ОРИГИНАЛЬНАЯ СТАТЬЯ
УДК 004.8:631.15
https://doi.org/10.26897/2687-1149-2026-5-15-27

Разработка AI-агента поддержки принятия решений для промышленного садоводства на основе больших языковых моделей (LLM) и компьютерного зрения

А.И. Кутырёв
Федеральный научный агроинженерный центр ВИМ; г. Москва, Россия
alexeykutyrev@gmail.com; https://orcid.org/0000-0001-7643-775X

Аннотация. Современное промышленное садоводство оснащают автоматизированными метеостанциями, датчиками мониторинга состояния растений и почвы, а также IoT-платформами. Развитие цифровизации производственных процессов садоводства обусловливает необходимость разработки интеллектуальной системы поддержки принятия решений. Применение методов обработки естественного языка (NLP) с использованием больших языковых моделей (LLM), компьютерного зрения и анализа данных позволило разработать программный комплекс «AI-агент поддержки принятия решений» для садоводства. В составе AI-агента реализованы механизм генерации с дополненным поиском (RAG), позволяющий получать ответы на профессиональные вопросы, интерфейс анализа изображений растений на базе YOLO, модуль мониторинга с настройкой уведомлений и интеграцией с внешними источниками (MQTT, API), а также инструменты А/Б тестирования для сравнения эффективности ответов. AI-агент построен на открытых библиотеках Python, предоставляет веб-интерфейс (Gradio) и REST API. Экспериментальная оценка показала, что применение RAG повышает релевантность ответов: в 84% случаев эксперты отдали предпочтение ответам, дополненным информацией из базы знаний (доверительный интервал для 95% составил [78,2%; 89,8%] при p менее 0,001). Распознавая объекты яблоневого сада, модуль компьютерного зрения на основе YOLO достиг точности 0,82 и полноты 0,76. Система мониторинга обеспечивает прием данных через MQTT и HTTP с пропускной способностью до 500 записей/с и отправляет оповещения в Telegram со средней задержкой 2 с. Разработанный AI-агент можно использовать как исследовательский инструмент для проведения научных экспериментов и создания коммерческих решений в области точного садоводства: например, автоматизированных рабочих мест агронома, платформ для мониторинга растений, сервисов дистанционной диагностики болезней, систем раннего оповещения о неблагоприятных событиях (заморозки, засуха, вредители) и модулей прогнозирования урожайности.

Ключевые слова: садоводство; анализ изображений растений; мониторинг; AI-агент поддержки принятия решений; LLM; RAG; YOLO

Финансирование. Работа выполнена при поддержке Минобрнауки РФ в рамках Государственного задания ФГБНУ «Федеральный научный агроинженерный центр ВИМ» (тема № FGUN-2025-0011).

Для цитирования: Кутырёв А.И. Разработка интеллектуальной системы поддержки принятия решений для промышленного садоводства на основе больших языковых моделей (LLM) и компьютерного зрения // Агроинженерия. 2026. Т. 28, № 5. С. 15-27. https://doi.org/10.26897/2687-1149-2026-5-15-27

ORIGINAL ARTICLE

LLM and computer vision-based intelligent decision support system for digital horticulture

A.I. Kutyrev
Federal Scientific Agroengineering Center VIM; Moscow, Russia
alexeykutyrev@gmail.com; https://orcid.org/0000-0001-7643-775X

Abstract. Modern industrial horticulture is increasingly integrating automated weather stations, soil and plant sensors, and the Internet of Things (IoT) platforms. This pervasive digitalization necessitates the development of intelligent decision support systems (IDSS) capable of interpreting complex agro-data. By leveraging natural language processing (NLP) via Large Language Models (LLMs), computer vision, and real-time data analysis, the author has developed the “AI Decision Support Agent” – a specialized software package for digital horticulture.

© Кутырёв А.И., 2026

The system features a Retrieval-Augmented Generation (RAG) mechanism for expert-level query resolution, a YOLO-based computer vision interface for plant phenotyping and object detection, and a monitoring module with MQTT and API integration for real-time sensor data acquisition. Additionally, an A/B testing framework was implemented to evaluate the qualitative accuracy of system responses. Built using open-source Python libraries, the agent provides a Gradio-based web interface and a scalable REST API. Experimental evaluation demonstrated that the RAG mechanism significantly enhances response relevance: in 84% of cases, experts preferred RAG-augmented responses over standard LLM outputs (95% confidence interval (CI) [78.2%, 89.8%]; $p < 0.001$). For apple orchard object detection, the YOLO-based vision module achieved a precision of 0.82 and a recall of 0.76. The data acquisition system maintains a throughput of up to 500 records per second via MQTT and HTTP, with notification delay averaging 2 seconds. The developed IDSS serves as a versatile tool for both academic research and commercial applications in precision horticulture, including automated agronomist workstations, plant monitoring platforms, remote disease diagnostics, early-warning systems for biotic and abiotic stresses (frost, drought, pests), and yield forecasting modules.

Keywords: horticulture; plant image analysi; monitoring; AI decision support agent; large language models (LLM); retrieval-augmented generation (RAG); YOLO

Funding. The work was supported by the Ministry of Science and Higher Education of the Russian Federation within the framework of the State Assignment of the Federal Scientific Agroengineering Center VIM (Theme No. FGUN-2025-0011).

For citation: Kutyrev A.I. LLM and computer vision-based intelligent decision support system for digital horticulture. *Agricultural Engineering* (*Moscow*). 2026;28(5):15-27. (In Russ.). https://doi.org/10.26897/2687-1149-2026-5-15-27

Введение

В современное промышленное садоводство внедряются автоматизированные метеостанции, стационарные и мобильные датчики для мониторинга состояния растений и почвы, датчики фитосанитарного контроля, а также IoT-платформы, объединяющие устройства в единую сенсорную сеть [1-3]. В результате формируются большие объемы разнородных данных (Big Data), требующие эффективных методов обработки. Для анализа данных и планирования выполнения технологических операций используется специализированное программное обеспечение [4, 5]. Чтобы использовать эти данные, требуется понимание биологических особенностей растений и факторов, влияющих на продуктивность насаждений [6]. Внедрение цифровых технологий сдерживается в связи с недостатком специалистов, способных интерпретировать поступающую информацию для принятия управленческих решений, а также ввиду отсутствия единой платформы, объединяющей существующие информационные системы. В результате данные остаются невостребованными, а решения принимаются несвоевременно и на основе неполной информации, что приводит к потерям урожая и нерациональному использованию ресурсов.

