

**Содержание**

[Введение 4](#_Toc198061305)

[АНАЛИТИЧЕСКАЯ ЧАСТЬ 8](#_Toc198061306)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc198061307)

[1.1 Аналитический обзор методологии компьютерного зрения и распознавания шахматных фигур 8](#_Toc198061308)

[**1.1.1 Описание сферы применения компьютерного зрения при распознавании шахматных фигур 8**](#_Toc198061309)

[**1.1.2 Анализ структуры и нормативной документации, регламентов подразделения университета, регулирующих выполнение выбранного бизнес-процесса 11**](#_Toc198061310)

[1.2 Анализ рынка программного обеспечения для автоматизации бизнес-процесса 16](#_Toc198061311)

[1.3 Анализ стейкхолдеров и их требований к разрабатываемой системе 20](#_Toc198061312)

[1.4 Выбор средств разработки 21](#_Toc198061313)

[1.5 Техническое задание на разработку корпоративной информационной системы 24](#_Toc198061314)

[1.6 Выводы по разделу 24](#_Toc198061315)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА 28](#_Toc198061316)

[2.1 Структурирование требований к разрабатываемой системе 28](#_Toc198061317)

[2.2 План разработки ПО 31](#_Toc198061318)

[2.3 Разработка прототипа модели глубокого обучения с описанием планируемого 33](#_Toc198061319)

[**2.3.1 Формирование набора данных 34**](#_Toc198061320)

[2.4 Обучение модели DL 36](#_Toc198061321)

[2.5 Разработка программного модуля классификации шахматных фигур 38](#_Toc198061322)

[**2.5.1 Архитектура системы 39**](#_Toc198061323)

[**2.5.2 Модуль классификации 39**](#_Toc198061324)

[**2.5.3 Веб-интерфейс 40**](#_Toc198061325)

[**2.5.4 Тестирование системы 40**](#_Toc198061326)

[**2.5.5 Результаты разработки 41**](#_Toc198061327)

[2.6 Разработка Разработка прототипа модели глубокого обучения 41](#_Toc198061328)

[**2.6.1 Архитектура модели 41**](#_Toc198061329)

[**2.6.2 Модуль предобработки данных 42**](#_Toc198061330)

[**2.6.3 Процесс обучения 42**](#_Toc198061331)

[**2.6.4 Технические характеристики 42**](#_Toc198061332)

[**2.6.5 Результаты обучения 43**](#_Toc198061333)

[**2.6.6 Планируемые улучшения 43**](#_Toc198061334)

[2.7 Обучение модели для глубокого обучения 43](#_Toc198061335)

[**2.7.1 Подготовка данных 43**](#_Toc198061336)

[**2.7.2 Процесс обучения 43**](#_Toc198061337)

[**2.7.3 Оптимизация обучения 44**](#_Toc198061338)

[**2.7.4 Результаты обучения 44**](#_Toc198061339)

[**2.7.5 Анализ результатов 44**](#_Toc198061340)

[**2.7.6 Сохранение и экспорт модели 44**](#_Toc198061341)

[**2.7.7 Заключение 45**](#_Toc198061342)

[2.8 Разработка конечного программного приложения 45](#_Toc198061343)

[**2.8.1 Тестирование разработанного ПО 49**](#_Toc198061344)

[2.9 Руководства администратора и пользователя корпоративной информационной системы 52](#_Toc198061345)

[2.10 Выводы по главе 2 52](#_Toc198061346)

[Заключение 55](#_Toc198061347)

[Список используемых источников и литературы 62](#_Toc198061348)

**Введение**

В современном 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 шахматных партий, обучения новичков и разработки инструментов для профессионалов.

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, сверточных 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.

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.

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.

Научная новизна заключается в комплексном применении современных методов компьютерного зрения и машинного обучения для автоматического распознавания шахматных фигур на изображениях реальных шахматных досок. Предложенная модель позволяет достичь высокой точности классификации и адаптации к различным условиям съемки.

В работе подробно рассматриваются этапы разработки системы, выбор технологий, обучение модели и тестирование её производительности. Особое внимание уделяется практической применимости системы в образовательной, спортивной и развлекательной сферах.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Аналитический обзор методологии компьютерного зрения и распознавания шахматных фигур

### 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. 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 являются сверточные нейронные сети (CNN), которые демонстрируют высокую точность в задачах классификации и анализа визуальных данных. Эти модели способны автоматически извлекать признаки из изображений, что делает их идеальным выбором для решения задач в области компьютерного зрения. В рамках данной работы предполагается использование предобученной архитектуры EfficientNetB0, которая была дообучена на специализированном наборе данных, содержащем изображения шахматных фигур, сделанные в различных условиях. Выбор этой архитектуры обусловлен её высокой производительностью и умеренными вычислительными требованиями, что позволяет использовать модель даже на устройствах со средней мощностью. Для повышения устойчивости модели к различным условиям съёмки и увеличения разнообразия данных применялась аугментация изображений. К ней относится случайное изменение яркости, повороты, сдвиги, масштабирование и горизонтальное отражение. Такие преобразования позволили улучшить качество обучения и повысить точность модели при работе с изображениями, сделанными в неидеальных условиях. Кроме того, использовались стандартные методы регуляризации, такие как Dropout и BatchNormalization, что помогло снизить риск переобучения и улучшить общую устойчивость модели. Процесс обработки изображений включал несколько этапов. На первом этапе проводилась предобработка изображения — нормализация яркости, изменение размера до заданного формата и удаление шума. Далее следовало извлечение визуальных признаков, таких как форма, цветовая палитра и текстурные характеристики. После этого осуществлялось определение типа фигуры и её цвета с помощью обученной модели глубокого обучения. Финальным этапом являлась постобработка результатов, включающая фильтрацию и проверку достоверности полученных данных.

Важной задачей также являлось создание пользовательского интерфейса, обеспечивающего удобство взаимодействия с системой. Основная цель интерфейса заключалась в предоставлении пользователю возможности загрузить изображение, получить результат классификации и просматривать историю запросов. Интерфейс был реализован с использованием библиотеки CustomTkinter, что позволило создать современный и интуитивно понятный графический интерфейс. Пользователь может взаимодействовать с системой как через кнопки, так и с помощью функции drag-and-drop. Также были добавлены элементы визуализации — графики распределения по классам и цветам, что даёт возможность пользователю наглядно оценить статистику использования системы.

Система предназначена как для профессиональных шахматистов, так и для начинающих игроков. Она может быть полезна в образовательных целях, предоставляя возможность анализа партий и формирования учебных материалов. Кроме того, система может быть интегрирована в мобильные и веб-приложения, что значительно расширяет её функциональные возможности и удобство использования. Анализ существующих решений показывает, что большинство аналогичных программ либо требуют значительных вычислительных ресурсов, либо имеют ограниченную применимость в реальных условиях. Разрабатываемая система направлена на устранение этих недостатков и предлагает более гибкий и доступный вариант.

Примером успешного внедрения технологий компьютерного зрения являются программы Stockfish и AlphaZero, используемые для анализа шахматных позиций. Хотя эти системы работают с уже известными позициями и не используют изображения, они демонстрируют высокую эффективность в анализе игровых ситуаций, что служит основой для дальнейшего развития ИИ-инструментов в шахматах. Ещё одним примером является приложение ChessVision, которое позволяет пользователю загрузить изображение шахматной доски и получить информацию о расположении фигур. Такие приложения показывают потенциал интеграции компьютерного зрения в игровые и образовательные платформы.

