное зрение представляет собой одну из наиболее активно развивающихся областей искусственного интеллекта, ориентированную на обработку и анализ цифровых изображений. В последние годы оно стало неотъемлемой частью многих информационных систем, позволяя автоматизировать задачи, ранее выполняемые человеком. Среди таких задач — классификация объектов, распознавание текста, сегментация изображений и другие системы. Особенно актуально применение технологий компьютерного зрения в тех случаях, когда требуется высокая точность и устойчивость к изменению внешних условий, таких как освещение, ракурс съёмки и фоновые помехи.

Одним из перспективных направлений использования компьютерного зрения является автоматизация анализа шахматных партий. Современные шахматные приложения всё чаще сталкиваются с необходимостью распознавать фигуры на фотографиях реальных шахматных досок. Это позволяет пользователям быстро получать информацию о текущей позиции, строить анализ игры и использовать её для обучения или подготовки к соревнованиям. Однако задача распознавания шахматных фигур на изображениях является сложной и требует применения современных методов машинного обучения, способных эффективно решать такие задачи.

Основным подходом к решению задачи распознавания объектов на изображениях или компьютерном зрением, являются сверточные нейронные сети (CNN), которые демонстрируют высокую точность в задачах классификации и анализа визуальных данных. Эти нейронные сети, способны автоматически извлекать признаки из изображений, что делает их идеальным выбором для решения задач в области компьютерного зрения.

## Анализ структуры, регламентов подразделения ЗАО «ЕС-ЛИЗИНГ», регулирующих выполнение выбранного бизнес-процесса

Для успешного внедрения разрабатываемого программного обеспечения необходимо провести анализ действующих нормативных документов ЗАО «ЕС-ЛИЗИНГ».

Деятельность подразделения регулируется внутренними положениями, такими как:

* Положение о применении ИКТ
* Положение о научно-методической деятельности
* Регламенты использования программного обеспечения
* Документы по информационной безопасности и защите данных

Эти документы устанавливают требования к качеству порядку применения цифровых технологий, хранению информации и обеспечению её доступности. При этом особое внимание уделяется тому, чтобы внедряемые технологии соответствовали целям и задачам организации, были совместимы с имеющейся инфраструктурой и соответствовали стандартам защиты данных.

## Роль и востребованность интеллектуальных систем распознавания шахматных фигур

Возьмём в пример наш клуб «Любители шахмат» в компании ЗАО «ЕС-ЛИЗИНГ» Он играет важную роль для развития аналитический решений и приятного время провождения в перерыв.

В связи с ростом популярности шахмат как инструмента развития логического мышления и аналитических навыков, они помогают отвлечься от работы.

Интеллектуальные системы распознавания шахматных фигур позволяют эффективно решать задачи анализа шахматных партий, для анализа стратегий ходов фигур.

Применение таких систем значительно упрощает обработку графических материалов. Традиционные методы требуют ручного анализа изображений, что занимает много времени и сопряжено с риском человеческих ошибок. Система, основанная на компьютерном зрении и глубоком обучении, позволяет сократить временные затраты, повысить точность анализа и создать условия для автоматической генерации обучающих материалов. Это особенно важно при подготовке интерактивных курсов, где требуется оперативное получение информации о текущей позиции на доске.

Кроме выше сказанного, внедрение технологии распознавания способствует улучшению пользовательского опыта. Образовательные платформы могут использовать информацию о типе и цвете фигур для фильтрации партий, поиска похожих позиций и построения рекомендаций. Такие функции делают платформу более удобной и функциональной, особенно для начинающих игроков, которым сложно самостоятельно классифицировать сложные шахматные позиции.

Важным аспектом является возможность персонализации обучения. На основе автоматически полученных данных система может предлагать индивидуальные задания, отслеживать прогресс пользователя и корректировать уровень сложности материалов. Это позволяет адаптировать обучение под уровень подготовки каждого студента, повышая эффективность освоения материала и мотивацию к дальнейшему обучению.

Интеграция системы распознавания шахматных фигур в инфраструктуру шахматного клуба "Любители шахмат" при ЗАО «ЕС-ЛИЗИНГ» обеспечивает стабильную работу и взаимодействие с другими компонентами цифровой среды клуба. Получаемые данные используются не только для отображения на пользовательском интерфейсе, но и для ведения статистики партий, аналитики, формирования рекомендаций. Таким образом, система становится важным элементом клубной цифровой экосистемы, способствуя развитию интереса к шахматам и повышению уровня игры среди сотрудников компании.

## Моделирование бизнес-процесса «КАК ЕСТЬ»

Моделирование бизнес-процесса "КАК ЕСТЬ" по методике IDEF0, позволило описать текущее состояние обработки графических материалов в ЗАО "ЕС-ЛИЗИНГ" и шахматном клубе "Любители шахмат". Основной узел диаграммы отражает следующие этапы:

* Приём изображения шахматной доски.
* Выделение характерных признаков: форма, цвет. Обработка и классификация.
* Передача результатов в образовательную платформу или архив.

На начальном этапе проводится проверка качества изображения и исключение дубликатов. Затем осуществляется анализ позиции, выделение ключевых характеристик и их сопоставление с эталонными значениями. Информация передаётся в виде меток, которые используются для организации данных, поиска и создания учебных материалов.

Данный подход к моделированию обеспечивает целостное представление о логике работы системы, хотя не детализирует маршруты перемещения информации внутри Клуба "Любители шахмат". Тем не менее, он служит основой для построения модели "КАК ДОЛЖНО БЫТЬ", в которой будут указаны пути оптимизации и автоматизации.

Таким образом, внедрение интеллектуальной системы распознавания шахматных фигур имеет большое значение для деятельности Центра дистанционных образовательных технологий. Она позволяет повысить качество обучения, улучшить взаимодействие с графическими данными и автоматизировать процессы, ранее выполняемые вручную. Это делает систему актуальным элементом цифровой трансформации и усиливает её конкурентоспособность в сфере онлайн-образования.

## Анализ рынка программного продукта и анализ к своей системе

Для реализации интеллектуальной системы распознавания шахматных фигур важно провести анализ существующих решений, доступных на рынке программного обеспечения. Это позволяет определить уровень зрелости технологий, выявить наиболее популярные и эффективные решения, а также обосновать целесообразность создания собственной системы. На сегодняшний день существует ряд программ и приложений, предназначенных для автоматического анализа шахматных позиций и классификации фигур. Они применяются как в образовательной среде, так и в спортивной практике. Ниже приведён обзор наиболее известных систем, решающих задачи, близкие к той, которая реализована в рамках преддипломной работы.

**Рассмотри аналоги:**

**ChessVisionAI** — это инструмент, позволяющий пользователям загружать изображения шахматных досок и получать информацию о текущей позиции. Система поддерживает работу с мобильными устройствами и интегрируется с платформами для обучения и анализа партий.

Основные функции:

* Распознавание типа фигуры
* Определение цвета фигуры
* Генерация FEN-кода позиции
* Интеграция с онлайн-анализаторами

Преимущества:

* Высокая точность распознавания
* Современный интерфейс
* Поддержка мобильных устройств

Недостатки:

* Высокая стоимость лицензии
* Требуется стабильное интернет-соединение
* Необходимость ручной корректировки в сложных случаях

**ShahmatyPro** — программный продукт, ориентированный на профессиональных игроков и тренеров. Он позволяет анализировать партии, строить рекомендации и сохранять историю игр.

Основные функции:

* Автоматический анализ позиции
* Игровые рекомендации
* Сохранение истории партий
* Визуализация ходов

Преимущества:

* Поддержка офлайн-режима
* Подробная статистика игры
* Возможность сравнения партий

Недостатки:

* Низкая устойчивость к изменению условий освещения
* Ограниченная поддержка пользовательских изображений
* Сложный интерфейс для новичков

**ChessMate** — образовательное приложение, ориентированное на обучение шахматам с использованием графических данных. Позволяет пользователям загружать изображения досок и получать обратную связь по позициям.

Основные функции:

* Классификация фигур на основе изображения
* Вывод рекомендаций по улучшению позиции
* Создание персонализированных учебных материалов

Преимущества:

* Удобный интерфейс
* Возможность использования в обучении
* Простота внедрения в учебные курсы

Недостатки:

* Зависимость от качества изображения
* Отсутствие гибкой настройки параметров
* Ограниченная функциональность в условиях плохого освещения

Как мы можем заметить, прямых аналогов с распознаванием классов фигур – нет.

Таблица 1.1

|  |  |  |
| --- | --- | --- |
| Название системы | Определение типа фигуры | Стоимость годовой лицензии |
| **ChessVisionAI** | да | высокая |
| **ShahmatyPro** | частично | средняя |
| **ChessMate** | да | высокая |

Предлагаемое решение отличается от рассмотренных аналогов следующими особенностями:

* Лёгкость внедрения: система не требует мощного оборудования и может работать на стандартных ПК.
* Высокая устойчивость к внешним условиям: благодаря применению методов аугментации и предобработки изображений модель способна работать даже при низком качестве фото.
* Дополнительная функция определения цвета: в отличие от большинства аналогов, наша система не только определяет тип фигуры, но и её цвет (белая или чёрная).
* Сохранение истории запросов и статистики: система позволяет сохранять данные о предыдущих классификациях и выводить их в виде графиков.
* Экономическая целесообразность: стоимость внедрения и эксплуатации значительно ниже, чем у коммерческих решений.
* Делаем логический вывод: Анализ существующего программного обеспечения показал, что ни одна из известных систем не предлагает полного решения, сочетающего высокую точность, определение цвета фигуры, удобство интерфейса и экономическую доступность.

Разрабатываемая система распознавания шахматных фигур имеет потенциал для внедрения в образовательные и игровые платформы, поскольку сочетает современные технологии машинного зрения и глубокого обучения с простым и понятным интерфейсом.

Разработка системы направлена на закрытие пробела между профессиональными инструментами и потребностями начинающих пользователей, что делает её актуальным решением для автоматизации анализа шахматных партий в различных сферах.

## Анализ стейкхолдеров и их требований к разрабатываемой системе

Разработка интеллектуальной системы распознавания шахматных фигур предполагает создание программного обеспечения, которое будет использоваться различными категориями пользователей. Для того чтобы система соответствовала ожиданиям и была востребована, необходимо учитывать интересы и потребности всех заинтересованных сторон — стейкхолдеров. В ходе анализа были выделены ключевые группы пользователей, чьи требования оказали влияние на выбор архитектуры, функциональности и интерфейса системы.

Первую группу составляют студенты и начинающие шахматисты, которые будут использовать систему в образовательных целях. Для них важна простота взаимодействия с приложением, возможность быстро загрузить изображение и получить точный результат классификации. Также значимым является наличие истории запросов и графиков статистики, что позволяет отслеживать прогресс и анализировать свои партии.

Вторая группа — преподаватели и тренеры по шахматам. Они заинтересованы в том, чтобы система могла сохранять данные о результатах анализа, предоставлять возможности фильтрации и экспорта информации. Это необходимо для создания персонализированных заданий, проверки домашних работ и автоматизации анализа учебных материалов.

Третьей важной группой являются представители Центра дистанционных образовательных технологий университета, отвечающие за внедрение программных решений в образовательную среду. Их требования связаны с совместимостью разработанного ПО с уже используемыми платформами, минимальными затратами на установку и эксплуатацию, а также соблюдением стандартов информационной безопасности и качества предоставляемых услуг.

Четвёртая группа — технический персонал и администраторы, которые занимаются сопровождением программного обеспечения. Для них важно наличие полной технической документации, простота установки и запуска, а также механизм логирования действий программы, который помогает в диагностике ошибок и обслуживании системы.

Пятую группу составляют любители шахмат вне образовательной среды. Эти пользователи ценят удобство, скорость работы и мобильность решения. Для них система должна быть легкой в освоении, не зависеть от интернет-соединения и поддерживать популярные форматы изображений, такие как JPG и PNG.

Учет интересов всех указанных групп позволил сформировать четкие требования к системе, направленные на обеспечение её надежности, точности определения типа и цвета фигуры, удобства интерфейса, возможностей сохранения и визуализации данных. Разрабатываемое решение стремится удовлетворить максимальное количество этих требований, делая систему универсальной и применимой как в образовательной сфере, так и среди широкой аудитории любителей шахмат.