В исследованиях в области создания систем поддержки принятия решений [7-9] для классификации болезней растений по фотографиям применяют сверточные нейронные сети (CNN). Данные методы успешно применяются для анализа состояния плодов и листьев яблони [10, 11]. Однако в отличие от экспертных систем, учитывающих множество факторов, CNN-модели не могут интерпретировать результат в комплексе с другими данными – они способны распознавать только те признаки заболеваний, на которых обучались [12]. Тем не менее технологии дообучения с низким рангом (LoRA) [13] позволяют адаптировать предобученные модели под предметную область с минимальными вычислительными затратами. Кроме того, в сельском хозяйстве стали применять большие языковые модели (LLM) с дополненным поиском (Retrieval-Augmented Generation, RAG) для генерации ответов с использованием информации из справочников и научных статей [14, 15]. В «умном сельском хозяйстве» широко применяют системы мониторинга на основе MQTT и анализа потоковых данных [16].

Интеграция LLM с данными датчиков в реальном времени остается актуальной научной задачей. В исследованиях [17, 18] представлены архитектуры мультимодальных ассистентов, сочетающих анализ изображений и текста. Несмотря на активное развитие отдельных направлений, комплексные решения, объединяющие методы обработки естественного языка, компьютерного зрения и потокового анализа данных, в едином веб-интерфейсе отсутствуют.

Цель исследований: разработка программного комплекса «AI-агент поддержки принятия решений», объединяющего функции консультационного сервиса на основе RAG, автоматический сбор и анализ данных с датчиков, а также мультимодальный анализ изображений растений на базе компьютерного зрения

и LLM для проведения экспериментов (А/Б тестирование) и механизма обучения модели под конкретные условия хозяйства.

Практическая реализация AI-агента позволит агроному получать оперативные, научно обоснованные рекомендации, основанные как на актуальных данных мониторинга, так и на проверенных источниках знаний.

Материалы и методы

Разработка AI-агента выполнена на языке Python с использованием широкого спектра библиотек. Система построена по модульному принципу и включает в себя такие компоненты, как модуль взаимодействия с пользователем (веб-интерфейс и REST API), модуль языковой модели, модуль базы знаний, модуль компьютерного зрения, модуль мониторинга и уведомлений, модуль логирования и А/Б-тестирования, модуль обучения, модуль администрирования (рис. 1). Все модули интегрированы и обмениваются данными через единую систему баз данных.

Модуль базы знаний и RAG реализован на основе векторной базы данных ChromaDB. Для оценки смысловой близости между текстовыми фрагментами используется косинусная метрика сравнения их векторных представлений (эмбеддингов), создаваемых моделью SentenceTransformer intfloat/multilingual-e5-small размерностью 384. Процесс добавления документа включает в себя извлечение текстового содержимого в зависимости от формата исходного файла. Для текстовых файлов (.txt) выполняется чтение в кодировках UTF-8, cp1251, latin1 и др. с использованием библиотеки chardet. Из PDF-документов текст извлекается построчно с помощью PyPDF2. Для файлов Microsoft Word (.docx) применяется библиотека python-docx для чтения параграфов. Таблицы Excel (.xlsx) обрабатываются библиотекой pandas с извлечением всех листов и преобразованием содержимого в текстовое представление с разделителями табуляции. CSV-файлы анализируются с автоматическим определением разделителя (csv.Sniffer) и последующим построчным чтением. HTML-страницы обрабатываются библиотекой BeautifulSoup4 с удалением скриптов и стилей. После извлечения текст сегментируется на фрагменты. Размер фрагмента

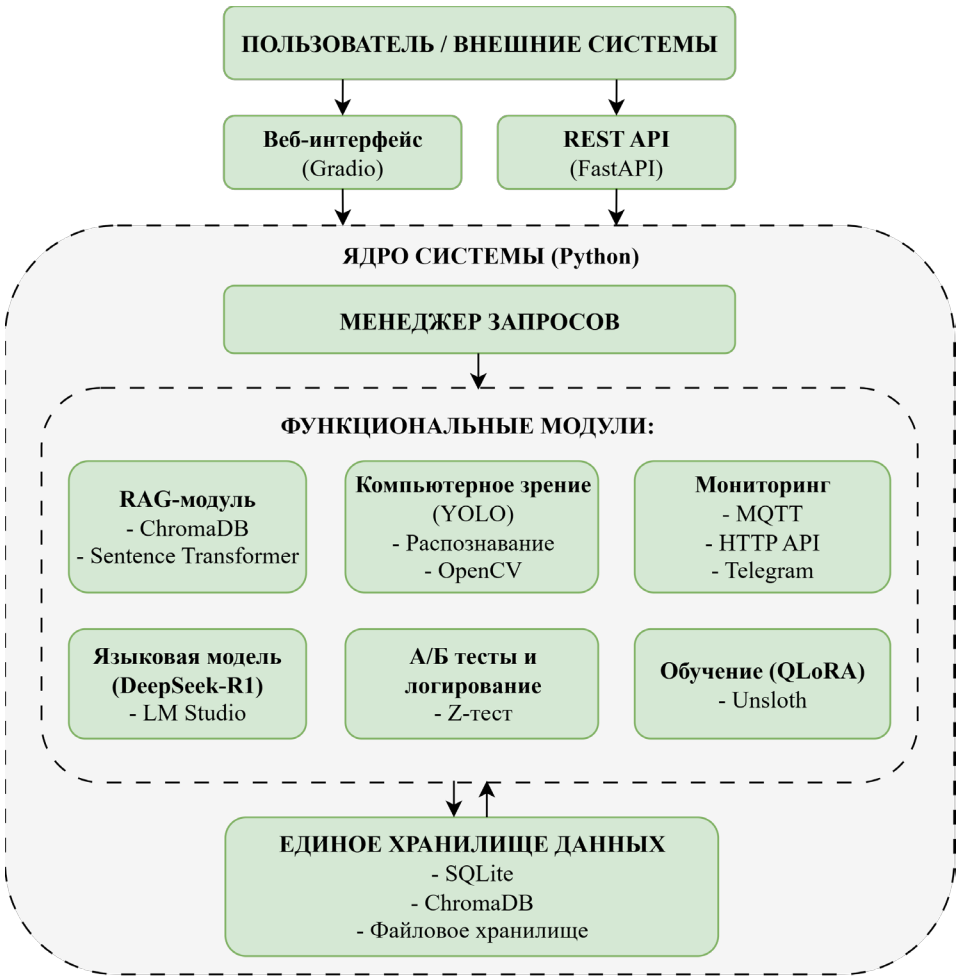


Рис. 1. Архитектура системы

Fig. 1. System architecture

и величина перекрытия между соседними фрагментами задаются пользователем (по умолчанию 500 символов с перекрытием 50 символов).