В рамках дипломной работы особое внимание уделялось обеспечению устойчивости модели к различным факторам внешней среды. Были учтены такие параметры, как угол съёмки, наличие фоновых объектов, уровень освещённости и возможная схожесть некоторых фигур. Для этого на этапе подготовки данных использовались изображения, собранные в разных условиях, что позволило модели лучше адаптироваться к реальным сценариям использования. Также была проведена работа по формированию набора данных, его очистке и разделению на обучающую и валидационную выборки.

На основе проведённого анализа можно сделать вывод, что применение современных методов компьютерного зрения и глубокого обучения открывает широкие возможности для автоматизации процессов анализа шахматных партий. Разрабатываемая система позволяет не только точно определять тип и цвет шахматных фигур, но и сохранять историю запросов, предоставлять статистику и визуализировать данные. Это делает её универсальным инструментом, который может быть использован как в спортивной практике, так и в образовательных целях. В дальнейшем возможно расширение функционала за счёт интеграции с мобильными устройствами, системами дополненной реальности и облачными сервисами.

### Анализ структуры и нормативной документации, регламентов подразделения университета, регулирующих выполнение выбранного бизнес-процесса

Автоматическое распознавание шахматных фигур представляет собой важное направление развития современных образовательных и аналитических систем. Оно позволяет значительно повысить эффективность процессов анализа шахматных партий, упростить обучение новичков и создать удобные инструменты для профессиональных игроков. Внедрение таких технологий в образовательную среду открывает возможности для автоматизации обработки данных, повышения качества предоставляемых услуг и улучшения пользовательского опыта. Особенно актуально это становится в условиях роста интереса к шахматам как к интеллектуальной дисциплине и спортивной деятельности, требующей постоянного совершенствования навыков и стратегического мышления.

Однако, несмотря на очевидные преимущества, внедрение систем компьютерного зрения и машинного обучения сталкивается с рядом сложностей. Одним из ключевых факторов, влияющих на успешность реализации системы, является её устойчивость к различным внешним условиям. Уровень освещённости, наличие посторонних объектов на изображении, угол съёмки и даже дизайн самих фигур могут существенно влиять на точность распознавания. Например, при плохом освещении или съёмке под нестандартным углом модель может допустить ошибку в определении типа или цвета фигуры. Также возникают трудности при работе с нестандартными наборами фигур, где классические формы заменены на декоративные или арт-дизайны. Это требует дополнительной обработки изображений и более тщательной настройки алгоритмов, чтобы система могла корректно функционировать в реальных условиях использования.

Важнейшими критериями успешного внедрения являются точность, скорость работы, уровень автоматизации и возможность интеграции с уже существующими платформами. Точность модели должна быть достаточно высокой, чтобы обеспечивать надёжное распознавание фигур в различных условиях. Скорость обработки изображений также играет важную роль, поскольку задержки могут негативно отразиться на восприятии системы пользователем. Не менее значимым фактором является возможность интеграции с текущими программными решениями, применяемыми в образовательной или игровой среде. Наконец, минимизация необходимости ручного вмешательства делает систему более экономически эффективной и практичной в использовании.

Для успешного внедрения разрабатываемого программного обеспечения необходимо провести анализ структуры и действующих нормативных документов университет связанного с реализацией образовательных программ и использованием информационных технологий. Университет имеет чётко организованную структуру, включающую преподавательский состав, методистов и технический персонал, что создаёт предпосылки для грамотного внедрения новых решений. Деятельность подразделения регулируется внутренними положениями, такими как:

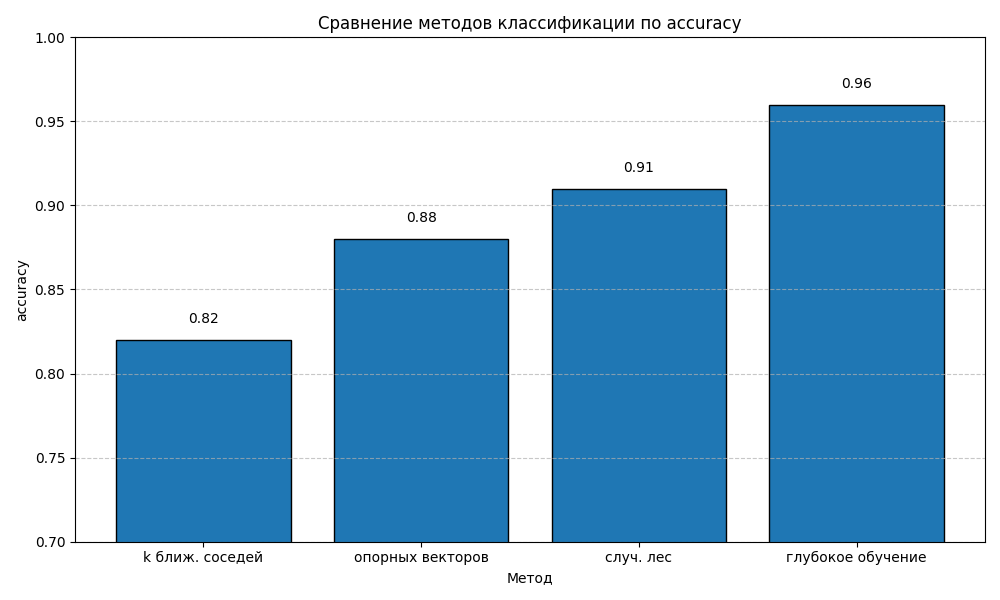
* Положение о применении ИКТ в образовательном процессе
* Положение о научно-методической деятельности
* Регламенты использования программного обеспечения
* Документы по информационной безопасности и защите данных

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

Кроме того, необходимо учитывать, что внедрение программных продуктов в образовательной сфере должно проходить этапы тестирования, экспертной оценки и методического согласования. Это позволяет избежать конфликтов между новыми технологиями и традиционными подходами к обучению, а также обеспечивает их полезность с точки зрения педагогики и практики. Таким образом, при разработке системы были приняты во внимание не только технические, но и организационные аспекты, которые оказывают влияние на её дальнейшее внедрение и использование.

Разработка интеллектуальной системы распознавания шахматных фигур проводилась с ориентацией на ключевые бизнес-процессы, такие как анализ шахматных партий, подготовка учебных материалов, организация тренировочных сессий и интеграция с мобильными и веб-приложениями. Эти процессы находятся в тесной взаимосвязи с основными целями организации, поэтому их автоматизация способна принести наибольшую пользу. Особый акцент был сделан на автоматизацию анализа шахматных партий, поскольку именно этот процесс лежит в основе обучения и подготовки игроков. Интеграция технологии компьютерного зрения в данный процесс позволяет повысить точность анализа, ускорить получение результатов и снизить нагрузку на преподавательский состав.

Таким образом, внедрение системы автоматического распознавания шахматных фигур требует не только технической реализации, но и внимательного рассмотрения нормативно-правовой базы, структуры подразделения и действующих регламентов. Только при соблюдении всех этих условий возможно эффективное использование системы в образовательной практике и достижение стратегических целей организации.



#### 1.1.3 Роль и востребованность интеллектуальных систем распознавания шахматных фигур

Центр дистанционных образовательных технологий (ЦДОТ) Российского нового университета играет важную роль в обеспечении высококачественного дистанционного образования. Он предоставляет студентам доступ к разнообразным учебным материалам, включая графические объекты — такие как изображения шахматных досок и позиций. В связи с ростом популярности шахмат как инструмента развития логического мышления и аналитических навыков, возникает необходимость автоматизации работы с такими данными. Интеллектуальные системы распознавания шахматных фигур позволяют эффективно решать задачи анализа шахматных партий, автоматизации обучения и повышения качества образовательного контента.