Подведем небольшой итог: при анализе данной работы, действительно, можно найти стейкхолдеров, которые могут быть заинтересованы в дальнейшей покупке.

## Выбор средств разработки

При выборе языка программирования основное внимание уделялось удобству работы с данными, наличию библиотек машинного обучения и простоте интеграции в существующую IT-инфраструктуру. Наиболее подходящим вариантом стал язык Python 3.8+, который широко используется в области машинного обучения и компьютерного зрения благодаря своей читаемости, богатой экосистеме и высокому уровню поддержки сообществом разработчиков. Python позволяет быстро прототипировать модель, использовать предобученные архитектуры глубокого обучения и взаимодействовать с различными библиотеками обработки изображений и визуализации.

Для построения модели машинного обучения были рассмотрены такие фреймворки, как TensorFlow, PyTorch и Keras. Окончательный выбор пал на TensorFlow/Keras, поскольку они обеспечивают устойчивую работу с CNN-сетями, имеют удобный API, хорошо документированы и поддерживают сохранение моделей в формате .h5, что упрощает их дальнейшее использование. Эти инструменты уже применялись в ряде исследовательских проектах.

В качестве среды разработки использовалась комбинация Jupyter Notebook и Visual Studio Code. Jupyter позволял проводить эксперименты с моделью, быстро тестировать изменения и анализировать результаты, тогда как Visual Studio Code был выбран для финальной сборки и интеграции всех компонентов системы. Обе среды являются бесплатными и совместимыми с основными операционными системами (Windows, Linux и MacOS, соблюдая установленные библиотеки и фреймворки).

Разработка пользовательского интерфейса потребовала выбора легковесной и кроссплатформенной библиотеки. После сравнительного анализа были рассмотрены такие библиотеки, как Tkinter, PyQt, Kivy и CustomTkinter. Было принято решение использовать CustomTkinter, поскольку он сочетает в себе современный дизайн, простоту интеграции и минимальные системные требования. Это позволило создать интуитивно понятный графический интерфейс без необходимости установки дополнительных зависимостей или сложных настроек среды.

Также была проведена оценка различных библиотек для обработки изображений: OpenCV, PIL/Pillow, Scikit-image. В проекте была выбрана комбинация PIL и OpenCV, так как они обеспечивают необходимый уровень гибкости, совместимость с популярными фреймворками и стабильную работу с массивами изображений.

Для хранения истории запросов и статистики рассматривались различные подходы: от простых CSV-файлов до полноценных реляционных баз данных. Учитывая простоту задачи и масштаб системы, было решено использовать локальное хранение в формате CSV, которое не требует запуска сервера баз данных и легко интегрируется в текущую архитектуру. Такое решение минимизировало накладные расходы на развертывание и адаптацию системы в рамках компании инфраструктуры.

Сравнение различных подходов к реализации модели машинного обучения показало, что предобученные архитектуры, такие как EfficientNetB0 и MobileNetV2, дают наилучший баланс между точностью и скоростью работы. Для данного проекта была выбрана модель EfficientNetB0 , которая после этапа дообучения (fine-tuning) показала высокую точность классификации как по типу фигуры, так и по её цвету. Также учитывался уровень доступности вычислительных ресурсов — система должна была работать даже на устройствах со средними характеристиками, что ограничило использование более тяжёлых моделей.

Немаловажным фактором стал опыт разработчика и доступность документации. Все выбранные технологии имеют широкую поддержку, большое количество примеров реализации и активное сообщество, что снижает временные затраты на обучение и разработку. Кроме того, Python и его библиотеки совместимы с множеством других систем и могут быть легко интегрированы как в локальные приложения, так и в облачные сервисы.

Таким образом, выбор языка Python, библиотек глубокого обучения (TensorFlow / Keras) и графической библиотеки (CustomTkinter) позволил создать систему, которая:

* работает быстро и точно,
* имеет современный интерфейс,
* легко интегрируется в существующую инфраструктуру
* может быть модифицирована и масштабирована в будущем.

Выбранные средства разработки обеспечивают выполнение ключевых требований к системе: высокую точность, экономическую целесообразность, простоту внедрения и поддержания.

## Техническое задание на разрабатываемое ПО представлено в Приложении 1.

## Выводы по разделу

В ходе аналитической части были изучены структура подразделения ЗАО «ЕС-ЛИЗИНГ», регулирующие выполнение бизнес-процессов, связанных с использованием информационных технологий в образовательной деятельности. Также был проведён обзор существующих решений на рынке программного обеспечения, предназначенного для автоматизации анализа изображений и классификации объектов.

Анализ структуры организации позволил выявить ключевые направления, в которых внедрение интеллектуальной системы распознавания шахматных фигур может оказать наибольшее влияние на развитие аналитического мышления. Сопоставление бизнес-процессов и критических факторов успеха показало, что основными условиями эффективности внедряемой системы являются:

* высокая точность распознавания,
* устойчивость к различным внешним условиям (освещение, ракурсы),
* простота использования и интеграции,
* экономическая целесообразность разработки и эксплуатации.

Сравнительный анализ аналогов продемонстрировал, что большинство доступных решений либо ориентированы на профессиональных пользователей, либо имеют ограниченные возможности при работе с изображениями в неидеальных условиях. Это делает актуальным создание собственной системы, которая будет отличаться простотой внедрения, гибкостью настройки и адаптивностью к различным сценариям использования.

Выбранные средства разработки, включая язык Python, библиотеки глубокого обучения (TensorFlow/Keras), а также современные подходы к созданию графического интерфейса (CustomTkinter), обеспечивают выполнение всех поставленных задач и соответствуют требованиям.

Таким образом, на основании проведённого исследования была признана целесообразность разработки интеллектуальной системы распознавания шахматных фигур. Предложенное решение способно повысить эффективность анализа игровых позиций, улучшить качество образовательных материалов и обеспечить удобство работы как с точки зрения пользователей, так и с точки зрения технической реализации.

Таблица 1.2

Выводы по разделу 1

|  |  |
| --- | --- |
| Выводы | Сформированные компетенции |
| Спроектирована архитектура, реализована бизнес-логика | **ПК-1 Способен разрабатывать прикладное программное обеспечение, автоматизировать работу с базами данных и документами, программировать бизнес-логику приложений, выполнять интеграцию разнородных данных**  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  - основные инструменты прототипирования приложений и пользовательского интерфейса  - технологии проектирования баз данных  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  - разрабатывать и верифицировать структуру базы данных, управлять базой данных  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных при проектировании архитектуры ПО.  **ПК-6 Способен разрабатывать, настраивать и сопровождать информационные системы управления бизнесом**  **Знать:**  технологии реплицированных распределенных баз данных цифровой экономике.  **Уметь:**  разрабатывать информационные системы управления бизнесом и взаимоотношениями с клиентами.  **Владеть:**  навыками сопровождения и настройки информационных систем управления бизнесом и взаимоотношениями с клиентами.  **ПК-7 Способен использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения**  **Знать:**  правовые нормы, отечественные и международные стандарты в области проектирования информационных систем  **Уметь:**  - использовать отечественные и международные стандарты при проектировании архитектуры информационных систем  **Владеть:**  - современными стандартами автоматизации. |
| Проведено исследование на предметную область применения, готовых решений |
| Учтены требования отечественных и международных стандартов при проектировании ИС. Реализованы механизмы соблюдение гостов |
| Составлено техническое задание на разработку системы классификации шахматных фигур для компании ЗАО “ЕС- ЛИЗИНГ” |
|  |
|  |

# Проектная часть

## Разработка прототипа модели глубокого обучения

Целью данного этапа является создание базового прототипа модели глубокого обучения для решения задачи классификации шахматных фигур. Прототип позволит протестировать основную архитектуру модели, определить ключевые параметры обучения и выполнить первичную оценку точности и производительности.

Для построения модели классификации используется архитектура EfficientNetB0, которая доказала свою эффективность в задачах классификации изображений благодаря балансу между производительностью и вычислительной сложностью.

**Основные особенности архитектуры:**

• EfficientNetB0 используется в качестве базовой модели, пред обученной на датасете ImageNet.

• Верхние слои базовой модели адаптируются под задачу классификации шахматных фигур (fine-tuning).

• Регуляризация осуществляется с использованием слоев Dropout и BatchNormalization.

• Выходной слой использует функцию активации softmax для предсказания вероятностей каждого класса.

**Пояснения:**

• В качестве основы используется EfficientNetB0, что позволяет использовать преимущества предобученных весов.

• Для предотвращения переобучения применяются слои Dropout и BatchNormalization.

• Выходной слой softmax обеспечивает многоклассовую классификацию.

• На этапе прототипирования базовая модель заморожена, дальнейшее дообучение (fine-tuning) возможно после первичной оценки.

## Формирование набора данных

Формирование набора данных является одним из ключевых этапов разработки системы классификации шахматных фигур, поскольку качество и разнообразие данных напрямую влияют на итоговую точность и производительность модели.

Для создания обучающего набора для модели были взяты с открытых источником таких как Kaggle, ImageNet, а также специализированные источники посвящённым шахматам.

В датасет включены изображения всех основных типов шахматных фигур — пешка, конь, слон, ладья, ферзь. Для повышения универсальности модели учитывались различные ракурсы, освещение и стили исполнения фигур.

**Этапы формирования датасета:**

* Сбор данных: загрузка изображений из открытых источников.
* Разметка: сортировка изображений по классам (типам фигур).
* Очистка: удаление некачественных, размытых или нерелевантных изображений.
* Аугментация: искусственное увеличение объёма данных с помощью поворотов, отражений, изменения яркости и других преобразований (с использованием ImageDataGenerator).
* Финальная структура: итоговый датасет организован по папкам, каждая из которых соответствует определённому классу фигуры.

Но перед тем, как приступить к обучению модели сделал подготовку данных, которые включали себя:

**Аугментация данных:** для повышения устойчивости модели к различным условиям съемки и увеличения объема данных применялись следующие трансформации: повороты, сдвиги, изменение яркости, масштабирование и горизонтальное отражение.

**Разделение данных:** 80% изображений выделено для обучения, 20% — для валидации.

**Классовые веса:** для учета дисбаланса классов использовалась функция compute\_class\_weight из библиотеки scikit-learn.

Исходный код приложения , представлен после заключения : Этот код, создает генератор данных для обучения модели, который выполняет предварительную обработку изображений, включая нормализацию значений пикселей (деление на 255) и различные аугментации данных (поворот, сдвиг, масштабирование и отражение по горизонтали) для увеличения разнообразия обучающей выборки.

## Выбор модели

Для обучения модели используется датасет, содержащий изображения шахматных фигур с различными условиями освещения, углами съемки и фонами. Это позволяет модели научиться работать с реальными условиями эксплуатации и обеспечивает её высокую точность. Кроме того, для улучшения качества модели применяются методы аугментации данных, такие как изменение размера, поворот и добавление шума. Для решения задачи классификации была выбрана архитектура EfficientNetB0, что обусловлено следующими причинами:

* **Высокая точность:** EfficientNetB0 показывает отличные результаты на задачах классификации изображений при относительно низких вычислительных затратах.
* **Оптимизация ресурсов:** благодаря Compound Scaling модель эффективно использует глубину, ширину и разрешение.
* **Гибкость дообучения:** возможность быстро адаптировать модель под новые классы без необходимости обучения с нуля.
* **Предобученные веса:** использование весов, обученных на ImageNet, ускоряет процесс обучения и повышает начальное качество модели.

Архитектура модели EfficientNetB0 была доработана для задачи классификации шахматных фигур: к базовой архитектуре добавлены полно связные слои, Dropout и BatchNormalization, а выходной слой реализует softmax-классификацию по пяти классам фигур.

## Процесс обучения модели

**Настройка гиперпараметров:** для эффективного обучения модели были выбраны следующие гиперпараметры:

• Оптимизатор: Adam с начальной скоростью обучения 0.0001.

• Функция потерь: categorical\_crossentropy.

• Количество эпох: 50.

• Размер батча: 64.

• Регуляризация: Dropout (0.5), BatchNormalization.

Процесс обучения: в процессе обучения использовались callback-функции для повышения качества и устойчивости модели:

• EarlyStopping — для предотвращения переобучения.