Алгоритм разбиения стремится сохранить целостность слов: граница фрагмента устанавливается по последнему пробелу в пределах заданной длины. Перекрытие необходимо для сохранения контекста на границах и предотвращения потери релевантной информации при поиске. Каждый фрагмент преобразуется в векторное представление с помощью модели intfloat/multilingual-e5-small. Сформированные векторы вместе с идентификатором (хеш от имени файла и номера фрагмента), текстом фрагмента и метаданными (имя исходного файла, номер фрагмента) сохраняются в ChromaDB. При выполнении поиска запрос пользователя аналогичным образом преобразуется в вектор той же моделью. ChromaDB выполняет поиск k ближайших векторов (по умолчанию k = 3) на основе косинусного сходства по формуле:

$$\text{cosine}_{\text{similarity (a, b)}} = \frac{a \cdot b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}, \quad (1)$$

где a, b $\in R^n$ – векторные представления (эмбеддинги) запроса и фрагмента документа, полученные с помощью модели intfloat/multilingual-e5-small, $n = 384$ – размерность эмбеддингов.

Результат принимает значения в диапазоне [–1, 1], при этом значение, близкое к 1, указывает на высокую смысловую близость. Найденные три наиболее релевантных фрагмента передаются языковой модели, подставляются в системный промпт (запрос к модели) в качестве контекста.

В качестве базовой языковой модели AI-агента использована DeepSeek-R1 (версия deepseek/deepseek-r1-0528-qwen3-8b, 8 млрд параметров). Модель загружается локально в 4-битном формате через приложение LM Studio (HTTP-сервер, совместимый с API OpenAI), что обеспечивает достаточную производительность для задач консультирования на русском языке при умеренных аппаратных требованиях (рекомендуется GPU с 8+ ГБ видеопамяти) (рис. 2).

Для режима RAG разработан специальный системный промпт (инструкция или входные текстовые данные, которые отправляются нейросети для выполнения конкретной задачи), предписывающий модели отвечать строго в рамках тематики садоводства, указывать источники информации и формулировать краткие научно обоснованные ответы. Для свободного диалога, не ограниченного предметной областью, используется отдельный промпт без тематических ограничений. Дообучение модели под конкретные задачи реализовано с помощью библиотеки Unsloth, обеспечивающей применение метода QLoRA. Процесс дообучения начинается со сбора датасета из логов взаимодействий с AI-агентом; из таблицы базы данных SQLite извлекаются пары «вопрос-ответ», которые экспортируются в формат JSONL. Базовая

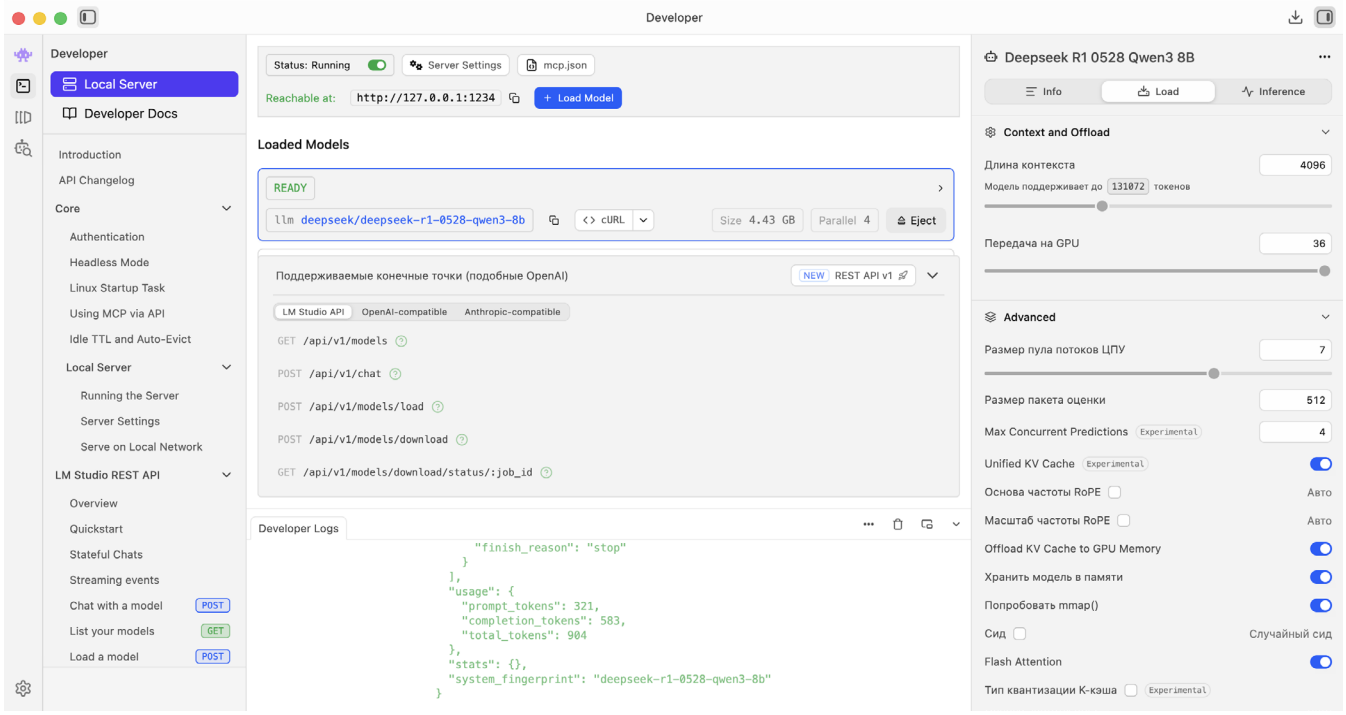


Рис. 2. Приложение LM Studio с запущенным локальным сервером с языковой моделью DeepSeek-R1

Fig. 2. LM Studio application with a local server running the DeepSeek-R1 language model

модель загружается с применением 4-битного квантования для снижения потребления видеопамяти. Для обучения используется класс SFTTrainer из библиотеки TRL со следующими гиперпараметрами: размер пакета данных (batch size) – 2; накопление градиентов (gradient accumulation) – 4; скорость обучения (learning rate) – 0,0002; количество шагов – 60; оптимизатор – AdamW 8-bit. По завершении обучения полученные адаптеры сохраняются в указанную директорию. Описанный метод обеспечивает возможность дообучения языковой модели на данных, характерных для конкретного хозяйства, что способствует повышению релевантности, стилистической точности и содержательной глубины генерируемых ответов. При этом требуемый объем видеопамяти не превышает 5 ГБ.