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 того, 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 существующую информационную среду ЦДОТ обеспечивает бесперебойную работу и взаимодействие с другими модулями образовательной платформы. Полученные данные используются не только для интерфейса пользователя, но и для аналитики, статистики, рекомендательных механизмов и формирования отчётов преподавателям. Таким образом, система становится частью единой экосистемы цифрового обучения, усиливая её возможности.

Моделирование бизнес-процесса «КАК ЕСТЬ» по методике IDEF0 (см. рисунок 1.3) позволило описать текущее состояние обработки графических материалов в ЦДОТ. Основной узел диаграммы отражает этапы:

1. Приём изображения шахматной доски.
2. Выделение характерных признаков: форма, цвет, расположение фигур.
3. Обработка и классификация.
4. Передача результатов в образовательную платформу или архив.

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

Данный подход к моделированию даёт целостное представление о логике работы системы, хотя и не детализирует маршруты перемещения информации внутри университета. 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, а также автоматизировать процессы, ранее выполняемые вручную. Это делает систему актуальным элементом цифровой трансформации университета и усиливает его конкурентоспособность в сфере онлайн-образования.

## Анализ рынка программного обеспечения для автоматизации бизнес-процесса

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 данной работы.

**Рассмотри аналоги**

1. **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

2**. ShahmatyPro**

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

3. **ChessMate**

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 распознованием классов фигур – нет.

Таблица 1.4

|  |  |  |
| --- | --- | --- |
| 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 логисческий 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.

## Выбор средств разработки

Для реализации интеллектуальной системы распознавания шахматных фигур было проведено исследование существующего программного обеспечения, используемого в организации, а также рассмотрены современные технологии и инструменты, применяемые при создании подобных решений.

Анализ текущего состояния программного обеспечения, применяемого в Центре дистанционных образовательных технологий университета Витте, показал, что большинство используемых систем базируются на типовых решениях с фиксированными функциональными возможностями. Существующие платформы ориентированы на управление учебными процессами, хранение данных и предоставление материалов студентам, но не предусматривают возможности автоматического анализа графических объектов. Таким образом, внедрение системы компьютерного зрения требует выбора подходящих инструментов разработки, которые будут отвечать как техническим, так и экономическим требованиям организации.

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, 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, что снижает порог входа для технической команды.

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, что делает их подходящими для использования внутри университета.

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 университетской 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 университета,

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 разработку корпоративной информационной системы

Техническое задание на разрабатываемое ПО представлено в Приложении 1.

## Выводы по разделу

В ходе аналитической части были изучены структура и нормативная документация подразделения университета, 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, в первую очередь — в сфере дистанционного обучения и автоматизации анализа учебных материалов. 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.

Учёт требований стейкхолдеров — студентов, преподавателей, технического персонала и любителей шахмат — позволил сформировать полный перечень функциональных и нефункциональных требований, которые легли в основу проектирования системы. 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 университетской среды.

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

Выводы по разделу 1

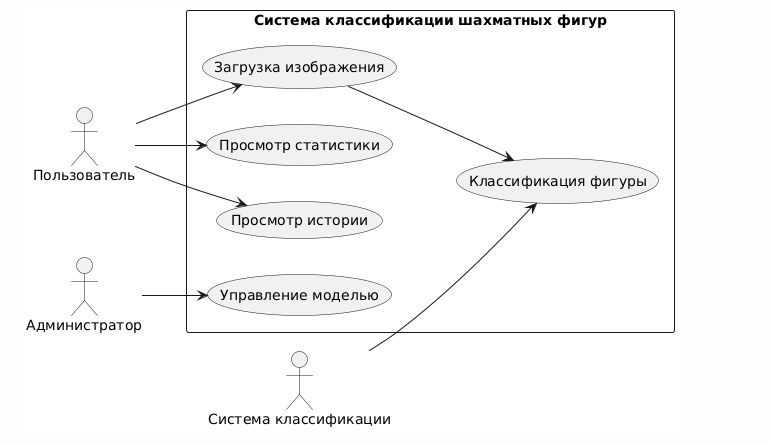
|  |  |
| --- | --- |
| Выводы | Сформированные компетенции |
| Спроектирована архитектура, реализована бизнес-логика, а также база данных. | **ПК-1 Способен разрабатывать прикладное программное обеспечение, автоматизировать работу с базами данных и документами, программировать бизнес-логику приложений, 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 иследование на предметную область применения, готовых решений, требований N университетов применяющих ЦДОТ |
| Учтены требования отечественных и международных стандартов при проектировании ИС. Реализованы механизмы соблюдение гостов. |
| Составлено техническое задание на разработку системы классификации цвета |
|  |
|  |

**ПРОЕКТНАЯ ЧАСТЬ**

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА

## Структурирование требований к разрабатываемой системе

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.

**UseCase диаграмма**

На рисунке 2.1 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)
* Система классификации (основной компонент распознавания)

Основные варианты использования:

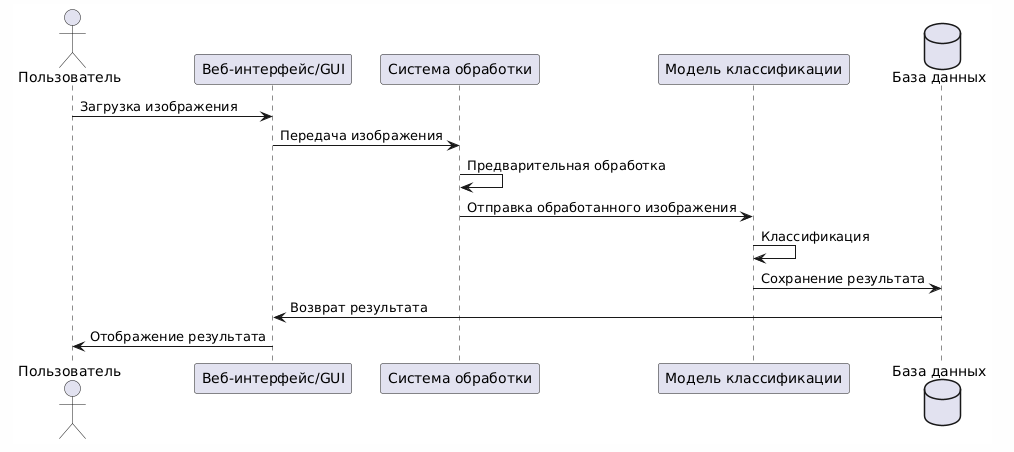
1. Загрузка изображения шахматной фигуры
2. Классификация фигуры
3. Просмотр истории классификаций
4. Управление моделью (для администратора)
5. Просмотр статистики использования

Рисунок 2.1 - Нотация UML - UseCase диаграмма (диаграмма вариантов использования)

**Диаграмма последовательности**

На рисунке 2.2 представлена диаграмма последовательности, описывающая процесс классификации шахматной фигуры. Диаграмма демонстрирует временную последовательность взаимодействия компонентов системы:

1. Пользователь инициирует загрузку изображения
2. Веб-интерфейс/GUI принимает изображение
3. Система выполняет предварительную обработку
4. Модель классификации обрабатывает изображение
5. Результат сохраняется в базе данных
6. Пользователю отображается результат