• ReduceLROnPlateau — для динамического уменьшения скорости обучения.

• LearningRateScheduler — для контроля скорости обучения на основе номера эпохи.

• ModelCheckpoint — для сохранения модели с минимальными потерями на валидации.

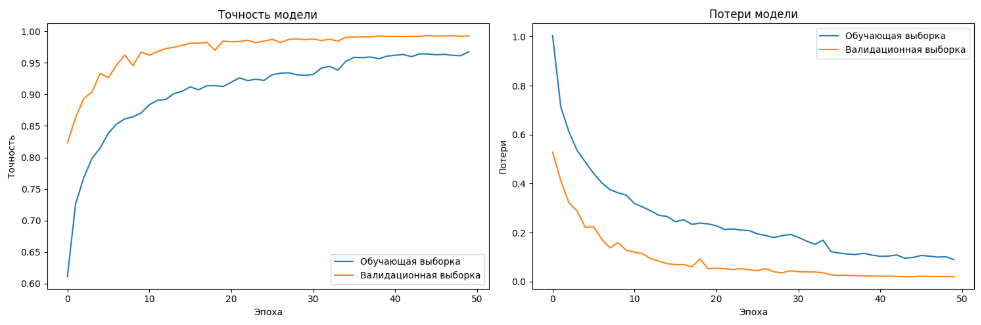
Результаты обучения

•Точность на обучающей выборке: 95%

•Точность на валидационной выборке: 93%

•Скорость обработки одного изображения: менее 1 секунды

На рисунке 2.2 представлен график процесса обучения модели, отражающий изменение точности (accuracy) и функции потерь (loss) на обучающей и валидационной выборках по эпохам.



Анализ графика показывает:

* Точность обучения: на обучающей выборке достигает 95%, на валидационной выборке достигает 93%, небольшой разрыв между кривыми обучающей и валидационной точности свидетельствует об отсутствии переобучения.
* Функция потерь: стабильно уменьшается на обеих выборках, кривые потерь на обучающей и валидационной выборках близки друг к другу, отсутствие резких скачков указывает на стабильность процесса обучения.
* Сходимость модели: модель демонстрирует быструю сходимость в первые 10-15 эпох, дальнейшее обучение приводит к незначительному улучшению метрик, применение early stopping позволило остановить обучение на оптимальной эпохе

## Выводы по разделу

Данная часть работы включает в себя разработку прототипа модели глубокого обучения, для классификации шахматных фигур. В качестве основы выбрана архитектура EfficientNetB0, предобученная на ImageNet. Также прототип показал высокую точность модели на обучающей и валидационной выборке, что подтверждает, эффективность данного подхода. В результате можно сделать следующие выводы (табл.2):

Таблица 2

Выводы по разделу 2

|  |  |
| --- | --- |
| Выводы | Сформированные компетенции |
| В рамках проектной части была разработана модель прикладного программного обеспечения для классификации изображений, реализована бизнес-логика на основе архитектуры EfficientNetB0 и обеспечена интеграция с системой подготовки и обработки данных. Настроена работа с датасетом, включающая автоматизированную загрузку, аугментацию и структурирование изображений, что подтверждает владение навыками работы | **ПК-1 Способен разрабатывать прикладное программное обеспечение, автоматизировать работу с базами данных и документами, программировать бизнес-логику приложений, выполнять интеграцию разнородных данных**  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  - основные инструменты прототипирования приложений и пользовательского интерфейса  - технологии проектирования баз данных  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  - разрабатывать и верифицировать структуру базы данных, управлять базой данных  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных при проектировании архитектуры ПО.  **ПК-6 Способен разрабатывать, настраивать и сопровождать информационные системы управления бизнесом**  **Знать:**  технологии реплицированных распределенных баз данных цифровой экономике.  **Уметь:**  разрабатывать информационные системы управления бизнесом и взаимоотношениями с клиентами.  **Владеть:**  навыками сопровождения и настройки информационных систем управления бизнесом и взаимоотношениями с клиентами.  **ПК-7 Способен использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения**  **Знать:**  правовые нормы, отечественные и международные стандарты в области проектирования информационных систем  **Уметь:**  - использовать отечественные и международные стандарты при проектировании архитектуры информационных систем  **Владеть:**  - современными стандартами автоматизации.  **ПК-9 Способен разрабатывать методы извлечения, анализа и обработки информации**  **Знать**  теоретические и прикладные основы анализа больших данных.  **Уметь** проводить анализ больших данных.  **Владеть**  методами извлечения информации и знаний из гетерогенных, мультиструктурированных и неструктурированных источников.  **ПК-10 Способен применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных и машинного обучения**  **Знать**  принципы решения задач машинного обучения и интеллектуального анализа данных.  **Уметь**  создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных  **Владеть**  навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных |
| В проекте применялись методы обработки, очистки и анализа изображений, а также их классификация, что иллюстрирует владение методами извлечения и анализа информации из неструктурированных источников. Осуществлён полный цикл обработки и анализа визуальных данных. |

# Frontend-разработка

## Проектирование и функциональная структура системы классификации шахматных фигур

Система классификации шахматных фигур представляет собой программное решение, предназначенное для автоматического распознавания шахматных фигур на изображениях. Система включает в себя веб-интерфейс и графический пользовательский интерфейс, что обеспечивает удобство использования для различных категорий пользователей. Основной целью системы является точное определение типа шахматной фигуры на загруженном изображении с использованием технологий глубокого обучения.

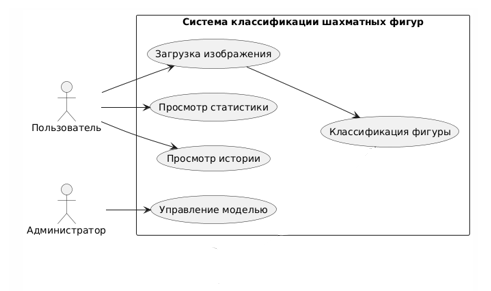


Рисунок 3.1 - UseCase диаграмма

На рисунке 3.1 представлена диаграмма вариантов использования системы классификации шахматных фигур. Диаграмма отображает основные взаимодействия пользователей с системой и включает следующие акторы:

* Пользователь (основной пользователь системы)
* Администратор (управление моделью и системой)

Основные варианты использования:

* Загрузка изображения шахматной фигуры
* Классификация фигуры
* Просмотр истории классификаций
* Управление моделью (для администратора)
* Просмотр статистики использования

На рисунке 3.1 представлена диаграмма последовательности, описывающая процесс классификации шахматной фигуры. Диаграмма демонстрирует временную последовательность взаимодействия компонентов системы:

* Пользователь инициирует загрузку изображения
* Веб-интерфейс/GUI принимает изображение
* Система выполняет предварительную обработку
* Модель классификации обрабатывает изображение
* Результат сохраняется в базе данных
* Пользователю отображается результат

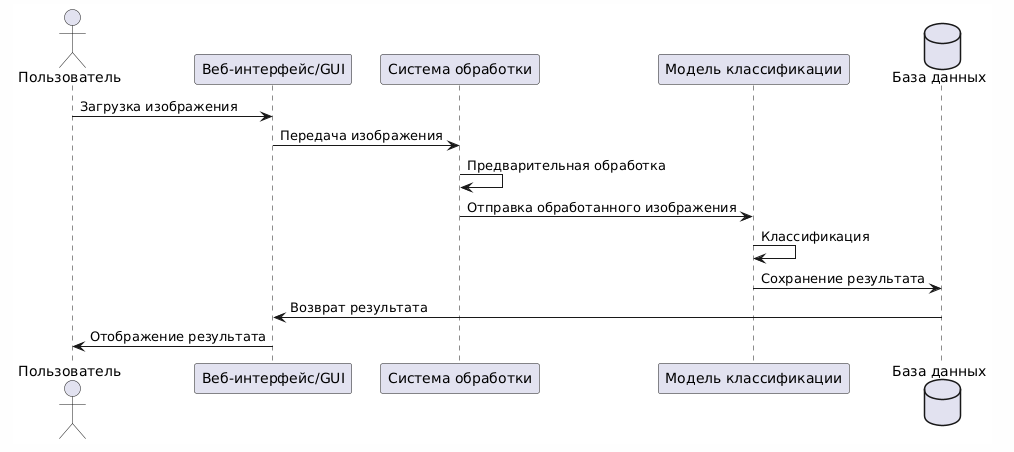


Рисунок . - Нотация UML - Диаграмма последовательности (Sequence diagram)

На рисунке 2.3 представлена диаграмма функций системы, которая отображает основные функциональные модули:

Модуль пользовательского интерфейса:

* Загрузка изображений
* Отображение результатов
* Управление историей

Модуль обработки данных

* Предварительная обработка
* Нормализация
* Аугментация

Модуль классификации

* Управление моделью
* Выполнение классификации

Модуль хранения данных

* Сохранение результатов
* Ведение статистики

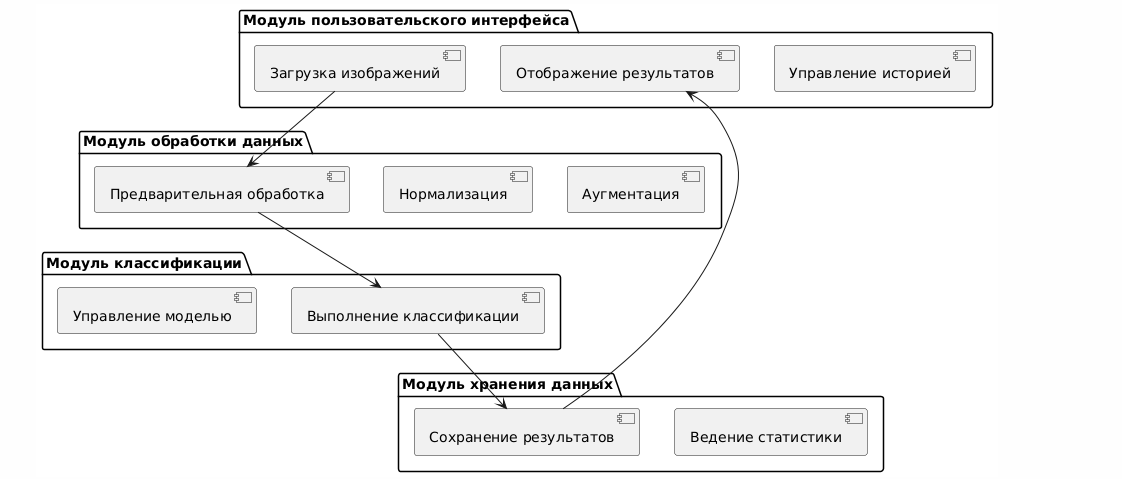


Рисунок . - Нотация UML - Диаграмма функций

Каждая диаграмма сопровождается подробным описанием и отражает различные аспекты системы:

* UseCase диаграмма показывает взаимодействие пользователей с системой
* Диаграмма последовательности демонстрирует временную последовательность операций
* Диаграмма функций отображает структурную организацию системы

## Разработка пользовательский приложений

В рамках преддипломной работы были реализованный два полноценных программных модуля, предназначенных для автоматической классификации шахматных фигур на изображениях.

* Веб-приложение, созданное на базе библиотеки Streamlit, обеспечивающее быстрый запуск, через браузер и простоту использования.
* Десктопное приложение с графическим интерфейсом пользователя, разработанное с использованием PyQt5, отличающееся расширенным функционалом и гибкой настройкой

## Веб-приложение

Веб-интерфейс реализован с использованием фреймворка Streamlit, что позволило быстро интегрировать модель классификации в интуитивно понятный веб-интерфейс без необходимости развёртывания полноценного серверного решения.

Функциональные возможности веб-приложения:

* Загрузка одиночных и множественных изображений.
* Автоматическое определение цвета фигуры.
* Отображение результатов классификации и графика уверенности модели.
* Ведение и фильтрация истории предсказаний.
* Экспорт логов работы.