В основе модуля компьютерного зрения лежат сверточные нейронные сети семейства YOLO последних поколений, поддерживаемые фреймворком Ultralytics: YOLOv8, YOLOv10, YOLO11, YOLO12 и YOLO26. В качестве основной модели детекции выбрана YOLO26n (nano-версия), которая изначально предобучена на наборе данных COCO, а затем дообучена на специализированной выборке изображений, содержащих фотографии яблок, цветов, листьев и плодов [8, 10, 12]. Выбор YOLO26n обусловлен оптимальным соотношением точности (Precision = 0,82) и скорости инференса (0,05 с на кадр на GPU), что является важным для работы в режиме реального времени.

Дообучение выполнялось в течение 300 эпох с использованием оптимизатора AdamW (начальная скорость обучения – 0,001 с распадом по косинусному закону), размер пакета данных (Batch size) составил 16 изображений. Для компенсации дисбаланса классов (доминирование класса «apple») применялась взвешенная функция потерь, а также аугментация данных (горизонтальное отражение, поворот на ±15°, изменение яркости и контрастности, добавление шума). Модель загружается однократно при запуске приложения и постоянно находится в оперативной памяти для обеспечения быстрого отклика. Алгоритм обработки изображения включает в себя несколько этапов, исходное RGB-изображение преобразуется в цветовое пространство BGR, требуемое библиотекой OpenCV, после чего выполняется инференс – этап распознавания YOLO с настраиваемыми порогами уверенности (conf) и пересечения (iou). Результатами работы модели являются координаты ограничивающих рамок, идентификаторы классов и значения уверенности для каждого распознанного объекта. На основе этих данных на изображение наносятся ограничивающие рамки, после чего оно конвертируется обратно в RGB для корректного отображения в веб-интерфейсе. Обработанное изображение может быть сохранено в различных форматах с использованием библиотеки Pillow. Дополнительно рассчитываются статистика по обнаруженным объектам, количество экземпляров каждого класса и средняя уверенность распознавания. Эта информация может быть передана в языковую модель в качестве текстового контекста для мультимодального диалога, позволяя агенту учитывать визуальную информацию при формировании ответов. Предусмотрена возможность отключения модуля YOLO, в случае чего изображение передается в систему без детекции, и языковая модель отвечает исключительно на основе своего предварительного обучения.

Модуль мониторинга и уведомлений обеспечивает сбор данных от трех типов источников: ручного ввода через форму интерфейса (как одиночные показания, так и пакетная загрузка данных в формате JSON), подписки на MQTT-топики, указанные в конфигурации активных источников данных (реализовано через библиотеку paho-mqtt), а также HTTP-эндпоинта /api/monitoring, принимающего POST-запросы с массивом записей в формате JSON. Каждая запись мониторинга содержит временную метку, название метрики, числовое значение, единицу измерения и идентификатор источника. Все поступившие данные сохраняются в таблицу базы данных SQLite. Для последующего анализа и визуализации накопленных показаний используются библиотеки pandas и numpy. При добавлении каждой записи система автоматически проверяет ее соответствие правилам уведомлений, которые могут задавать условия вида >, <, =, >=, <= или «в диапазоне» с одним или двумя пороговыми значениями. В случае срабатывания правила соответствующее событие фиксируется в таблице базы данных SQLite, и уведомление отправляется в Telegram с помощью HTTP-запросов (библиотека requests). Также через requests выполняется интеграция с внешним погодным API (OpenWeatherMap) для получения метеоданных. Пользователь может создавать, редактировать и удалять правила через специальный интерфейс на вкладке «Уведомления», а также просматривать историю срабатываний.

Параллельно с мониторингом функционирует модуль логирования и А/Б тестирования. Все взаимодействия пользователя с системой, включая задаваемые вопросы, полученные ответы (сгенерированные LLM), режим использования RAG, перечень источников информации и оценки релевантности, сохраняются в таблицу базы данных. Для проведения А/Б тестирования, сравнения двух версий ответов с использованием и без использования RAG предусмотрена

отдельная вкладка. Пользователь вводит вопрос, после чего система генерирует два варианта ответа с помощью LLM, с применением RAG и без него. Сравнив ответы, пользователь голосует за лучший вариант, и результат фиксируется в базе данных.

Для объективной оценки эффективности различных режимов работы системы применен Z-тест, позволяющий определить, является наблюдаемая разница в предпочтениях (ответах) пользователей статистически значимой или же она обусловлена случайными факторами. Данный статистический критерий вычисляется по формуле:

$$Z = \frac{\hat{p}_A - \hat{p}_B}{\sqrt{\hat{p}\left(1-\hat{p}\right)\left(\frac{1}{n_A}+\frac{1}{n_B}\right)}}, \quad (2)$$

где $\hat{p}_A$ – доля предпочтений (например, доля ответов, признанных лучшими) для варианта A (с RAG); $\hat{p}_B$ – доля предпочтений для варианта B (без RAG); n_A, n_B – количество оценок (голосов) для каждого варианта; $\hat{p}$ – доля ответов, которые пользователи или эксперты посчитали лучшими во всем эксперименте, без разделения на варианты A и B. Долю можно рассчитать по формуле:

$$\hat{p} = \frac{N_A^{\text{релевантно}} + N_B^{\text{нерелевантно}}}{n_A + n_B}, \quad (3)$$

где $N_A^{\text{релевантно}}$ – количество голосов, в которых вариант A признан релевантным; $N_B^{\text{нерелевантно}}$ – количество голосов, в которых вариант B признан нерелевантным.

При уровне значимости $\alpha = 0{,}05$ значение $|Z| > 1{,}96$ свидетельствует о наличии статистически достоверной разницы между вариантами. Данный критерий применяется при анализе результатов А/Б тестирования, накопленных в базе данных системы, что позволяет объективно оценить эффективность использования механизма дополненной генерации.