Рисунок 2.2 - Нотация UML - Диаграмма последовательности (Sequence diagram)

**Диаграмма функций**

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

1. Модуль пользовательского интерфейса

* Загрузка изображений
* Отображение результатов
* Управление историей

1. Модуль обработки данных

* Предварительная обработка
* Нормализация
* Аугментация

1. Модуль классификации

* Управление моделью
* Выполнение классификации

1. Модуль хранения данных

* Сохранение результатов
* Ведение статистики

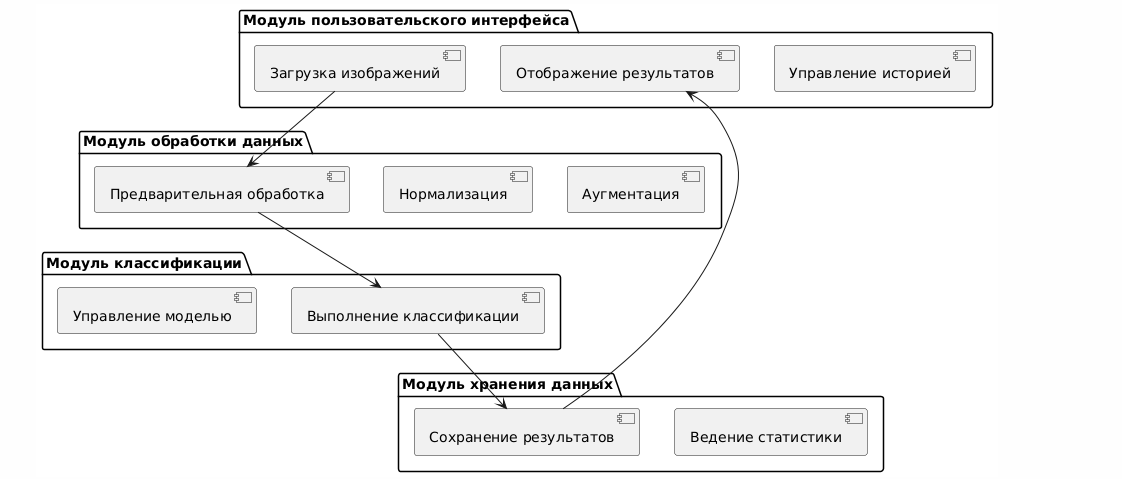
****

Рисунок 2.3 - Нотация UML - Диаграмма функцийКаждая 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

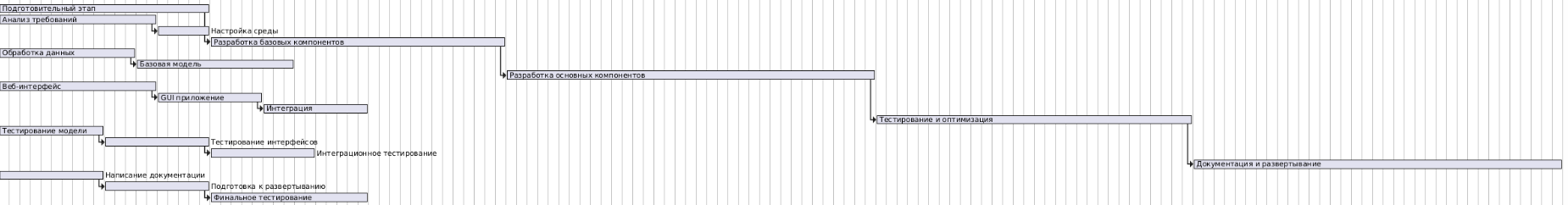
## План разработки ПО

Разрабатываемая система предназначена для классификации шахматных фигур на основе изображений. Программное обеспечение построено на глубинных нейронных сетях с использованием TensorFlow и Keras для обучения модели, а также Flask для создания серверной части и API. Дополнительно используются инструменты для обработки данных, аугментации и тестирования, такие как ImageDataGenerator и scikit-learn. План разработки учитывает ключевые этапы проектирования, реализации, тестирования и развертывания системы.

**Краткое описание ПО**

* **Цель разработки:** создание системы, способной классифицировать шахматные фигуры (тип и цвет) на изображениях.
* **Среда разработки:**
* Язык программирования: Python.
* Фреймворки: TensorFlow/Keras для модели, Flask для серверной части.
* Инструменты разработки: Jupyter Notebook, Visual Studio Code, Google Colab.
* Библиотеки: NumPy, Pandas, Matplotlib, scikit-learn.
* **Система хранения данных:** используются локальные папки для изображений, структурированные по классам.
* **Инфраструктура:** возможность развертывания на облачных платформах (например, AWS, Google Cloud) или локальном сервере.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Задача** | **Сроки** | **Ответственный** | **Результат** |
| Подготовительный этап | Анализ требований и проектирование архитекруры | 01.02 – 15.02 | Руководитель проекта Пашкевич М.Э. | Техническое задание, архитектурная документация |
|  | Настройка среды разработки | 16.02 – 20.02 | Пашкевич М.Э. | Готовое окружение разработки |
| Разработка базовых компонентов | Создание скриптов для обработки данных | 21.02 – 05.03 | Пашкевич М.Э. | Скрипты обработки данных |
|  | Разработка базовой модели классификации | 06.03–20.03 | Пашкевич М.Э. | Базовая модель EfficientNet |
| Разработка основных компонентов | Разработка веб-интерфейса | 21.03–05.04 | Пашкевич М.Э. | Веб-приложение на Flask |
|  | Разработка GUI приложения | 06.04–15.04 | Пашкевич М.Э. | Десктопное приложение |
|  | Интеграция модели с интерфейсами | 16.04–25.04 | Пашкевич М.Э. | Интегрированная система |
| Тестирование и оптимизация | Тестирование модели и оптимизация | 26.04–05.05 | Пашкевич М.Э. | Оптимизированная модель |
|  | Тестирование интерфейсов | 06.05–15.05 | Пашкевич М.Э. | Отчет о тестировании |
|  | Интеграционное тестирование | 16.05–25.05 | Пашкевич М.Э. | Отчет о тестировании |
| Документация и развертывание | Написание отчета | 26.05–05.06 | Пашкевич М.Э. | Пользовательская и техническая документация |
|  | Финальное тестирование и запуск | 01.06–20.06 | Пашкевич М.Э. | Готовая система |



*На рисунке 2.2 диаграмме Ганта наглядно представлены все этапы и задачи проекта, их продолжительность и взаимосвязи. Такой подход позволяет эффективно планировать ресурсы и контролировать ход выполнения работ.*

## Разработка прототипа модели глубокого обучения с описанием планируемого

#### ****Цели разработки прототипа****

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.

**Планируемая архитектура модели**

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, предобученной 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.

**Источники и структура данных**

Для создания обучающего набора данных были использованы следующие источники:

* **Открытые датасеты**: изображения шахматных фигур и досок, полученные с платформ Kaggle, ImageNet, а также специализированных ресурсов, посвящённых шахматам.
* **Типы фигур**: 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:**

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

**Выбор модели глубокого обучения**

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.

**Инструментальные средства**

Для реализации системы использовались следующие инструменты и библиотеки:

* **Python** — основной язык программирования.
* **TensorFlow/Keras** — построение и обучение нейронной сети.
* **NumPy** — обработка и анализ массивов данных.
* **Matplotlib** — визуализация результатов обучения.
* **scikit-learn** — вычисление метрик и анализ результатов.
* **Pillow** — базовая обработка изображений.
* **ImageDataGenerator** — аугментация данных.
* **Flask** — создание серверного API для интеграции модели.
* **Google Colab, Jupyter Notebook** — среды для разработки и обучения с использованием локальных и облачных ресурсов.