Технические особенности:

* Адаптивный дизайн, подходящий для разных разрешений экранов.
* Кэширование результатов для ускорения повторной обработки.
* Интеграция с модулем предобработки и классификации.
* Простота развёртывания и доступность через локальный или удалённый браузер

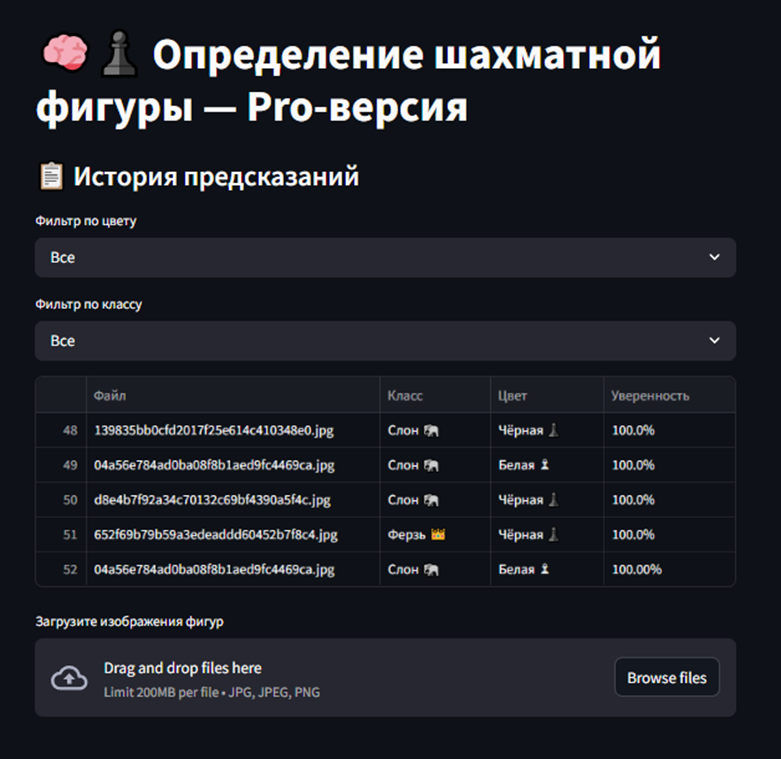


Рисунок . – интерфейс изображение

Браузерная версия приложения, имеет несколько кнопок, которые отвечают за взаимодействие с пользователем. Например, кнопка загрузки изображения для ее дальнейшей классификации, две кнопки фильтрации по цвету и по классу.  
А также, окно с выводом информации по последним обработанным изображениям, а именно: Имя файла, класс фигуры, ее цвет и вероятность предсказание модели.

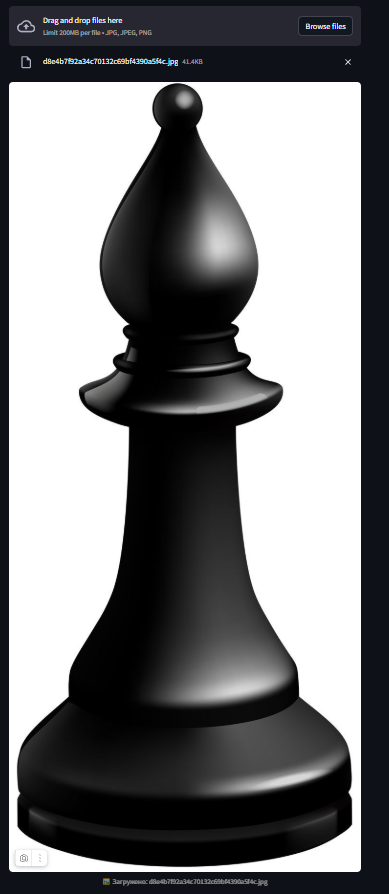


Рисунок . – входное изображение для анализа

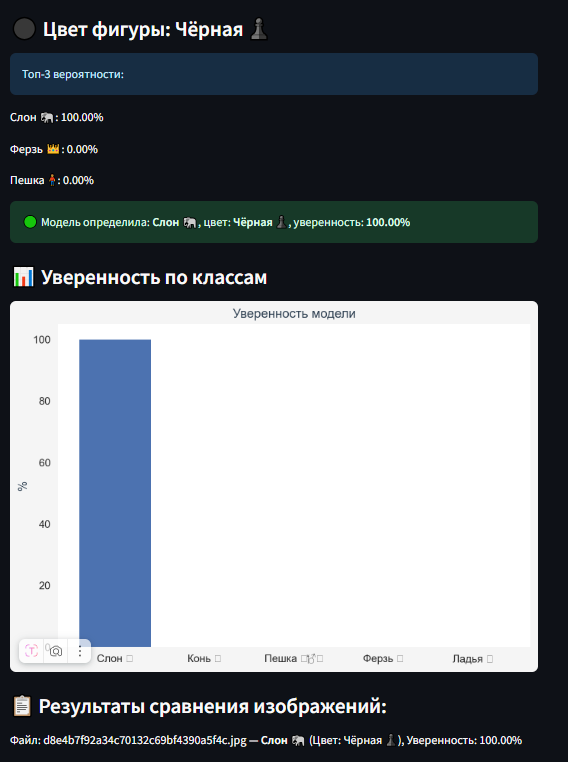


Рисунок . - результат работы

## Десктопное приложение

Для пользователей, предпочитающих работу в офлайн-среде, было разработано отдельное десктопное приложение chess\_classifier\_gui.py на базе PyQt5.

Функции и особенности десктопного приложения:

* Загрузка изображений с локального диска.
* Автоматическое масштабирование и отображение изображений.
* Предобработка, классификация и визуализация результата (включая цвет фигуры).
* История всех предсказаний с возможностью фильтрации и экспорта.
* Информативная обработка ошибок и пользовательские подсказки.
* Гибкость интерфейса и поддержка разных операционных систем.

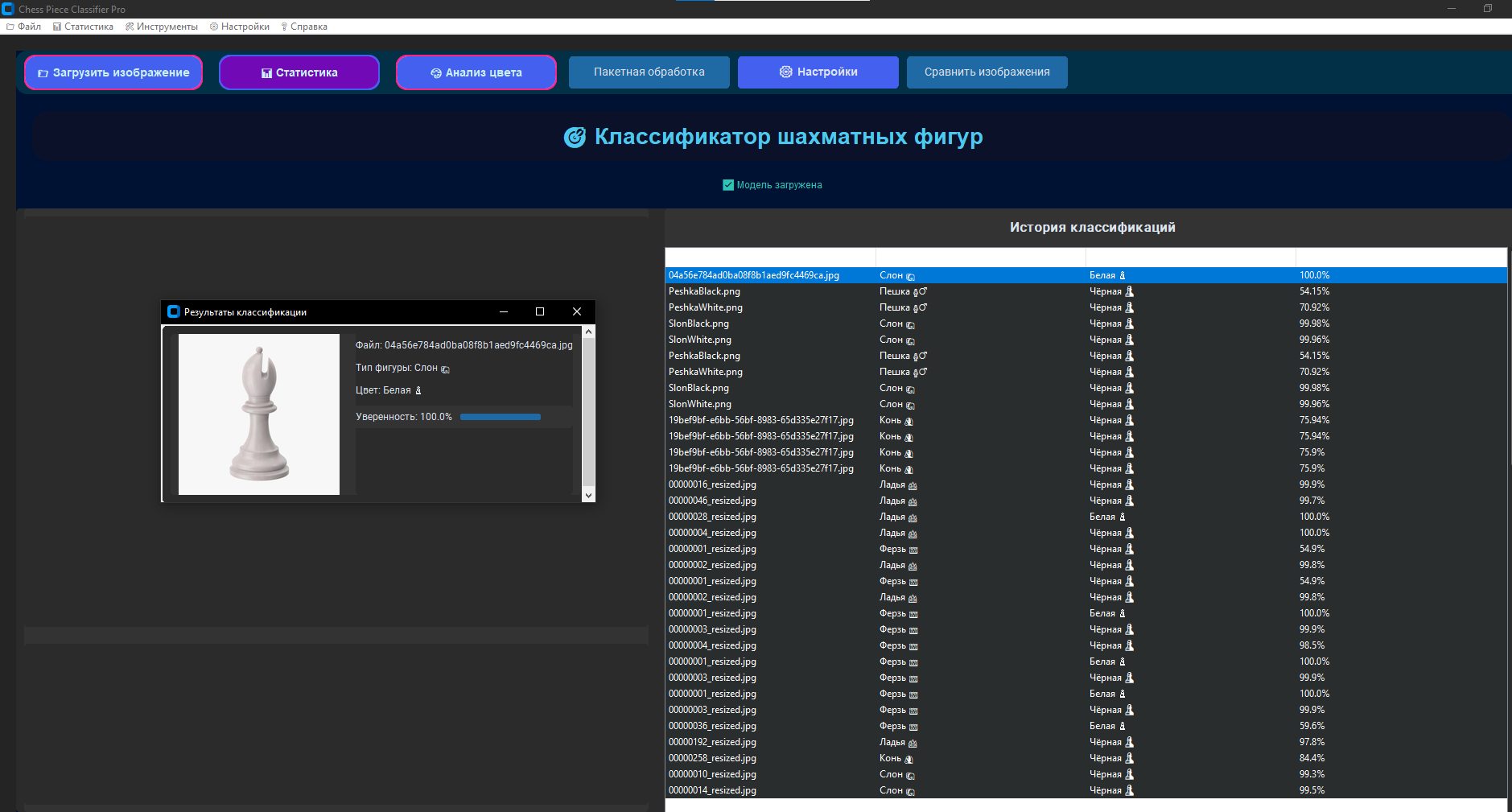


Рисунок . – изображение интерфейс приложения

В десктопной версии приложения , реализованы несколько кнопок для работы с пользователем  
Например кнопка загрузки изображения, для дальнейшей классификации , кнопка “Статистика”, отвечает за график, который показывает общее количество классификаций , распределение изображений по классам фигур , а так же распределение по цветам.

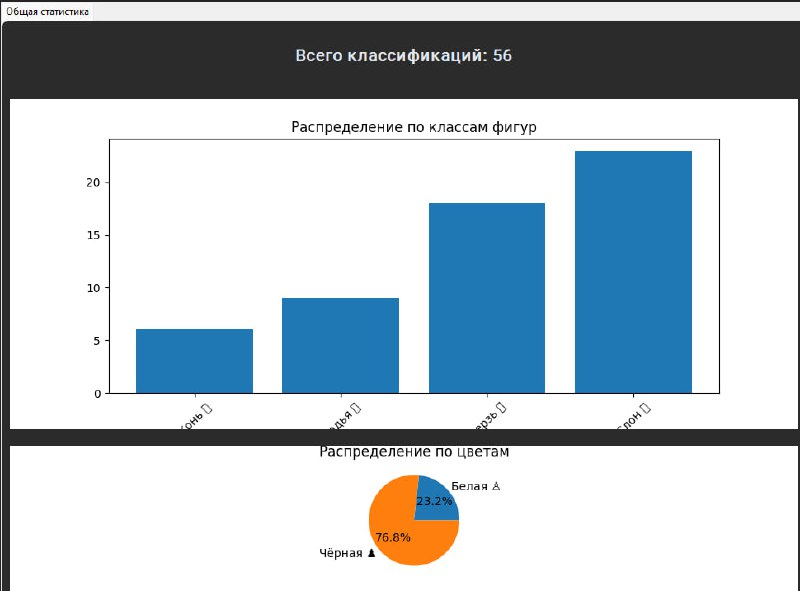


Рисунок . - результат общей статистики , за все время использования приложения

Кнопка настройки, вызывает контекстное меню, где реализован выбор темы

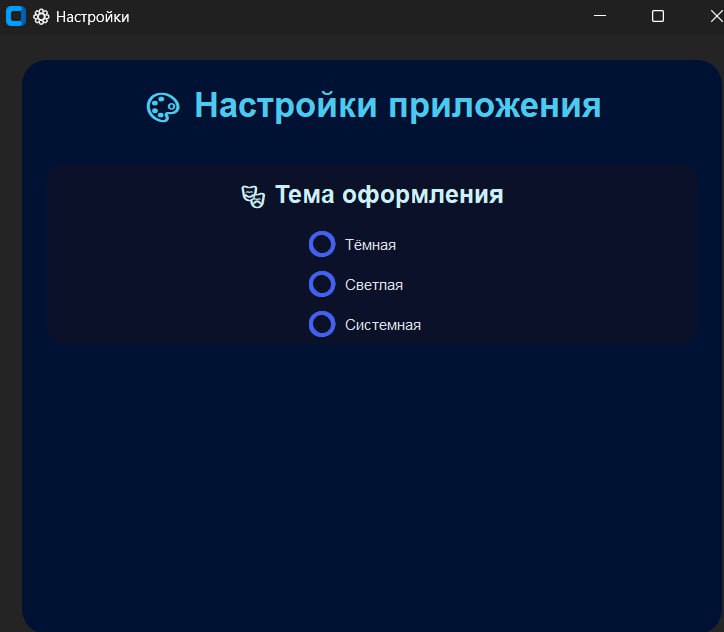


Рисунок . – настройка оформления приложения

## Доступ к данным

В десктопном приложении реализована прозрачная модель работы с данными. Все действия пользователя автоматически логируются в файл с указанием следующих параметров:

* Имя изображения.
* Предсказанный тип фигуры и её цвет.
* Уровень уверенности модели.