Накопленные данные могут быть использованы для анализа эффективности различных режимов работы и для улучшения системы в целом. Возможна загрузки файлов с логами из внешних программ мониторинга (например, результаты мониторинга насаждений с использованием нейросетевых моделей) с форматами .txt, .pdf, .docx, .xlsx, .csv, .html. Содержимое файла автоматически извлекается и передается в языковую модель в качестве контекста (с использованием RAG). Пользователь может задать уточняющий вопрос, а модель – дать рекомендации на основе как загруженного файла, так и имеющейся базы знаний. Эта функция позволяет интегрировать данные из различных источников и получать экспертные заключения по уже собранным материалам.

Веб-интерфейс системы, построенный на библиотеке Gradio, содержит семь основных вкладок. Для обеспечения программного доступа к функциям системы создан REST API на базе FastAPI, сервер запускается под управлением ASGI-сервера uvicorn. Вкладка «AI-агент» включает в себя подразделы для обычного чата с возможностью включения RAG, интерактивного агента, собирающего уточняющие данные, мультимодального диалога с изображениями, свободного чата без тематических ограничений и анализа загруженных логов. Вкладка «Мониторинг» позволяет просматривать последние данные датчиков, строить графики выбранных метрик, добавлять показания и настраивать интеграцию с метеостанциями и погодным API. Вкладка «Анализ изображений» предоставляет инструменты для загрузки фотографий растений, настройки параметров модели YOLO и визуализации результатов распознавания. На вкладке «Уведомления» представлены функции создания правил уведомлений, просмотра истории их срабатывания и настройки Telegram-уведомлений с указанием токена и идентификатора чата. Вкладка «Датасет» отображает статистику собранных диалогов, позволяет экспортировать их в формате JSONL для дообучения LLM и проводить А/Б тестирование. Вкладка «База знаний» предназначена для загрузки документов, просмотра содержимого векторной базы. Вкладка «Администрирование» включает в себя инструменты резервного копирования всех данных (SQLite-базы, папка ChromaDB, датасеты) в ZIP-архив, восстановления из резервной копии, удаления записей мониторинга старше заданного числа дней, а также проверки работоспособности системы, которая оценивает доступность LM Studio, состояние MQTT-клиента, целостность баз данных, наличие настроек API метеостанции и доступность сервера Telegram-бота.

Результаты и их обсуждение

В результате проведенных исследований разработан AI-агент поддержки принятия решений. Для оценки качества работы механизма генерации с дополненным поиском (Retrieval-Augmented Generation, RAG) подготовлено 50 вопросов по садоводству, учитывающих интерпретацию данных мониторинга, прогнозирование фенологических фаз (цветение, завязывание и созревание плодов), оценку фитосанитарного состояния, выявление корреляций между погодными условиями и урожайностью. В базу знаний загружены справочники (.PDF), научные публикации (.DOCX). Вопросы были сформированы совместно с экспертами таким образом, чтобы охватить пять основных направлений (фенология,

защита растений, полив, удобрения, интерпретация данных). Для каждого домена было составлено по 10 вопросов. Экспертное голосование включало в себя рандомизацию порядка предъявления ответов (с RAG и без RAG) для каждого эксперта с использованием генератора случайных чисел, чтобы исключить эффект порядка. Экспертам предлагали три варианта оценки: «Вариант А более релевантен», «Вариант Б более релевантен» или «Равноценны по релевантности». Голоса «Равноценны» были исключены из расчета Z-теста для проверки гипотезы о явном преимуществе одного из подходов по релевантности генерируемых ответов. Примеры запросов и ответов показывают, что ответы, сформированные языковой моделью с использованием найденных в базе знаний фрагментов, содержат конкретные рекомендации с указанием источника, тогда как ответы без привлечения внешних источников являются более общими.

Сравнение двух версий ответов (с использованием RAG и без него), проведенное с участием 10 экспертов в области агрономии, показало, что в 84% случаев эксперты отдали предпочтение ответам, дополненным информацией из базы знаний, в 10% случаев – ответам без RAG; оставшиеся 6% ответов были признаны равноценными. Доверительный интервал (ДИ) 95% для доли предпочтений с учетом кластеризации оценок (эффективный размер выборки $n_{eff} = 154$, вычисленный с поправкой на внутрикластерную корреляцию ICC = 0,25 при 10 экспертах и 50 вопросах) рассчитан по формуле:

$$\text{ДИ} = \hat{p} \pm 1{,}96 \cdot \sqrt{\frac{\hat{p}\left(1-\hat{p}\right)}{n_{eff}}}, \qquad (4)$$

где $\hat{p} = 0{,}84$ – доля предпочтений.

Полученный интервал [78,2%; 89,8%] подтверждает статистическую надежность оценки.

Статистическая значимость полученных различий была подтверждена с помощью Z-теста, примененного к оценкам 10 независимых экспертов ($n_A = n_B = 10$). При доле предпочтений $\hat{p}_A = 0{,}84$ (с RAG) и $\hat{p}_B = 0{,}10$ (без RAG) получено: $|Z| = 3{,}21 > 1{,}96$, $p < 0{,}001$, что указывает на преимущество ответов, дополненных информацией из базы знаний.

Согласованность мнений экспертов оценена с помощью статистического коэффициента каппы Коэна (κ), который применен для оценки межэкспертной согласованности при попарном сравнении двух вариантов ответов (с RAG и без RAG). Коэффициент рассчитан на основе матрицы согласий экспертов по 50 вопросам. Согласно шкале Лэндиса и Коха значение κ = 0,72 соответствует «существенному» уровню согласия. Это подтверждает, что предпочтения экспертов не являются случайными, а отражают объективное преимущество ответов, дополненных информацией из базы знаний. Полученные результаты подтверждают эффективность применения базы знаний для повышения релевантности ответов, генерируемых LLM. Время ответа с использованием поиска увеличивалось в среднем на 0,8 с, что не нарушает интерактивность диалога с LLM (рис. 3).

Для тестирования YOLO использован набор из 100 фотографий яблоневого сада с разметкой классов «apple», «flower», «ovary». Датасет для обучения включал в себя 1500 размеченных изображений, из которых 80% использовалось для обучения, 10% – для валидации, 10% – для тестирования. Разметка выполнялась вручную экспертами с использованием полигонов для точного выделения объектов. Валидационная и тестовая выборки формировались с сохранением пропорций классов. Метрики составили точность (precision) = 0,82, полнота (recall) – 0,76, средняя точность (mAP@0,5) – 0.78. Наиболее часто ошибки возникали при распознавании плодов, цветков и завязей, закрытых листьями, а также в условиях низкой освещенности. Время инференса для изображения 640 × 640 на Intel Core i9-10900X составляло ~0,4 с, на GPU NVIDIA GeForce RTX 2080 Ti – ~0,05 с. В мультимодальном режиме распознавания модели YOLO передавались в LLM, что позволило агроному получать осмысленные ответы: например, о количестве яблок на дереве с учетом возможной скрытости части плодов (рис. 4).