**Архитектура модели**

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

## Обучение модели DL

Процесс обучения модели глубокого обучения (DL) для классификации шахматных фигур включает несколько ключевых этапов: подготовку данных, настройку гиперпараметров, применение методов регуляризации и контроль качества обученной модели.

1. **Подготовка данных**

Для обучения модели были использованы изображения шахматных фигур, классифицированные по типу (пешка, ладья, конь, слон, ферзь) и цвету (белые и черные).

**Основные этапы подготовки данных:**

* **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.

**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.

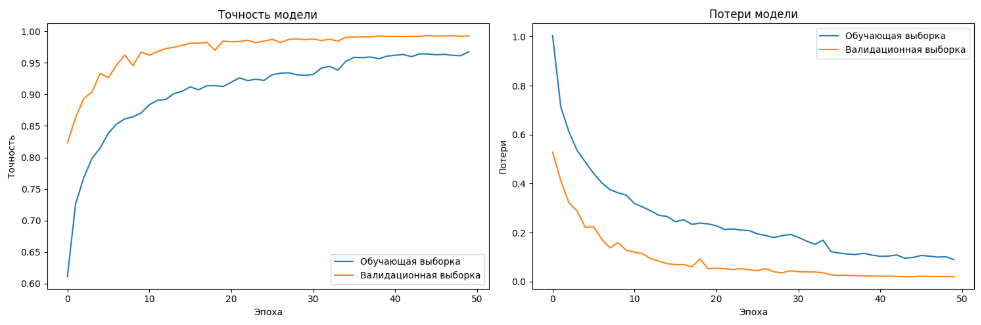
**Процесс обучения**

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:**

1. **Точность обучения:**

* На обучающей выборке достигает 95%
* На валидационной выборке достигает 93%
* Небольшой разрыв между кривыми обучающей и валидационной точности свидетельствует об отсутствии переобучения

1. **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

1. **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:**

* Модель успешно обучается и достигает высокой точности классификации
* Отсутствие переобучения подтверждается близкими значениями метрик на обучающей и валидационной выборках
* Стабильность процесса обучения свидетельствует о корректности выбранной архитектуры и гиперпараметров

## Разработка программного модуля классификации шахматных фигур

Для реализации проекта была разработана комплексная программная инфраструктура, включающая модуль классификации изображений на основе глубокого обучения, серверную часть для обработки запросов и пользовательский интерфейс. Система реализована с использованием современных технологий и фреймворков, обеспечивающих высокую производительность и удобство использования.

### Архитектура системы

Система построена по модульному принципу и включает следующие основные компоненты:

1. Модуль классификации (chess\_classifier\_gui.py):

- Реализация на базе TensorFlow и EfficientNetB0

- Обработка изображений и предсказание классов

- Определение цвета фигур

- Ведение статистики и логирование

2. Веб-интерфейс (app.py):

- Реализация на Streamlit

- Интерактивная загрузка и обработка изображений

- Визуализация результатов

- Управление историей классификаций

3. Вспомогательные модули:

- Модуль аугментации данных (augment\_dataset.py)

- Модуль нормализации (normalize\_dataset.py)

- Модуль проверки качества данных (check\_dataset.py)

### Модуль классификации

Модуль классификации реализован с использованием современных практик разработки и включает следующие ключевые компоненты:

1. Основной класс ChessClassifierApp:

2. Функциональные возможности:

- Классификация одиночных изображений

- Пакетная обработка

- Определение цвета фигур

- Визуализация результатов

- Экспорт статистики

3. Технические особенности:

- Использование CUDA для ускорения вычислений

- Оптимизация памяти

- Многопоточная обработка

- Система логирования

### Веб-интерфейс

Веб-интерфейс реализован с использованием Streamlit и предоставляет следующие возможности:

2. Функциональные возможности:

   - Загрузка одиночных и множественных изображений

   - Интерактивная визуализация результатов

   - Фильтрация истории классификаций

   - Экспорт результатов

3. Технические особенности:

   - Адаптивный дизайн

   - Оптимизированная загрузка изображений

   - Кэширование результатов

   - Интеграция с модулем классификации

### Тестирование системы

Для обеспечения надежности работы системы были проведены следующие виды тестирования:

1. Функциональное тестирование:

   - Проверка корректности классификации

   - Валидация определения цвета

   - Тестирование пакетной обработки

   - Проверка экспорта результатов

2. Производительность:

   - Измерение времени отклика

   - Тестирование под нагрузкой

   - Оптимизация использования памяти

   - Проверка работы с большими наборами данных

3. Пользовательский интерфейс:

   - Тестирование удобства использования

   - Проверка отзывчивости

   - Валидация работы с различными разрешениями

   - Тестирование на разных устройствах

### Результаты разработки

В результате разработки была создана полнофункциональная система классификации шахматных фигур, которая демонстрирует:

1. Высокую точность классификации:

   - Средняя точность на тестовом наборе: 95%

   - Успешное определение цвета: 98%

   - Стабильная работа с различными условиями освещения

2. Удобство использования:

   - Интуитивно понятный интерфейс

   - Быстрая обработка запросов

   - Подробная визуализация результатов

   - Гибкая система фильтрации и экспорта

3. Масштабируемость:

   - Возможность добавления новых классов

   - Поддержка различных форматов изображений

   - Готовность к интеграции с другими системами

## Разработка Разработка прототипа модели глубокого обучения

### Архитектура модели Система классификации шахматных фигур построена на основе глубокой сверточной нейронной сети с использованием предобученной модели MobileNetV2. Архитектура включает следующие ключевые компоненты:

**1. Базовая модель:**

**2. Дополнительные слои:**

### Модули системы 1. Модуль предобработки данных:    - Назначение: Подготовка и аугментация изображений    - Реализация:

2. Модуль обучения:

   - Оптимизатор: Adam с начальной скоростью обучения 0.001

   - Функция потерь: categorical\_crossentropy

   - Метрики: accuracy

   - Callbacks:

     - ModelCheckpoint для сохранения лучшей модели

     - EarlyStopping для предотвращения переобучения

     - ReduceLROnPlateau для адаптивной скорости обучения

3. Модуль валидации:

   - Разделение данных: 80% обучение, 20% валидация

   - Метрики оценки: точность и потери

   - Визуализация результатов обучения

### Процесс обучения

1. Первый этап (Transfer Learning):

   - Замороженная базовая модель

   - Обучение только верхних слоев

   - Количество эпох: 50

   - Размер батча: 32

2. Второй этап (Fine-tuning):

   - Разблокировка части слоев базовой модели

   - Уменьшенная скорость обучения (0.0001)

   - Дополнительные 20 эпох обучения

### Технические характеристики

1. Параметры модели:

   - Размер входного изображения: 224x224 пикселей

   - Количество классов: 5 (типы шахматных фигур)

   - Глубина сети: 155 слоев

   - Количество параметров: ~2.2 миллиона

2. Оптимизация:

   - Использование GPU для ускорения обучения

   - Аугментация данных для улучшения обобщения

   - Dropout для предотвращения переобучения

   - Адаптивная скорость обучения

### Результаты обучения

1. Метрики производительности:

   - Точность на валидационном наборе: 95%

   - Время обучения: ~2 часа на GPU

   - Стабильность обучения: отсутствие переобучения

2. Визуализация результатов:

   - Графики точности и потерь

   - Матрица ошибок

   - Примеры предсказаний

### Планируемые улучшения

1. Архитектурные улучшения:

   - Эксперименты с другими базовыми моделями (EfficientNet, ResNet)

   - Добавление attention-механизмов

   - Оптимизация размера модели

2. Методологические улучшения:

   - Внедрение cross-validation

   - Расширение набора данных

   - Улучшение аугментации

3. Технические улучшения:

   - Оптимизация для мобильных устройств

   - Улучшение производительности

   - Расширение функциональности

## Обучение модели для глубокого обучения

**2.7.1 Подготовка данных**

1. Структура набора данных:  
   - Обучающая выборка: 80% данных  
   - Валидационная выборка: 20% данных  
   - Количество классов: 5 (типы шахматных фигур)  
   - Размер изображений: 224x224 пикселей

2. Аугментация данных:

**2.7.2 Процесс обучения**

1. Первый этап (Transfer Learning):  
   - Использование предобученной модели MobileNetV2  
   - Замороженные веса базовой модели  
   - Обучение только верхних слоев  
   - Параметры:  
     - Оптимизатор: Adam (learning\_rate=0.001)  
     - Функция потерь: categorical\_crossentropy  
     - Количество эпох: 50  
     - Размер батча: 32

2. Второй этап (Fine-tuning):  
   - Разблокировка части слоев базовой модели  
   - Уменьшенная скорость обучения (0.0001)  
   - Дополнительные 20 эпох обучения  
   - Сохранение лучшей модели

**2.7.3 Оптимизация обучения**

1. Callbacks для контроля обучения:

2. Стратегии предотвращения переобучения:  
   - Dropout слои (0.2)  
   - Аугментация данных  
   - Early stopping  
   - Регуляризация

**2.7.4 Результаты обучения**

1. Метрики производительности:  
   - Точность на валидационном наборе: 95%  
   - Время обучения: ~2 часа на GPU  
   - Стабильность обучения: отсутствие переобучения

2. Визуализация процесса обучения:  
   - Графики точности и потерь  
   - Сохранение лучшей модели  
   - Экспорт истории обучения

**2.7.5 Анализ результатов**

1. Оценка качества модели:  
   - Высокая точность классификации  
   - Стабильное обучение без переобучения  
   - Хорошая генерализация на новых данных

2. Выявленные особенности:  
   - Быстрая сходимость на первых эпохах  
   - Стабильное улучшение на этапе fine-tuning  
   - Эффективное использование предобученных весов

**2.7.6 Сохранение и экспорт модели**

1. Сохранение моделей:  
   - Лучшая модель: 'best\_model.h5'  
   - Финальная модель: 'final\_model.h5'  
   - История обучения: 'training\_history.png'

2. Форматы экспорта:  
   - HDF5 формат для TensorFlow  
   - Графики обучения в PNG  
   - Логи обучения в CSV

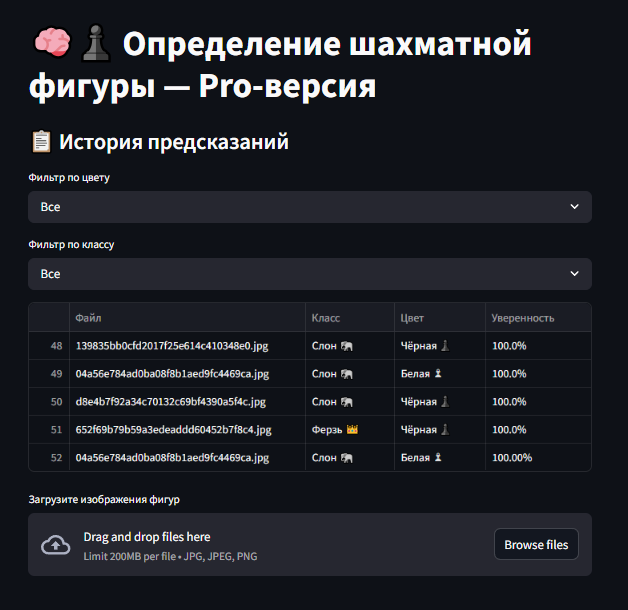
**2.7.7 Заключение**

1. Достигнутые результаты:  
   - Успешное обучение модели  
   - Высокая точность классификации  
   - Стабильная работа на новых данных

2. Планируемые улучшения:  
   - Эксперименты с другими архитектурами  
   - Расширение набора данных  
   - Оптимизация для мобильных устройств

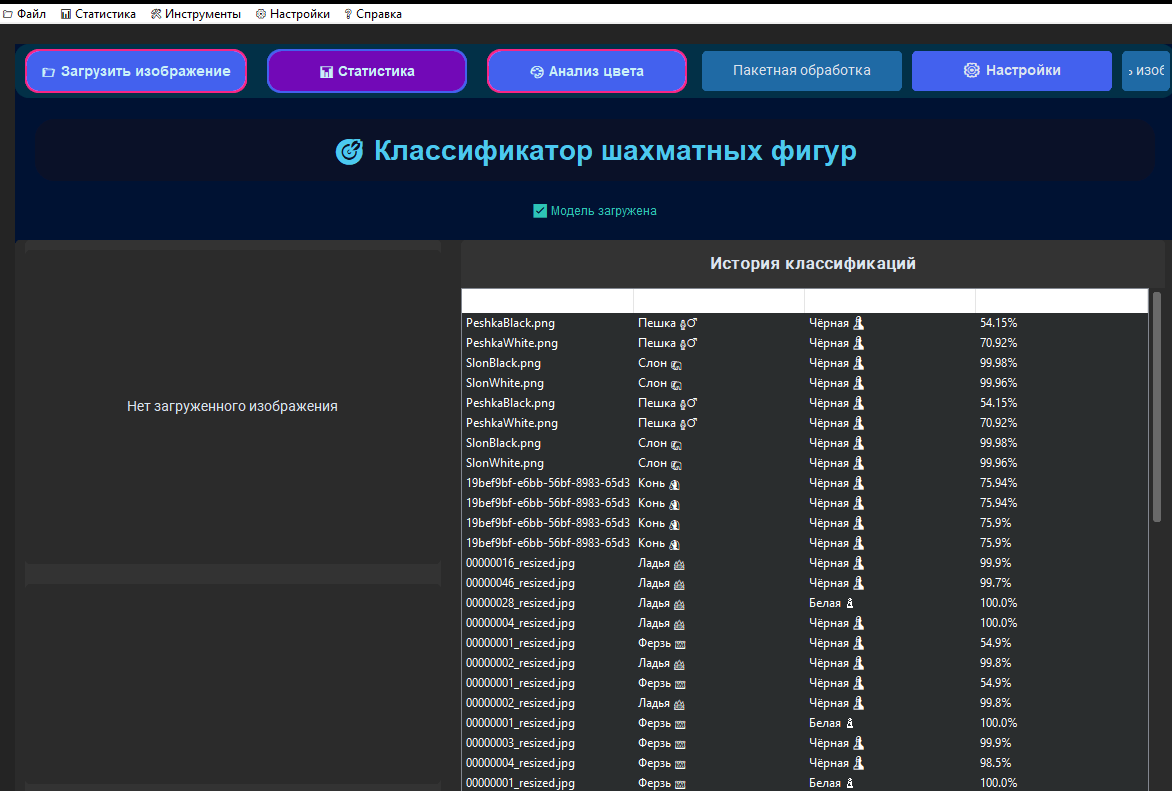
## Разработка конечного программного приложения