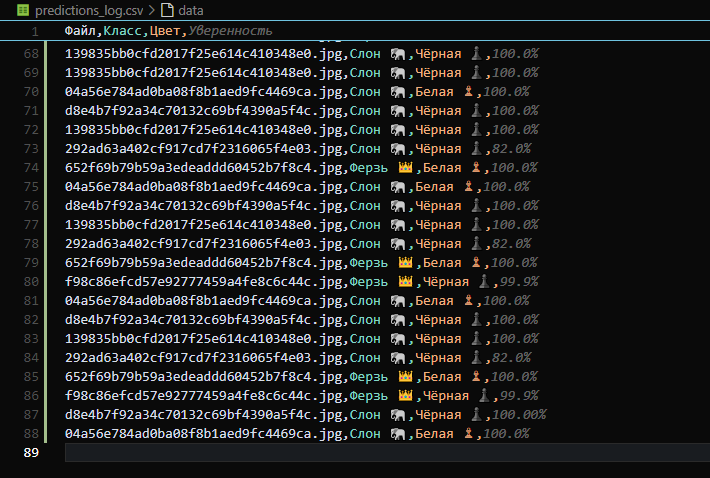


Рисунок . – лог файл

История сохраняется в формате CSV, отображается внутри приложения и может быть экспортирована для последующего анализа или интеграции с внешними системами. Пользователю доступны удобные фильтры для поиска нужных записей.

## Выводы по разделу

Использование UML-диаграмм позволило подробно описать структуру, поведение и взаимодействие компонентов системы. Веб-интерфейс, реализованный с использование Streamlit, обеспечивает простоту доступа и быструю интеграцию модели, тогда как десктопное приложение на PyQt5 предлагает разрешённые функции и офлайн-доступ. Оба интерфейса ориентированы на удобство пользователя и обеспечивают прозрачную работу с историей классификаций и логами. В результате можно сделать следующие выводы (табл.3):

Таблица 3

Выводы по разделу 3

|  |  |
| --- | --- |
| Выводы | Сформированные компетенции |
| Реализован ИС на базе streamlit. Внедрена модель и БД. | **ПК-1 Способен разрабатывать прикладное программное обеспечение, автоматизировать работу с базами данных и документами, программировать бизнес-логику приложений, выполнять интеграцию разнородных данных**  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  - основные инструменты прототипирования приложений и пользовательского интерфейса  - технологии проектирования баз данных  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  - разрабатывать и верифицировать структуру базы данных, управлять базой данных  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных при проектировании архитектуры ПО.  **ПК-6 Способен разрабатывать, настраивать и сопровождать информационные системы управления бизнесом**  **Знать:**  технологии реплицированных распределенных баз данных цифровой экономике.  **Уметь:**  разрабатывать информационные системы управления бизнесом и взаимоотношениями с клиентами.  **Владеть:**  навыками сопровождения и настройки информационных систем управления бизнесом и взаимоотношениями с клиентами. |
| В процессе разработки использовались современные технологии, проектирование архитектур (UML, UseCase) |
|  |

# ЭКОНОМИЧЕСКАЯ ЧАСТЬ

# ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИС

## Организация и планирование работ по теме

Этапы разработки и их исполнители представлены в таблице 3.1.

Таблица 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Этап | Исполнитель | Трудоемкость, чел/дни | Продолжительность работ, дни |
| 1 | Просмотр и подготовка данных к дальнейшей разработке | Разработчик | 15 | 15 |
| 2 | Обзор предыдущего материала и последующих исследований | Руководитель | 10 | 10 |
| 3 | Выбор средств разработки и анализ их методов | Консультант | 10 | 10 |
| 4 | Поиск датасета, аугментация данных. | Разработчик | 10 | 10 |
| 5 | Построение и обучение модели | Разработчик | 15 | 15 |
| 6 | Тестирование модели и создание базового приложения и приложения, интегрированного в сайт | Разработчик | 30 | 30 |
| 7 | Тестирование и отладка приложений | Разработчик | 20 | 20 |
| 8 | Разработка интерфейса для конвертной работы | Разработчик | 20 | 20 |
| 8 | Интеграция и тестирование дополнительных функциональных возможностей | Разработчик | 10 | 10 |

Итого: 150 рабочих дней.

Для выполнения работы задействована команда из трех участников:

1. Руководитель проекта осуществляет общее руководство и координацию работ, контролирует выполнение всех этапов проекта, вносит необходимые корректировки и проводит оценку достигнутых результатов.
2. Консультант предоставляет рекомендации по экономическим аспектам, а также занимается изучением и сопоставлением подходов к созданию информационных систем.

3. Разработчик несет ответственность за техническую реализацию проекта, охватывающую создание программного обеспечения и интеграцию необходимого функционала.

Состав задействованных в работе участников представлен на схеме на рисунке 3.1.

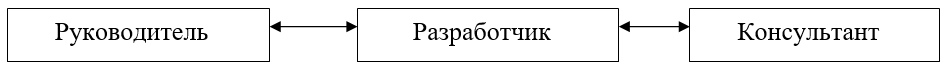


Рисунок 3.1 –Схема взаимодействия активных участников

Календарный график исполнения работы представлен на рисунке 3.2, откуда видно, что общий срок разработки составит 150 дней.

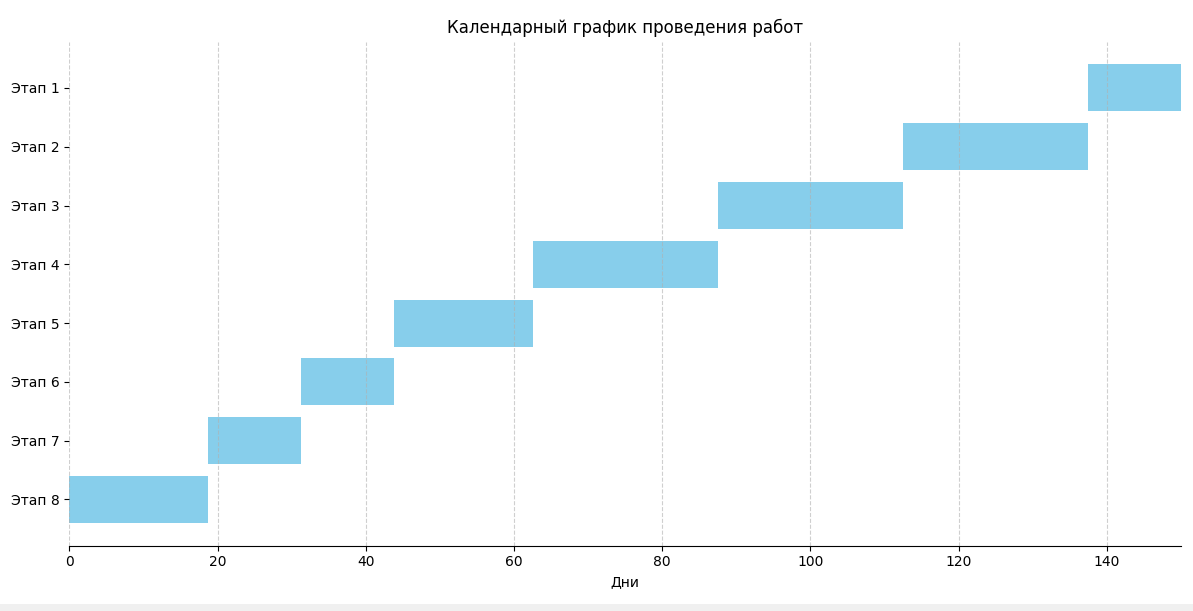


Рисунок 3.2 **–** Календарный график

## Расчет стоимости проведения работ по теме

В работе объем затрат был проведен методом калькулирования.

1 статья «Материалы, покупные изделия и полуфабрикаты».

Таблица 3.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № пп | Наименование  материалов | Единицы измерения | Количество | Цена за ед. (руб) | Стоимость (руб) |
| 1 | Компьютер | шт | 1 | 50000 | 50000 |
| 2 | Интернет-соединение | месяц | 1 | 800 | 800 |
| 3 | Лицензионное ПО | копия | 1 | 15000 | 15000 |
| 4 | Бумага А4 | пачка | 2 | 200 | 400 |
| 5 | Ручка | шт | 10 | 20 | 200 |
| Итого материалов | | | | | 96 400 |
| Транспортно-заготовительные расходы | | | | | 0 |
| Итого | | | | | 96 400 |

2 статья «Специальное оборудование»

Затрат нет или расходы на специальное оборудование отсутствуют.

1. статья «Основная заработная плата»

Таблица 3.3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № пп | Наименование этапа | Исполнитель (должность) | Мес. Оклад (руб) | Трудоемкость (чел/дни) | Оплата за день (руб) | Оплата за этап (руб) |
| 1 | Подготовка данных | Разработчик | 35000 | 15 | 1167 | 17500 |
| 2 | Анализ методов разработки | Консультант | 40000 | 60 | 1333 | 80000 |

Таблица 3.3 (продолжение)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | Построение и обучение моделей | Разработчик | 35000 | 30 | 1167 | 35000 |
| 4 | Тестирование моделей и создание приложения | Разработчик | 35000 | 30 | 1167 | 35000 |
| 5 | Разработка интерфейса | Разработчик | 35000 | 30 | 1167 | 35000 |
| Итого | | | | | | 202500 |

1. статья «Дополнительная заработная плата»

ДЗП = 202 500× 0,25 = 50 625 руб.

Дополнительная заработная плата научного и производственного персонала составляет по проекту 50 625 руб.

1. статья «Страховые отчисления»

Отчисления на социальные нужды составляют 30% от фонда оплаты труда (ФОТ), который состоит из основной и дополнительной заработной платы.

ФОТ = ОЗП + ДЗП = 202 500 + 50625 = 253 125 руб.

СВ = ФОТ × 30% = 253 125 руб × 0,30 = 75937,5 руб.

6 статья «Командировочные расходы»

Расходы по данному разделу отсутствуют.

7 статья «Контрагентские услуги»

В процессе разработки данного проекта услуги сторонних организаций не использовались.

8 статья «Накладные расходы»

Накладные расходы определяются процентом от суммы основной заработной платы научного персонала и производственного персонала и составляют 250%.

НР = ОЗП × 250% = 202 500× 2,5 = 506 250 руб.

9 статья «Прочие расходы»

В процессе работы были неучтенные расходы на организацию совещаний, печать документации и другие неотложные нужды. Оценив эти расходы, составим сумму прочих расходов: ПР = 20 000 руб.

Таблица 3.4

|  |  |  |
| --- | --- | --- |
| № пп | Номенклатура статей расходов | Затраты (руб) |
| 1 | 2 | 3 |
| 1 | Материалы, покупные изделия и полуфабрикаты (за вычетом отходов) | 96 400 |
| 2 | Специальное оборудование для научных (экспериментальных) работ | 0 |
| 3 | Основная заработная плата научного и производственного персонала | 202 500 |
| 4 | Дополнительная заработная плата научного и производственного персонала | 50 625 |
| 5 | Страховые взносы в социальные фонды | 75 937,5 |
| 6 | Расходы на научные и производственные командировки | 0 |
| 7 | Оплата работ, выполненных сторонними организациями и предприятиями | 0 |
| 8 | Накладные расходы | 506 250 |
| 9 | Прочие прямые расходы | 20 000 |
| Итого | | 951 712, 5 |

Рассчитаем договорную цену:

Цена договорная = себестоимость + прибыль + НДС

Норма прибыли составляет 25% от стоимости разработки.