Эффективность системы мониторинга проверялась путем эмуляции работы датчиков, отправляющих значения температуры воздуха через протокол MQTT. Были настроены два правила срабатывания: «температура ниже 2°C» (сигнал о риске заморозков) и «температура выше 30°C» (сигнал о необходимости полива) (рис. 5). Уведомления в Telegram приходили в среднем через 2 с после превышения пороговых значений. Пакетная загрузка JSON через HTTP-эндпоинт показала пропускную способность ~500 записей/с, проверка уведомлений для пакета из 1000 записей заняла 0,2 с.

Результаты дообучения модели методом QLoRA были оценены количественно в ходе дополнительного эксперимента. Пятеро экспертов, не участвовавших в первоначальном А/Б-тестировании RAG, сравнили ответы базовой и дообученной модели на тестовой выборке из 30 вопросов, не входивших в обучающий датасет. В 76% случаев эксперты отдали предпочтение ответам дообученной модели, указав на их большую детализированность и точность формулировок. Согласованность



Рис. 3. Пример работы RAG-механизма в интерфейсе AI-агента

Fig. 3. Example of RAG mechanism operation in the AI agent interface

экспертов (каппа Коэна) составила 0,68, что подтверждает воспроизводимость оценки. Результаты дообучения модели на 400 диалогах с положительными оценками пользователей (то есть диалоги, которые пользователи отметили как релевантные) показали улучшение стиля ответов. Языковая модель стала давать более развернутые рекомендации, чаще ссылаться на источники и избегать общих фраз (рис. 6). Например, на вопрос о борьбе с паршой яблони обученная модель предложила конкретные препараты и схему обработки, тогда как исходная модель давала лишь общую рекомендацию. Однако точность RAG зависит от качества загруженных документов. При наличии противоречивой информации в базе знаний модель может давать неоднозначные ответы.

Выполнена комплексная оценка других компонентов системы. Качество извлечения текста из документов различных форматов проверялось на тестовой выборке из 30 файлов включая PDF-документы, файлы Word, Excel-таблицы и HTML-страницы. Точность извлечения текста из PDF-файлов составила 98%, для форматов DOCX и XLSX – 100%, для HTML – 95% (ошибки возникали ввиду сложных вложенных таблиц). Среднее время обработки файла объемом 10 МБ не превышало 1,2 с.

Протестирована возможность загрузки внешних логов, файлов с результатами мониторинга, отчетами в форматах .txt, .pdf, .docx, .xlsx. На тестовой выборке из 10 файлов, содержащих данные фитосанитарных обследований, система успешно извлекала текст, находила релевантные фрагменты с помощью RAG

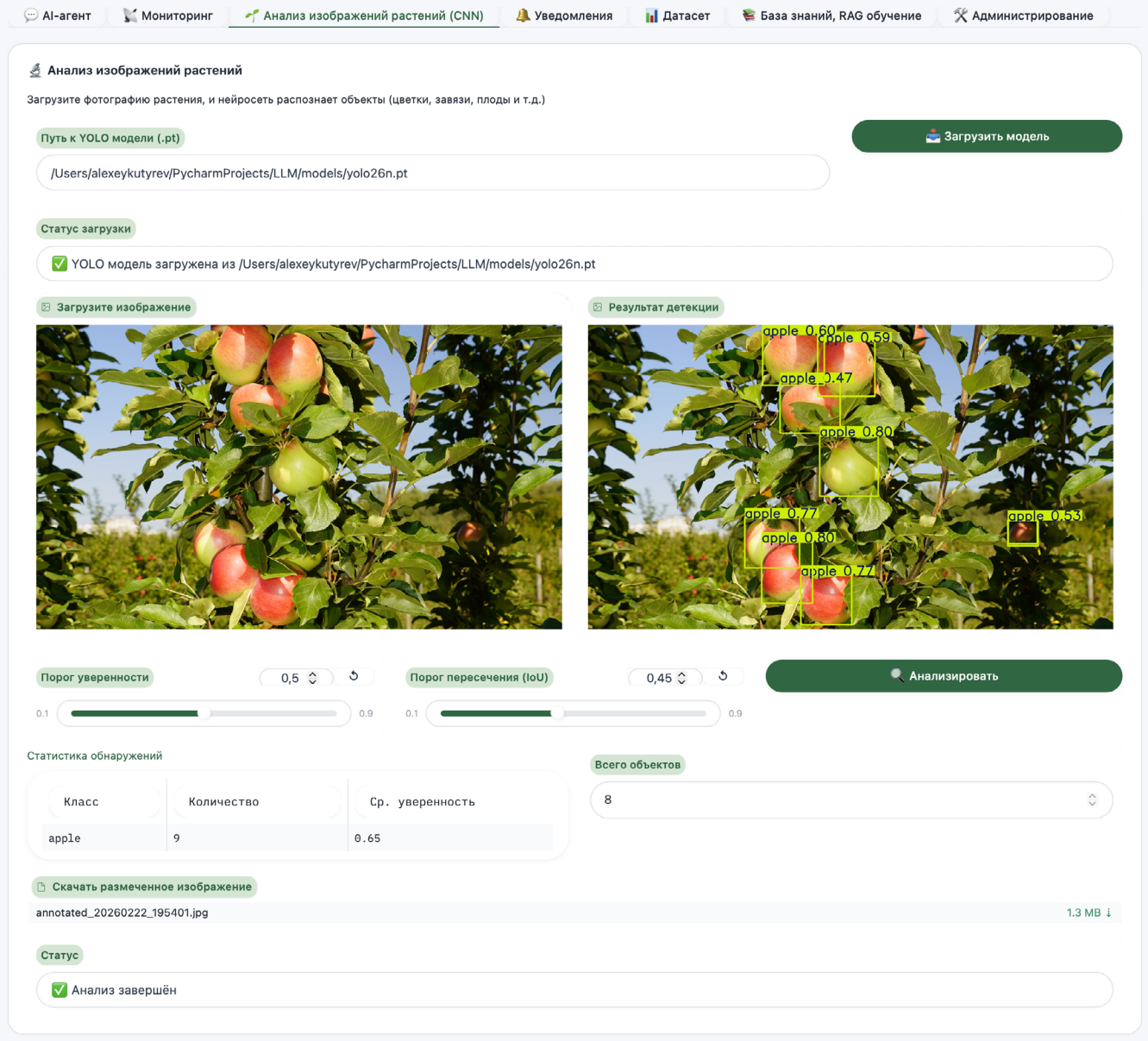


Рис. 4. Распознавание объектов с помощью модели YOLO26n в интерфейсе AI-агента

Fig. 4. Object detection using the YOLO26 model in the AI agent interface

и формировала экспертные заключения со ссылками на исходные данные. Во всех случаях ответы содержали отсылки к конкретным разделам загруженных файлов, что подтверждает эффективность интеграции внешних данных в диалог с AI-агентом.