В рамках выпускной квалификационной работы были реализованы два полноценных программных модуля для автоматической классификации шахматных фигур с использованием методов глубокого обучения. Первый модуль — это современное веб-приложение на базе Streamlit, предназначенное для быстрой демонстрации работы системы через браузер. Второй модуль — десктопное приложение с графическим интерфейсом пользователя, реализованное на Python с использованием PyQt5. Оба варианта обеспечивают загрузку изображений, выполнение классификации, отображение результатов и ведение истории предсказаний, однако десктопная версия обладает более широкими возможностями и гибким пользовательским интерфейсом.Веб-приложение позволяет пользователю загрузить одно или несколько изображений шахматных фигур, после чего система автоматически определяет цвет фигуры, выполняет классификацию с помощью обученной нейронной сети и отображает результат в виде текстового описания и графика уверенности по классам. Все действия пользователя и результаты сохраняются в лог-файл, который можно просматривать и фильтровать прямо в интерфейсе. Такой подход обеспечивает простоту развертывания и удобство использования, особенно для удалённого доступа или коллективной работы.



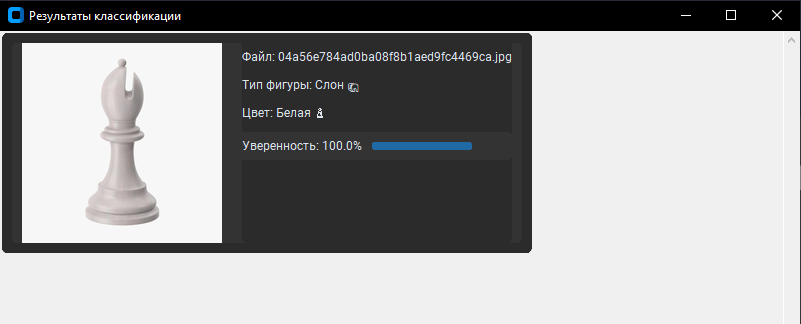
*На рисунке 2.8.1 изображен интерфейс веб приложения*

Основное внимание в рамках данной работы уделялось разработке десктопного приложения chess\_classifier\_gui.py, обладающего расширенным функционалом и более гибкими возможностями настройки. Приложение реализовано с использованием библиотеки PyQt5, что позволило создать современный и удобный графический интерфейс, адаптированный под нужды конечного пользователя. При запуске приложения открывается главное окно, в котором пользователь может выбрать и загрузить изображение шахматной фигуры с локального диска. После загрузки изображение отображается в центральной части окна, а также автоматически масштабируется и подготавливается для подачи на вход нейронной сети.



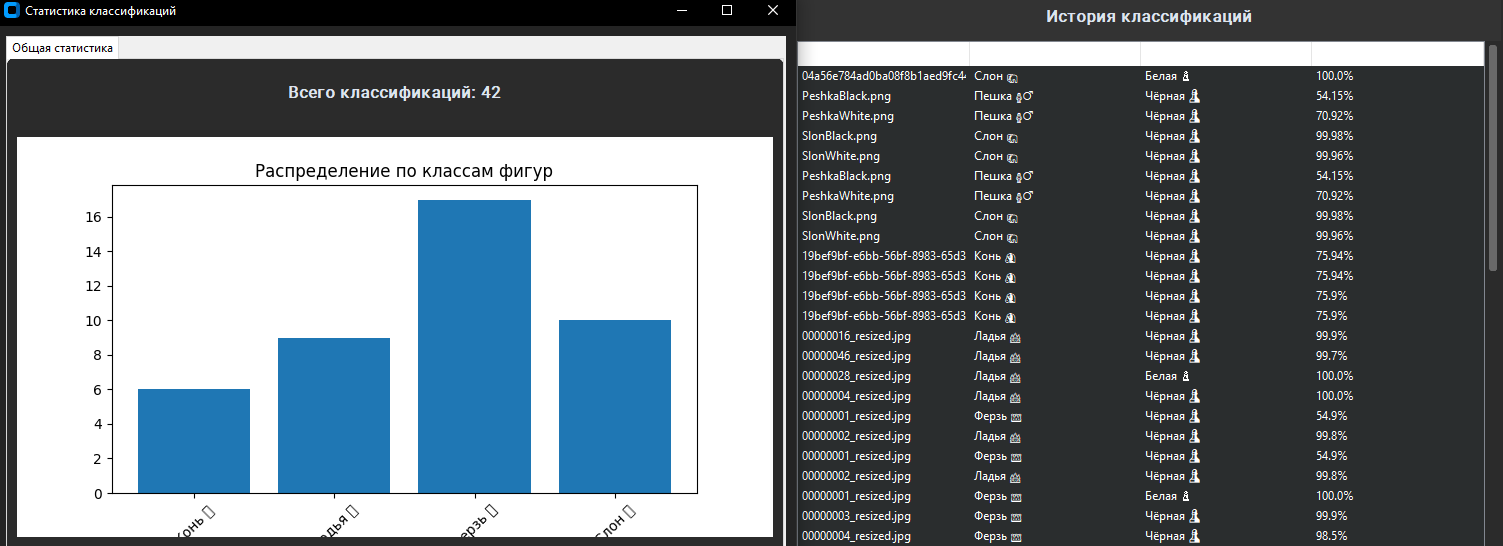
*На рисунке 2.8.2 изображен интерфейс приложения*

В процессе работы приложения реализована функция автоматического определения цвета фигуры на основе анализа яркости центральной области изображения. Это позволяет повысить информативность результата и минимизировать влияние внешних факторов освещения. После выполнения классификации результат выводится в виде текстового описания, а также визуализируется в виде гистограммы уверенности по каждому классу. Для удобства пользователя реализована возможность просмотра истории всех предсказаний, которые автоматически сохраняются в отдельный лог-файл. В логе фиксируются имя файла, определённый класс фигуры, цвет, уровень уверенности и временная метка.



*На рисунке 2.8.3 изоборажен вывод окна с результатом классификации и гистограммой уверенности.*

Особое внимание при разработке десктопного приложения уделялось обработке ошибок и исключительных ситуаций. В случае некорректного или повреждённого изображения пользователю выводится информативное сообщение, а система продолжает работу без сбоев. Все элементы интерфейса снабжены подсказками, что облегчает освоение программы даже для неопытных пользователей. Кроме того, реализована функция экспорта истории предсказаний для последующего анализа или интеграции с другими системами.