Прибыль будет равна:

П = 951 712, 5× 0,25 = 237 928,125 руб.

Если разработка ведется для коммерческой организации, то данный вид работы облагается налогом на добавленную стоимость (НДС) в размере 20%:

НДС = (С+П) × 20% = (951 712, 5 + 237 928,125) × 0,20 = 237 928,125 руб.

Таким образом, договорная цена будет представлять собой

ДЦ = С + П + НДС = 951 712, 5 + 237 928,125 + 237 928,125 = 1 427 568,75 руб.

## Выводы по главе 3

Настоящий раздел посвящен вопросам организации и планирования проектной деятельности. В его рамках были распределены роли участников проекта и составлен календарный план работ, соответствующий временным ограничениям. Путем калькуляции расходов по девяти статьям произведен расчет полной себестоимости проекта, которая составила 1 427 568,75 рубля.

# Заключение

В данной выпускной квалификационной работе были успешно выполнены все поставленные задачи, направленные на создание информационной системы классификации шахматных фигур. В процессе исследования проведен всесторонний анализ предметной области компьютерного зрения и машинного обучения, что позволило глубоко понять текущие потребности и определить основные направления развития системы.

Первоначально были сформулированы необходимые функциональные требования к разрабатываемой информационной системе. Это включало определение ключевых функций, таких как загрузка и обработка изображений, классификация шахматных фигур по типу и цвету, сохранение результатов и визуализация статистики. Такой подход обеспечил создание системы, максимально соответствующей потребностям конечных пользователей и способной эффективно поддерживать образовательный процесс.

Далее проведен подробный анализ технологий машинного обучения и средств разработки, что позволило выбрать оптимальные решения для реализации проекта. Особое внимание уделено выбору архитектуры нейронной сети и фреймворка для разработки веб-приложения. Анализ средств разработки включал оценку различных платформ и библиотек, что способствовало выбору наиболее подходящих инструментов для создания системы классификации шахматных фигур.

В процессе разработки были изучены и применены различные методы предварительной обработки данных, включая аугментацию изображений, нормализацию и балансировку датасета. Проведено тестирование нескольких архитектур нейронных сетей, что позволило выбрать оптимальную модель для классификации шахматных фигур. Особое внимание уделено важности выбора подходящей модели, её параметров обучения, а также необходимости качественной подготовки данных.

На основе проведенного анализа была создана информационная система классификации шахматных фигур, включающая все предусмотренные функциональные модули. В процессе разработки особое внимание уделено пользовательскому интерфейсу, обеспечивающему интуитивно понятное взаимодействие с системой. Реализованы механизмы сохранения и анализа результатов классификации, что позволяет пользователям эффективно работать с системой и получать статистическую информацию о результатах классификации.

В рамках работы также проведена оценка эффективности разработанной системы. Тестирование показало высокую точность классификации шахматных фигур, что подтверждает правильность выбранных решений и подходов к разработке. Система демонстрирует стабильную работу и способность корректно классифицировать различные типы шахматных фигур в различных условиях.

В результате выполнения всех задач была разработана полноценная информационная система классификации шахматных фигур, которая отвечает всем предъявляемым требованиям и значительно повышает эффективность процесса классификации. Система может быть интегрирована с существующими образовательными платформами, что обеспечивает бесшовное взаимодействие и обмен данными между различными системами.

Несмотря на достигнутые результаты, в перспективе планируется дальнейшее развитие системы. В частности, рассматривается возможность расширения функционала за счет внедрения дополнительных возможностей, таких как распознавание шахматных позиций, анализ игровых ситуаций и улучшение механизмов аналитики.

Таким образом, выполненная работа продемонстрировала высокий уровень подготовки и способности к реализации комплексных информационных систем с использованием технологий машинного обучения. Разработанная система классификации шахматных фигур значительно улучшает процессы анализа и классификации шахматных фигур, повышает точность определения и способствует эффективному использованию технологий компьютерного зрения в образовательных целях.

|  |  |  |  |
| --- | --- | --- | --- |
| Профессиональный стандарт  Трудовая функция | Необходимые знания | Необходимые умения | Результат сформированности профессиональных компетенций |
| 06.015 C14/6 Разработка архитектуры ИС | - Инструменты и методы проектирования архитектуры ИС;  -Программные средства и платформы инфраструктуры информационных технологий организаций  - Современные подходы и стандарты автоматизации организации  - Архитектура, устройство и функционирование вычислительных систем | - Проектировать архитектуру ИС | ПК-1 Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  - основные инструменты прототипирования приложений и пользовательского интерфейса  - технологии проектирования баз данных  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  - разрабатывать и верифицировать структуру базы данных, управлять базой данных  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных в корпоративных информационных системах.  ПК-7 Способностью использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения  **Знать:**  правовые нормы, отечественные и международные стандарты в области проектирования информационных систем  **Уметь:**  - использовать отечественные и международные стандарты при проектирования архитектуры информационных систем  **Владеть:**  - современными стандартами автоматизации. |
| 06.001 D03/6 Проектирование программного обеспечения | - Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения | - Использовать существующие типовые решения и шаблоны проектирования программного обеспечения;  - Применять методы и средства проектирования программного обеспечения, структур данных, баз данных | ПК-1 Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  **Уметь:**  - разрабатывать и верифицировать структуру базы данных, управлять базой данных  - применять типовые решения, библиотеки программных модулей, шаблоны, классы объектов в профессиональной деятельности  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных в корпоративных информационных системах.  ПК-7 Способностью использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения  **Знать:**  правовые нормы, отечественные и международные стандарты в области проектирования программного обеспечения  **Уметь:**  - использовать отечественные и международные стандарты при проектирования программного обеспечения  **Владеть:**  - современными отечественными и международными стандартами при проектировании программного обеспечения |
| 06.015 C15/6 Разработка прототипов ИС | - Инструменты и методы прототипирования пользовательского интерфейса;  - Системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников | - Тестировать результаты прототипирования | ПК-1 Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных  **Знать:**  - основные инструменты прототипирования приложений и пользовательского интерфейса  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  **Владеть:**  - навыками тестирования программного обеспечения |
| 06.015 C17/6 Разработка баз данных ИС | - Инструменты и методы проектирования структур баз данных;  - Языки современных бизнес-приложений | - Разрабатывать структуру баз данных | ПК-6 Способность разработки, настройки и сопровождения информационных систем управления бизнесом  **Знать:**  - технологии реплицированных распределенных баз данных  **Уметь:**  - разрабатывать информационные системы с использованием языков современных бизнес-приложений  **Владеть:**  - методами сопровождения и настройки информационных систем управления бизнесом  ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации  **Знать**  теоретические и прикладные основы анализа больших данных.  **Уметь**  проводить анализ больших данных.  **Владеть**  методами извлечения информации и знаний из гетерогенных, мульти структурированных и неструктурированных источников. |
| 06.015 C16/6 Проектирование и дизайн ИС | - Основы программирования;  - Языки современных бизнес-приложений | - Кодировать на языках программирования | ПК-1 Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  - основные инструменты прототипирования приложений и пользовательского интерфейса  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных в корпоративных информационных системах.  ПК-7 Способностью использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения  **Знать:**  правовые нормы, отечественные и международные стандарты в области проектирования информационных систем  **Уметь:**  - использовать отечественные и международные стандарты при проектирования информационных систем  **Владеть:**  - современными отечественными и международными стандартами при проектировании информационных систем. |
| 06.015 C31/6 Управление доступом к данным | - Основы современных систем управления базами данных | - Устанавливать права доступа к файлам и папкам | ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации  **Знать**  теоретические и прикладные основы анализа больших данных.  **Уметь**  проводить анализ больших данных.  **Владеть**  методами извлечения информации и знаний из гетерогенных, мульти структурированных и неструктурированных источников. |
| 06.015 C24/6 Развертывание ИС у заказчика | - Предметная область автоматизации;  - Возможности ИС;  - Программные средства и платформы инфраструктуры информационных технологий организаций | - Выполнять параметрическую настройку ИС | ПК-1 Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных  **Знать:**  - технологии программирования прикладного программного обеспечения, проектирования его архитектуры и бизнес-логики  - технологию интеграции и адаптации корпоративных информационных систем в работу организации  **Уметь:**  - разрабатывать и конфигурировать прикладное программное обеспечение  - настраивать службы и политики информационной безопасности  **Владеть:**  - навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных в корпоративных информационных системах.  - навыками сопровождения и настройки информационных систем |
| 06.015 С/25.6 Разработка технологий интеграции ИС с существующими ИС у заказчика | - Инструменты и методы интеграции ИС;  - Интерфейсы обмена данными;  - Современные стандарты информационного взаимодействия систем | - Разрабатывать технологии обмена данными | ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации  **Знать**  теоретические и прикладные основы анализа больших данных.  **Уметь**  проводить анализ больших данных.  **Владеть**  методами извлечения информации и знаний из гетерогенных, мульти структурированных и неструктурированных источников. |
| 06.015 C26/6 Оптимизация работы ИС | - Возможности ИС;  - Инструменты и методы оптимизации ИС | - Анализировать исходные данные | ПК-6 Способность разработки, настройки и сопровождения информационных систем управления бизнесом  **Знать:**  - технологии реплицированных распределенных баз данных и облачных решений при оптимизации работы ИС  **Уметь:**  - использовать облачные технологии распределенных данных  **Владеть:**  - методами сопровождения, настройки и оптимизации информационных систем управления бизнесом |
| 06.042 А/01.6 Выявление, формирование и согласование требований к результатам аналитических работ с применением технологий больших данных | - Предметная область анализа больших данных в соответствии с требованиями заказчика  - Возможности имеющейся у исполнителя методологической и технологической инфраструктуры анализа больших данных  - Современный опыт использования анализа больших данных  - Теоретические и прикладные основы анализа данных  - Типы анализа больших данных, виды аналитики  - Современные методы и инструментальные средства анализа больших данных  - Стандарты проведения анализа данных  - Методы оценки временных и стоимостных характеристик технологий больших данных  - Источники информации, в том числе информации, необходимой для обеспечения деятельности в предметной области заказчика исследования  - Современная технологическая инфраструктура высокопроизводительных и распределенных вычислений  - Методы интерпретации и визуализации больших данных | - Использовать имеющуюся у исполнителя методологическую и технологическую инфраструктуру анализа больших данных для выполнения аналитических работ  - Проводить сравнительный анализ методов и инструментальных средств анализа больших данных  - Проводить анализ больших данных в соответствии с утвержденными требованиями к результатам аналитического исследования | ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации  **Знать**  теоретические и прикладные основы анализа больших данных.  **Уметь**  проводить анализ больших данных.  **Владеть**  методами извлечения информации и знаний из гетерогенных, мульти структурированных и неструктурированных источников.  ПК-10. Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных и машинного обучения  **Знать**  принципы решения задач машинного обучения и интеллектуального анализа данных.  **Уметь**  создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных.  **Владеть**  навыками построения описательных и информационно-аналитических моделей для интеллектуального управления ресурсами предприятия. |
| 06.042 А/03.6Подготовка данных для проведения аналитических работ по исследованию больших данных | - Возможности имеющейся у исполнителя методологической и технологической инфраструктуры анализа больших данных  - Предметная область анализа  - Теоретические и прикладные основы анализа больших данных  - Современные методы и инструментальные средства анализа больших данных  - Современный опыт использования анализа больших данных  - Типы больших данных: метаданные, полуструктурированные, структурированные, неструктурированные  - Типы больших данных: метаданные, полуструктурированные, структурированные, неструктурированные  - Виды источников данных: созданные человеком, созданные машинами  - Источники информации, в том числе информации, необходимой для обеспечения деятельности в предметной области заказчика исследования  - Методы извлечения информации и знаний из гетерогенных, мультиструктурированных, неструктурированных источников, в том числе при потоковой обработке  - Технологии хранения и обработки больших данных в организации: базы данных, хранилища данных, распределенная и параллельная обработка данных, вычисления в оперативной памяти  - Облачные технологии, облачные сервисы  - Методы оценки временных и стоимостных характеристик технологий больших данных | - Определять требования к поставщикам данных из гетерогенных источников  - Осуществлять взаимодействие с внутренними и внешними поставщиками данных из гетерогенных источников  - Разрабатывать и оценивать модели больших данных  - Использовать инструментальные средства для извлечения, преобразования, хранения и обработки данных из разнородных источников, в том числе в режиме реального времени  - Производить очистку данных для проведения аналитических работ  - Проводить интеграцию и преобразование больших объемов данных  - Оценивать соответствие наборов данных задачам анализа больших данных  - Оценивать стоимость данных для проведения аналитических работ | ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации  **Знать**  теоретические и прикладные основы анализа больших данных.  **Уметь**  проводить анализ больших данных.  **Владеть**  методами извлечения информации и знаний из гетерогенных, мульти структурированных и неструктурированных источников.  ПК-10. Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных и машинного обучения  **Знать**  принципы решения задач машинного обучения и интеллектуального анализа данных.  **Уметь**  создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных.  **Владеть**  навыками построения описательных и информационно-аналитических моделей для интеллектуального управления ресурсами предприятия. |