Функциональность AI-агента, работающего в режиме интерактивного диалога, оценивалась в ходе 20 диалогов, проведенных по типовым сценариям: диагностика болезней яблони, определение оптимальных сроков проведения технологических операций. В 85% случаев AI-агент успешно собирал всю необходимую информацию (культура, фаза развития, наблюдаемые симптомы) и формировал корректную рекомендацию. В остальных 15% случаев требовалось дополнительное уточнение ввиду неоднозначности ответов пользователя, что указывает на необходимость дальнейшего совершенствования сценариев диалога.

Разработанная система не только реализует заявленные функции, но и обладает достаточной производительностью, надежностью и удобством использования для практического применения при решении задач поддержки принятия решений в промышленном садоводстве. Сводные результаты экспериментальной оценки компонентов системы представлены в таблице.

Разработанный AI-агент представляет собой не просто инструмент для обработки запросов, а принципиально новый интерфейс взаимодействия человека с данными о состоянии садовых насаждений и факторах, влияющих на их продуктивность. В отличие от традиционных информационных

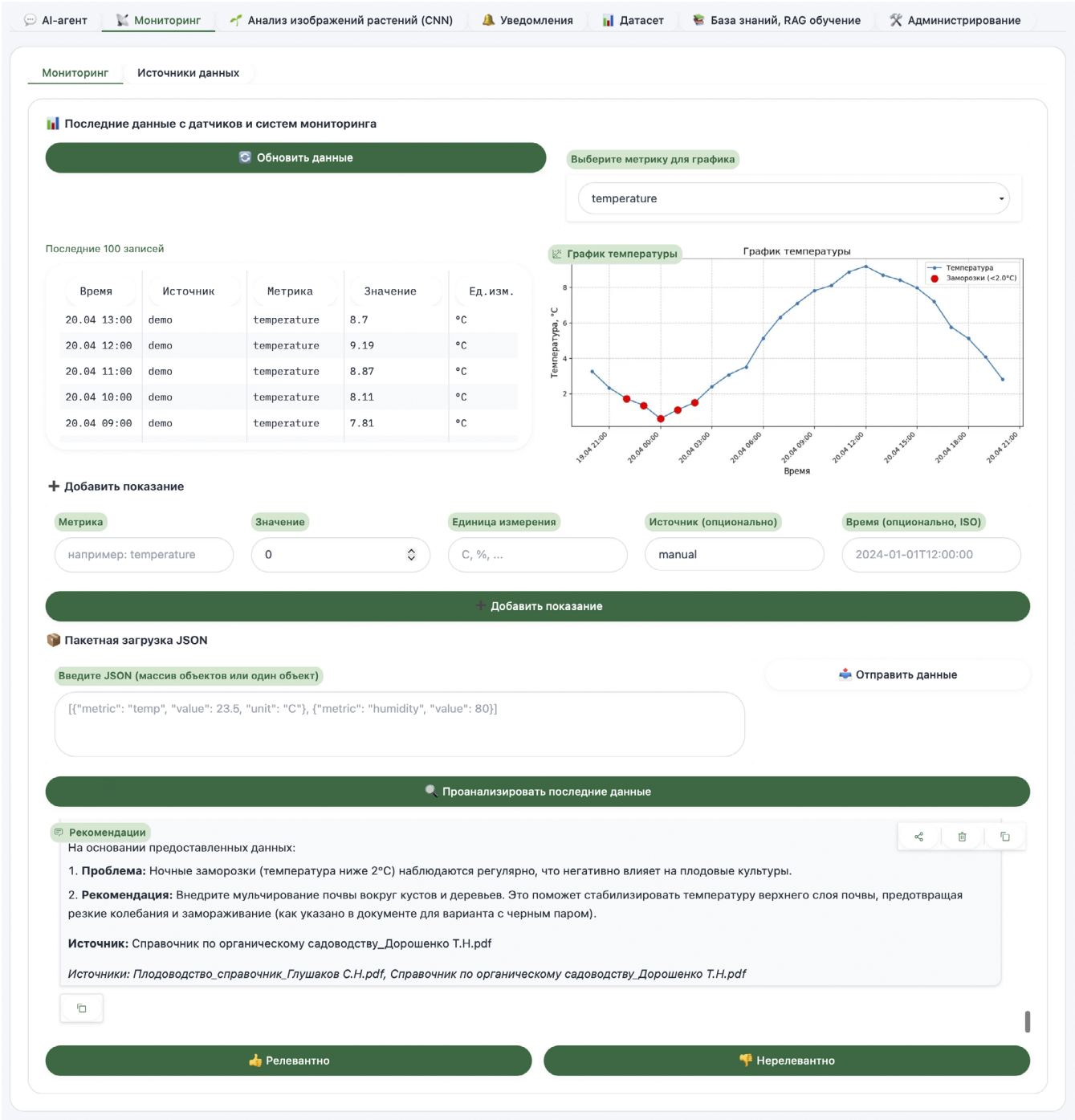


Рис. 5. Результаты мониторинга заморозков в интерфейсе AI-агента

Fig. 5. Results of frost monitoring in the AI agent interface

систем, где пользователь вынужден оперировать дашбордами, таблицами и сложными фильтрами, разработанный AI-агент позволяет перейти к диалоговой аналитике. Пользователь формулирует запрос на естественном языке, а система самостоятельно определяет контекст, использует необходимые данные из базы знаний, результаты мониторинга и компьютерного зрения, генерирует содержательный ответ и при необходимости запрашивает уточнения, продолжая итеративно уточнять решение.

Несмотря на высокие показатели эффективности, разработанный AI-агент имеет ряд следующих ограничений, которые необходимо учитывать при его практическом применении:

1. Как и любая генеративная модель, DeepSeek-R1 может генерировать фактически неверные утверждения (риски галлюцинаций LLM), особенно если запрос выходит за пределы компетенций, описанных в системном промпте, или если в базе знаний отсутствуют релевантные фрагменты. Механизм RAG снижает этот риск, но не исключает его полностью.