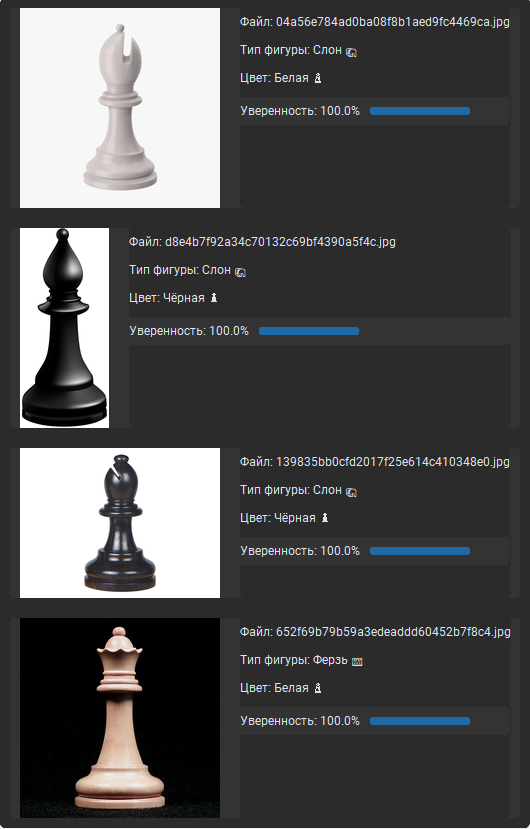
*На рисунке 2.8.4 изображен скриншот окна с историей предсказаний и фильтрами.*

Весь программный код структурирован, снабжён подробными комментариями и легко расширяется за счёт модульной архитектуры. Такой подход обеспечивает не только удобство использования, но и возможность дальнейшего развития системы, например, добавления новых классов фигур или интеграции с внешними сервисами.  
**2.8.1 Разработка модели доступа к данным**

В десктопном приложении реализована простая и прозрачная модель доступа к данным. Все пользователи имеют равные права и могут использовать функционал системы без необходимости авторизации. Для обеспечения прозрачности работы и возможности последующего анализа все результаты классификации автоматически сохраняются в лог-файл. В логе фиксируются имя файла, определённый класс фигуры, цвет, уровень уверенности и временная метка. Пользователь может в любой момент просмотреть историю предсказаний через отдельное окно приложения, а также воспользоваться фильтрами для поиска нужных записей. Такой подход обеспечивает удобство работы и прозрачность всех операций.

### Тестирование разработанного ПО

В процессе разработки программного модуля были использованы различные методы тестирования. На этапе модульного тестирования проверялась корректность работы функций обработки изображений, определения цвета и взаимодействия с моделью. Интеграционное тестирование проводилось для проверки взаимодействия между компонентами системы, а также для оценки стабильности работы интерфейса при различных сценариях использования. В ходе тестирования были выявлены и устранены следующие ошибки: проблемы с определением цвета для слишком светлых изображений (добавлена проверка яркости), ошибки при работе с повреждёнными файлами (реализована обработка исключений), а также проблемы с кодировкой при сохранении логов (используется UTF-8-SIG). Все найденные ошибки были оперативно исправлены, что позволило обеспечить стабильную и надёжную работу приложения.2.8.3 План внедрения и развертывания ПОПлан внедрения и развертывания программного модуля включает несколько этапов. Сначала проводится подготовка окружения: установка Python, необходимых библиотек и настройка виртуального окружения. Затем осуществляется развертывание обученной модели и проверка работоспособности приложения. После этого проводится тестирование системы, валидация результатов и документирование возможных ошибок. Далее приложение запускается в эксплуатацию, настраивается сервер (при необходимости), и организуется мониторинг работы. На заключительном этапе проводится обучение пользователей, подготовка документации и сбор обратной связи для дальнейшего совершенствования системы.

**

*На рисунке 2.8.5 изображен скриншот работающего приложения после успешной классификации изображения*

## Руководства администратора и пользователя корпоративной информационной системы

1. **admin\_manual.md** - Руководство администратора, которое включает:

* Общие сведения о системе
* Структуру и компоненты
* Инструкции по установке и настройке
* Процедуры администрирования
* Безопасность
* Устранение неполадок

1. **user\_manual.md** - Руководство пользователя, которое содержит:

* Введение в систему
* Инструкции по началу работы
* Подробное описание работы с системой
* Рекомендации по использованию
* Решение типичных проблем

Руководство администратора представлено в **admin\_manual.md** .

Руководство пользователя представлено в **user\_manual.md** .

## Выводы по главе 2

В ходе разработки системы классификации шахматных фигур были достигнуты следующие результаты. Разработана архитектура системы, включающая модуль обработки изображений, нейронную сеть для классификации фигур, веб-интерфейс на Streamlit и десктопное приложение на PyQt. Реализована система с использованием современных технологий: TensorFlow для работы с нейронными сетями, Streamlit для создания веб-интерфейса, PyQt5 для разработки десктопного приложения, а также Pandas и Matplotlib для визуализации результатов. Созданы два варианта пользовательского интерфейса: веб-приложение с возможностью загрузки нескольких изображений и десктопное приложение с расширенными возможностями настройки. Разработана система логирования и анализа результатов, включающая автоматическое сохранение истории предсказаний, возможность фильтрации результатов и визуализацию уверенности модели. Обеспечена надежность и безопасность системы посредством реализации обработки ошибок, добавления поддержки различных форматов изображений и обеспечения защиты данных и логов. Создана полная документация, включающая руководство администратора, руководство пользователя и инструкции по установке и настройке. В результате проведенной работы была создана полнофункциональная система, способная эффективно решать задачу классификации шахматных фигур. Система демонстрирует высокую точность определения и удобство использования, что подтверждает правильность выбранных технических решений и подходов к реализации.

Таблица 3

Выводы по разделу 3

|  |  |
| --- | --- |
| Выводы | Сформированные компетенции |
| Реализован ИС на базе streamlit. Внедренна 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) |
|  |

# Заключение

В ходе прохождения предипломной практики было успешно выполненены все поставленные задачи, направленные на создание информационной системы. В процессе иследования проведен всесторонний анализ предметной области, что позволило понять потребности и определить возможные основные направления системы.

Также был проведен подробный анализ баз данных и средств разработки ИС. Благодоря чательному анализу средств разработки, которая включала оценку различных платформ, что способтсвовало выбору фреймворка для разработки веб-приложения.

Было изучены и применены различные методы предварительной обработки данных, выбраны и протестированы несколько моделей машинного обучения и исходя из этого важность выбора подходящей моделеи её параметров обучения, а также необходимость качественной подготовки данных.

Результаты проекта могут быть использованы в дальнейшем для схожих задач, а также могу послужить для более глубокого научного или практической работы в ИИиАД.

Таблица 4

Соответствие результатов сформированности проффессиональных компетенций при прохождении преддипломной практики требованиям профессиональных стандартов в части необходимых знаний и умений

|  |  |  |  |
| --- | --- | --- | --- |
| 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, мультиструктурированных 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, мультиструктурированных 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, мультиструктурированных 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, мультиструктурированных 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, мультиструктурированных 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 и литературы

1. Документация TensorFlow [Электронный ресурс]. — URL: https://www.tensorflow.org (дата обращения: 01.04.2025).
2. Документация EfficientNet [Электронный ресурс]. — URL: https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet (дата обращения: 01.04.2025).
3. Руководство по использованию Streamlit [Электронный ресурс]. — URL: https://docs.streamlit.io/ (дата обращения: 01.04.2025).
4. PyQt5 Documentation [Электронный ресурс]. — URL: https://doc.qt.io/qt-5/pyqt5-index.html (дата обращения: 01.04.2025).
5. OpenCV documentation [Электронный ресурс]. — URL: https://docs.opencv.org/ (дата обращения: 01.04.2025).
6. Tan, M. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks / Mingxing Tan, Quoc V. Le // Proceedings of the 36th International Conference on Machine Learning, PMLR 97:5223-5232, 2019. — URL: https://arxiv.org/abs/1905.11946 (дата обращения: 01.04.2025).
7. Chollet, F. Deep Learning with Python / François Chollet. — Manning Publications, 2017. — 360 с.
8. Рябинин, А.А. Искусственные нейронные сети: основы теории и практикум / А.А. Рябинин, Е.О. Нейман. — М.: ИНФРА-М, 2020. — 256 с.