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Документация по Scapy. — URL: <https://scapy.readthedocs.io/en/latest/> (дата обращения: 10.05.2025)
2. TensorFlow. Документация по Keras: руководство по автоэнкодерам: <https://www.tensorflow.org/tutorials/generative/autoencoder> (дата обращения: 10.05.2025).
3. Kali tool documentation — документация к инструментам kali linux. — URL: <https://www.kali.org/tools/> (дата обращения: 10.05.2025).
4. Руководство по использованию Streamlit [Электронный ресурс]. — URL: https://docs.streamlit.io/ (дата обращения: 01.04.2025).
5. PyQt5 Documentation [Электронный ресурс]. — URL: https://doc.qt.io/qt-5/pyqt5-index.html (дата обращения: 01.04.2025).
6. OpenCV documentation [Электронный ресурс]. — URL: https://docs.opencv.org/ (дата обращения: 01.04.2025).

# 5 Приложение 1. Техническое задание на тему разрботкаинтеллектуальной системы классификации шахматных фигур

Название системы: Программное решение для классификации шахматных фигур по классу и цвет

Условное обозначение: СКИГО (Система Классификации Изображений шахматных фигур с Глубоким Обучением)  
Основание для разработки: настоящее техническое задание разработано в соответствии с требованиями ГОСТ 34.602-2020

Настоящее техническое задание разработано в соответствии с:

* Заданием на преддипломную практику
* Планом заказчика ЗАО” ЕС-Лизинг”

## 5.1 Цель и назначение создания автоматизированной системы

Целью разработки является создание программного решения для автоматической классификации изображений шахматных фигур с использованием современных методов глубокого обучения. Система предназначена для демонстрации практического применения знаний в области машинного обучения и разработки программного обеспечения. Система предполагает собой использование в организации ЗАО “ЕС-Лизинг” в клубе “Любители шахмат”, для дальнейшей доработки полноценной системы.

## 2 Требования

#### Требования к составу выполняемых функций

Система должна обеспечивать:

* Загрузку и предобработку изображений в формате JPG, PNG, JPEG
* Автоматическую классификацию изображений с выводом подробной информации, включающей в себя имя файла, результат классификации класса, точность определения и результат классификации по цвету
* Визуализацию результатов классификации в виде изображения, классификации и цвета
* Сохранение истории классификаций в виде лог файла, включающий в себя, название файла, результат классификации, цвет, а также, дату обращения к системе

## 3 Требования к организации входных данных

* Поддерживаемые форматы: JPG, PNG, JPEG
* Точность классификации не ниже 50%;
* Максимальный размер файла: 5 МБ
* Размер обрабатываемых изображений: 224x224 пикселей
* Шахматная фигура на изображении должна быть белой или черной

## 4 Требования к организации входных данных

* Результаты классификации в формате JSON с указанием имени файла, класса фигуры, цвета и вероятности
* Визуализация результатов классификации с указанием вероятности
* Визуализация графиков

## 5 Требования к функциональности системы

* Точность классификации: не менее 80%
* Время обработки одного изображения: не более 5 секунды
* Устойчивость к различным форматам входных данных

## 6 Требование к аппаратному обеспечению

* Процессор: поддерживающий инструкции AVX2
* Оперативная память: не менее 8 ГБ
* Видеокарта: с поддержкой CUDA (для ускорения вычислений)
* Свободное место на диске: не менее 15 ГБ

## 7 Требование к программному обеспечению

* Операционная система: Windows 10/11 или Linux
* Python 3.8 или выше
* TensorFlow 2.x
* CUDA Toolkit (для работы с GPU)

## 8 Требования к информационной и программной совместимости

* Совместимость с веб-браузерами: Chrome, Firefox, Edge, Yandex (В зависимости от браузера по умолчанию), для работы с веб-версией приложения
* Совместимость с форматами данных JSON, CSV

## 9 Этапы разработки

1. Подготовительный этап  
   Включающий в себя требования от заказчика и формирование технического задания (ТЗ).
2. Разработка

Разработка архитектуры программного обеспечения, включающей структуру нейронной сети, модули сбора и анализа данных.

1. Тестирование

Проведение нагрузочного тестирования производительности, и корректности работы программы, а также обучение и тестирование нейронной сети.

1. Внедрение и интеграция  
   Внедрение решения в инфраструктуру заказчика

**Требования к документированию**

Документация должна включать:

1. Руководство пользователя
2. Руководство администратора
3. Описание используемых алгоритмов и моделей нейронных сетей;

**Порядок контроля и приёмки**

1. Проверка соответствия программного обеспечения требованиям, указанным в ТЗ.
2. Оценка точности работы нейронной сети на тестовых и реальных данных.

# Приложение 2 Исходный код приложения

*import* streamlit *as* st

*from* tensorflow.keras.models *import* load\_model

*from* tensorflow.keras.preprocessing *import* image

*from* PIL *import* Image

*import* numpy *as* np

*import* pandas *as* pd

*import* os

*from* datetime *import* datetime

*import* matplotlib.pyplot *as* plt

*import* csv

*# Настройка темы и цветов*

st.set\_page\_config(*page\_title*="Chess Classifier Pro", *page\_icon*=, *layout*="centered")

*# Кастомные стили*

st.markdown("""

    <style>

    .main {

        background-color: #f5f5f5;

    }

    .stTitle {

        color: #2c3e50;

    }

    .stSubheader {

        color: #34495e;

    }

    .stSuccess {

        background-color: #d4edda;

        color: #155724;

    }

    .stInfo {

        background-color: #cce5ff;

        color: #004085;

    }

    .stWarning {

        background-color: #fff3cd;

        color: #856404;

    }

    .stError {

        background-color: #f8d7da;

        color: #721c24;

    }

    </style>

    """, *unsafe\_allow\_html*=True)

*# Настройка стиля графиков*

plt.style.use('seaborn-v0\_8')

plt.rcParams['figure.facecolor'] = '#f5f5f5'

plt.rcParams['axes.facecolor'] = '#ffffff'

plt.rcParams['axes.edgecolor'] = '#2c3e50'

plt.rcParams['text.color'] = '#2c3e50'

plt.rcParams['axes.labelcolor'] = '#2c3e50'

print(plt.style.available)

st.title(" Определение шахматной фигуры — Pro-версия")

*# Модель и классы*

model = load\_model("final\_model.h5")

class\_labels = {

    'bishop': 'Слон 🐘',

    'knight': 'Конь 🐴',

    'pawn': 'Пешка 🧍‍♂️',

    'queen': 'Ферзь 👑',

    'rook': 'Ладья 🏰'

}

log\_file = "predictions\_log.csv"

*# Автоматическая инициализация лога, если файл отсутствует или повреждён*

*if* not os.path.exists(log\_file):

*with* open(log\_file, 'w', *encoding*='utf-8-sig', *newline*='') *as* f:

        writer = csv.writer(f)

        writer.writerow(["Файл", "Класс", "Цвет", "Уверенность"])

*else*:

*# Проверяем корректность заголовков*

*try*:

*with* open(log\_file, 'r', *encoding*='utf-8-sig') *as* f:

            reader = csv.reader(f)

            headers = next(reader)

*if* headers != ["Файл", "Класс", "Цвет", "Уверенность"]:

*raise* ValueError

*except* Exception:

*with* open(log\_file, 'w', *encoding*='utf-8-sig', *newline*='') *as* f:

            writer = csv.writer(f)

            writer.writerow(["Файл", "Класс", "Цвет", "Уверенность"])

*# Определение цвета и вывод кропа*

def detect\_color\_preview(*uploaded\_file*):

    st.write(" detect\_color\_preview ВЫЗВАНА")

*try*:

        img = Image.open(uploaded\_file).convert("L")

        st.write(" Файл открыт через PIL, размер:", img.size)

        arr = np.array(img)

        h, w = arr.shape

        cx, cy = w // 2, h // 2

        s = min(h, w) // 2

        crop = arr[cy - s//2:cy + s//2, cx - s//2:cx + s//2]

        crop\_preview = Image.fromarray(crop)

        crop\_valid = crop[crop < 240]

*if* len(crop\_valid) == 0:

            st.warning(" Центр изображения слишком светлый. Цвет определить не удалось.")

*return* "Не удалось определить цвет ❔", 0, crop\_preview

        mean = np.mean(crop\_valid)

        color = "Чёрная " *if* mean < 127 *else* "Белая ♙"

*return* color, mean, crop\_preview

*except* Exception *as* e:

        st.error(f" Не удалось открыть изображение: {e}")

*return* "Ошибка ", 0, None

*# 🧾 История с фильтрацией*

*if* os.path.exists(log\_file):

*try*:

*# Пробуем разные кодировки*

        encodings = ['utf-8-sig', 'utf-8', 'cp1251']

        df\_log = None

*for* encoding *in* encodings:

*try*:

                df\_log = pd.read\_csv(log\_file, *encoding*=encoding)

*break*

*except* UnicodeDecodeError:

*continue*

*if* df\_log is None:

*raise* ValueError("Не удалось прочитать файл ни с одной кодировкой")

*# Проверяем наличие всех необходимых колонок*

        required\_columns = ["Файл", "Класс", "Цвет", "Уверенность"]

        missing\_columns = [col *for* col *in* required\_columns *if* col not in df\_log.columns]

*if* missing\_columns:

*# Если каких-то колонок нет, добавляем их*

*for* col *in* missing\_columns:

                df\_log[col] = "—"

*# Фильтры*

        st.subheader(" История предсказаний")

        color\_filter = st.selectbox("Фильтр по цвету", *options*=["Все", "Чёрная", "Белая"])

        class\_filter = st.selectbox("Фильтр по классу", *options*=["Все"] + list(class\_labels.values()))

*# Применяем фильтры*

*if* color\_filter != "Все":

            df\_log = df\_log[df\_log["Цвет"].str.contains(color\_filter, *na*=False)]

*if* class\_filter != "Все":

            df\_log = df\_log[df\_log["Класс"] == class\_filter]

*if* not df\_log.empty:

            st.dataframe(df\_log.tail(5), *use\_container\_width*=True)

*else*:

            st.info("История пуста")

*except* Exception *as* e:

        st.warning(f" Проблема с чтением лога: {str(e)}")

*# Создаем новый файл лога*

*with* open(log\_file, 'w', *encoding*='utf-8-sig', *newline*='') *as* f:

            writer = csv.writer(f)

            writer.writerow(["Файл", "Класс", "Цвет", "Уверенность"])

*# Загрузка и предсказание для нескольких изображений*

uploaded\_files = st.file\_uploader("Загрузите изображения фигур", *type*=["jpg", "jpeg", "png"], *accept\_multiple\_files*=True)

*if* uploaded\_files:

    predictions = []

*for* uploaded\_file *in* uploaded\_files:

        st.image(uploaded\_file, *caption*=f" Загружено: {uploaded\_file.name}", *use\_container\_width*=True)

        st.write(" Пытаемся определить цвет через detect\_color\_preview")

        fig\_color, brightness, center\_crop = detect\_color\_preview(uploaded\_file)

*if* center\_crop:

            st.image(center\_crop, *caption*=f" Центр для анализа цвета (яркость: {brightness:.2f})", *use\_container\_width*=True)

        color\_emoji = *if* "Чёрная" in fig\_color *else* "⚪️" *if* "Белая" in fig\_color *else* "❔"

        st.markdown(f"### {color\_emoji} Цвет фигуры: \*\*{fig\_color}\*\*")

        img = image.load\_img(uploaded\_file, *target\_size*=(224, 224))

        x = image.img\_to\_array(img)

        x = np.expand\_dims(x, *axis*=0) / 255.0

        prediction = model.predict(x)[0]

        idx = np.argmax(prediction)

        confidence = float(np.max(prediction)) \* 100

        predicted\_class = class\_labels[list(class\_labels.keys())[idx]]

*# Топ-3 вероятности*

        top3\_idx = np.argsort(prediction)[::-1][:3]

        top3\_labels = [list(class\_labels.values())[i] *for* i *in* top3\_idx]

        top3\_probs = [prediction[i]\*100 *for* i *in* top3\_idx]

        st.info("Топ-3 вероятности:")

*for* label, prob *in* zip(top3\_labels, top3\_probs):

            st.write(f"{label}: {prob:.2f}%")

        st.success(f" Модель определила: \*\*{predicted\_class}\*\*, цвет: \*\*{fig\_color}\*\*, уверенность: \*\*{confidence:.2f}%\*\*")

*# График уверенности*

        st.subheader(" Уверенность по классам")

        fig, ax = plt.subplots()

        ax.bar(class\_labels.values(), prediction \* 100)

        ax.set\_ylabel('%')

        ax.set\_title('Уверенность модели')

        st.pyplot(fig)

*# Добавляем результаты в список*

        predictions.append({

            "Файл": uploaded\_file.name,

            "Класс": predicted\_class,

            "Цвет": fig\_color,

            "Уверенность": f"{confidence:.2f}%"

        })

*# Показываем результаты всех изображений*

    st.subheader(" Результаты сравнения изображений:")

*for* pred *in* predictions:

        st.write(f"Файл: {pred['Файл']} — \*\*{pred['Класс']}\*\* (Цвет: {pred['Цвет']}), Уверенность: {pred['Уверенность']}")

*# Сохраняем лог*

    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    log\_df = pd.DataFrame(predictions)

*if* os.path.exists(log\_file):

*try*:

            current = pd.read\_csv(log\_file, *encoding*='utf-8')

*if* set(current.columns) == set(log\_df.columns):

                log\_df.to\_csv(log\_file, *mode*='a', *index*=False, *header*=False, *encoding*='utf-8-sig')

*else*:

                log\_df.to\_csv(log\_file, *index*=False, *encoding*='utf-8-sig')

*except*:

            log\_df.to\_csv(log\_file, *index*=False, *encoding*='utf-8-sig')

*else*:

        log\_df.to\_csv(log\_file, *index*=False, *encoding*='utf-8-sig')

# Приложение 3. Руководство администратора

**Руководство администратора системы классификации шахматных фигур**

**1. Введение**

Настоящее руководство предназначено для администраторов системы классификации шахматных фигур. Система представляет собой веб-приложение, разработанное на базе Python с использованием Streamlit и TensorFlow, предназначенное для автоматического определения типа и цвета шахматных фигур на изображениях.

**2. Назначение и условия применения**

Система предназначена для автоматизации процесса классификации шахматных фигур по их типу (слон, конь, пешка, ферзь, ладья) и цвету (белые/черные). Основные функции системы:

- Загрузка и обработка изображений шахматных фигур

- Определение типа фигуры с использованием модели глубокого обучения

- Определение цвета фигуры

- Сохранение результатов классификации

- Визуализация статистики и истории предсказаний

Условия применения:

- Операционная система: Windows/Linux

- Python 3.8 или выше

- Минимум 4 ГБ оперативной памяти

- Доступ к интернету для установки зависимостей

- Поддерживаемые форматы изображений: JPG, JPEG, PNG

**3. Подготовка к работе**

**3.1 Состав дистрибутива:**

- app.py - основной файл веб-приложения

- final\_model.h5 - обученная модель классификации

- chess\_classifier\_gui.py - графический интерфейс

- train\_model.py - скрипт для обучения модели

- predictions\_log.csv - файл для хранения истории предсказаний

- stats.json - файл статистики

**3.2 Установка зависимостей:**

pip install -r requirements.txt

Основные зависимости:

- tensorflow

- streamlit

- pillow

- numpy

- pandas

- matplotlib

**3.3 Проверка работоспособности:**

1. Запуск приложения:

streamlit run app.py

2. Открытие в браузере: http://localhost:8501

3. Загрузка тестового изображения шахматной фигуры

4. Проверка корректности определения типа и цвета фигуры

**4. Описание операций**

**4.1 Установка и развертывание**

1. Клонирование репозитория

2. Создание виртуального окружения Python

3. Установка зависимостей

4. Запуск приложения

5. Настройка прав доступа к файлам/

**4.2 Настройка системных файлов**

1. Конфигурация app.py:

- Настройка путей к модели

- Настройка параметров логирования

- Настройка интерфейса

2. Управление файлом логов:

- Мониторинг размера predictions\_log.csv

- Регулярное архивирование логов

- Очистка старых записей

**4.3 Управление моделью**

1. Обновление модели:

- Замена файла final\_model.h5

- Проверка совместимости версий

- Тестирование на валидационном наборе

2. Обучение новой модели:

- Подготовка датасета

- Запуск train\_model.py

- Валидация результатов

**5. Аварийные ситуации**

**5.1 Сбои в работе приложения**

1. Проверка логов ошибок

2. Перезапуск приложения

3. Проверка доступности ресурсов

**5.2 Проблемы с моделью**

1. Восстановление из резервной копии

2. Переобучение модели

3. Проверка входных данных

**5.3 Проблемы с базой данных**

1. Восстановление из резервной копии

2. Проверка целостности данных

3. Очистка поврежденных записей

**6. Рекомендации по обслуживанию**

**6.1 Регулярное обслуживание**

1. Ежедневное резервное копирование

2. Мониторинг использования ресурсов

3. Очистка временных файлов

**6.2 Обновление системы**

1. Регулярное обновление зависимостей

2. Обновление модели при необходимости

3. Тестирование после обновлений

**6.3 Безопасность**

1. Регулярная смена паролей

2. Мониторинг доступа

3. Обновление сертификатов

**4. Руководство пользователя системы классификации шахматных фигур**

# Приложение 4. Руководство пользователя корпоративной информационной системы

Настоящее руководство предназначено для пользователей системы классификации шахматных фигур. Система представляет собой веб-приложение, разработанное на базе Python с использованием Streamlit и TensorFlow, предназначенное для автоматического определения типа и цвета шахматных фигур на изображениях. Документ описывает основные функциональные возможности системы, а также предоставляет пошаговые инструкции по ее эксплуатации. Предполагается, что пользователи обладают базовыми знаниями работы с веб-приложениями и знакомы с основными принципами обработки изображений. Для полноценного использования всех возможностей системы рекомендуется ознакомиться с сопроводительной эксплуатационной документацией, включающей подробное описание функций и интерфейса.

2. Назначение и условия применения

Система классификации шахматных фигур разработана для автоматизации процесса определения типа (слон, конь, пешка, ферзь, ладья) и цвета (белые/черные) шахматных фигур на изображениях. Она предназначена для использования в образовательных целях, а также для интеграции с существующими образовательными платформами. Основные функции системы включают загрузку изображений, их автоматическую классификацию по типу и цвету фигур, сохранение результатов в базе данных и предоставление статистических отчетов.

Применение системы возможно при наличии доступа к интернету через современный веб-браузер, таком как Chrome, Firefox, Edge или Yandex. Для обеспечения корректной работы системы требуется стабильное интернет-соединение. Кроме того, необходимо, чтобы пользователи имели базовые навыки работы с веб-приложениями и понимали основные принципы обработки изображений. Система поддерживает работу с изображениями в форматах JPG, JPEG и PNG, при этом максимальный размер загружаемого файла не должен превышать 5 МБ.

3. Подготовка к работе

Перед началом эксплуатации системы необходимо выполнить несколько подготовительных шагов. Пользователю следует открыть веб-браузер и перейти по адресу *http://localhost:8501*. После загрузки интерфейса системы станет доступна функция загрузки изображений для классификации. Важно убедиться, что загружаемые изображения соответствуют поддерживаемым форматам и не превышают максимальный размер в 5 МБ.

Для проверки работоспособности системы рекомендуется загрузить одно из предоставленных тестовых изображений шахматной фигуры. После загрузки изображение будет отображено на экране, а система выполнит классификацию, предоставив результат с указанием предсказанного типа фигуры, ее цвета и уровня уверенности модели. Убедившись в корректности работы системы, пользователь может приступать к регулярному использованию функционала для своих задач.

4. Описание операций

Руководство подробно описывает основные операции, доступные пользователю для взаимодействия с системой, а также последовательность действий, необходимых для их выполнения.

Загрузка изображений для классификации начинается с перехода на главную страницу системы. Пользователь нажимает кнопку "Загрузите изображения фигур" и выбирает один или несколько файлов с локального устройства. После подтверждения выбора система автоматически обрабатывает изображения и отображает результаты классификации. Для каждой фигуры система показывает предсказанный тип, цвет и уровень уверенности модели в процентах. Дополнительно отображается топ-3 вероятности для каждого класса, что позволяет оценить надежность предсказания.

Просмотр результатов классификации осуществляется на той же странице, где загружены изображения. Пользователь может ознакомиться с результатами, а при необходимости сохранить их для дальнейшего использования. Дополнительно предусмотрен раздел "История предсказаний", где отображается история классификаций. В этом разделе представлены таблицы с подробной информацией о каждом предсказании, а также графические отчеты, демонстрирующие распределение предсказанных типов фигур и их динамику во времени.

Работа со статистикой позволяет проводить глубокий анализ результатов классификации. Пользователь может использовать фильтры по цвету и типу фигуры для поиска конкретных результатов. Система предоставляет визуализацию данных в виде графиков, что облегчает анализ и интерпретацию результатов. Регулярное изучение статистики помогает отслеживать эффективность работы системы и выявлять возможные проблемы в классификации.

5. Аварийные ситуации

В случае возникновения непредвиденных ситуаций при работе с системой, пользователю следует следовать определенным инструкциям для минимизации последствий и быстрого восстановления работоспособности.

При проблемах с загрузкой изображений рекомендуется проверить соответствие формата и размера файла требованиям системы. Если ошибка сохраняется, следует перезагрузить веб-страницу и повторить попытку загрузки. В случае повторных сбоев необходимо обратиться к администратору системы с подробным описанием проблемы.

Если результаты классификации отображаются некорректно, следует убедиться в стабильности интернет-соединения и обновить страницу браузера. При сохранении проблемы рекомендуется связаться с технической поддержкой для диагностики и устранения неисправности. Особое внимание следует уделить качеству загружаемых изображений, так как это напрямую влияет на точность классификации.

В случае проблем с отображением интерфейса пользователю следует попробовать очистить кэш браузера или использовать альтернативный браузер. Если проблема сохраняется, необходимо обратиться к администратору системы для получения дополнительной помощи.

6. Рекомендации по освоению

Для эффективного использования системы классификации шахматных фигур пользователям рекомендуется регулярно практиковаться в загрузке различных изображений для закрепления навыков работы с системой. Освоение инструментов фильтрации и сортировки в разделе статистики позволит проводить более глубокий анализ результатов. Регулярное изучение истории предсказаний поможет лучше понимать работу системы и выявлять возможные проблемы.

Оптимизация работы с системой включает в себя предварительную подготовку изображений, использование оптимальных форматов файлов и группировку похожих изображений для более эффективной обработки. Регулярная проверка результатов классификации поможет поддерживать высокое качество работы системы.

В случае возникновения вопросов или трудностей пользователи могут воспользоваться встроенным разделом помощи в системе или обратиться к администратору. Регулярное обновление знаний о новых функциях и возможностях системы позволит максимально полно использовать ее потенциал.