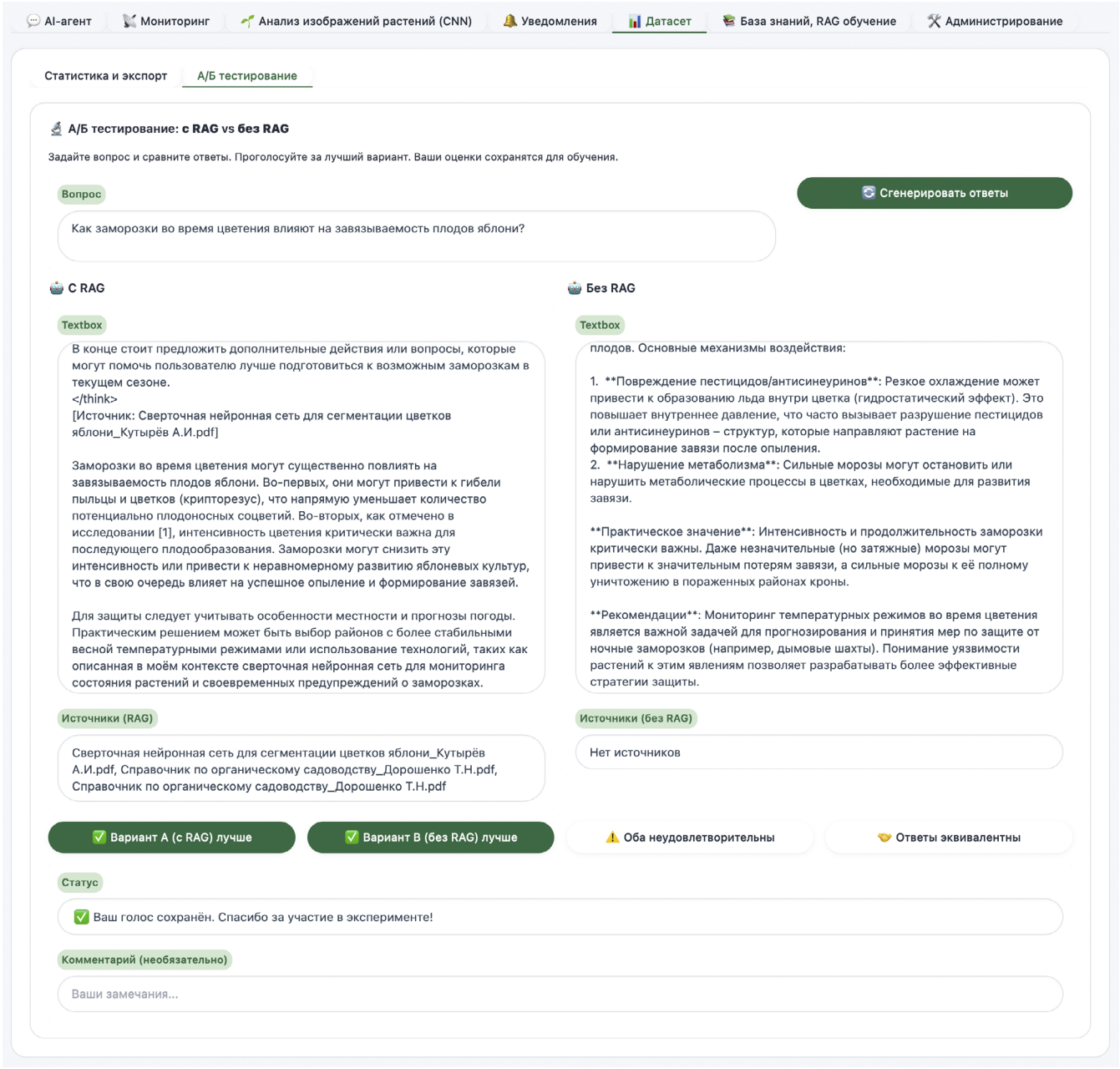


Рис. 6. Результаты А/Б-тестирования в интерфейсе AI-агента

Fig. 6. Results of A/B testing in the AI agent interface

В связи с этим все рекомендации, сгенерированные системой, должны рассматриваться как вспомогательные и требующие верификации агрономом.

2. Эффективность RAG напрямую зависит от полноты, актуальности и непротиворечивости загруженных документов. Наличие устаревших или взаимоисключающих данных в векторной базе может привести к формированию неоднозначных или ошибочных рекомендаций. Система требует регулярного администрирования базы знаний.

3. Работоспособность модуля уведомлений зависит от стабильного доступа в Интернет для отправки сообщений в Telegram и получения данных от внешних API. Для локального запуска LLM с приемлемой скоростью (менее 5 с на ответ) требуется наличие дискретного GPU с объемом видеопамяти не менее 8 ГБ, что может быть экономически ограничивающим фактором для небольших хозяйств.

4. Модуль YOLO показывает снижение точности при окклюзии (перекрытии) объектов листвой в условиях недостаточной освещенности или при нехарактерных ракурсах съемки. Это накладывает ограничения на условия проведения фотофиксации для диагностики.

В дальнейших исследованиях для повышения точности распознавания планируются дообучение модели YOLO на расширенной выборке, интеграция с моделями прогнозирования временных рядов (LSTM)

Таблица

Результаты экспериментальной оценки компонентов системы

Table

Results of experimental evaluation of system components

Компонент	Показатель	Значение
RAG-генерация	Доля предпочтительных ответов (экспертная оценка)	84% (95% ДИ: [78,2%; 89,8%])
	Увеличение времени ответа	+0,8 с
	Согласованность А/Б-тестов (Каппа Коэна)	0,72
YOLO-детекция	Precision (точность)	0,82
	Recall (полнота)	0,76
	mAP@0,5 (средняя точность)	0,78
	Время инференса на центральном процессоре (CPU)	~0,4 с
	Время инференса на графическом процессоре (GPU)	~0,05 с
Мониторинг	Срабатывание правил (заморозки)	4 раза за 1 час
	Задержка Telegram-уведомлений	~2 с
Извлечение текста	Точность (PDF/DOCX/XLSX/HTML)	98% / 100% / 100% / 95%
	Время обработки файла 10 МБ	≤ 1,2 с
Многошаговый агент	Успешных диалогов	85%
Дообучение модели (QLoRA)	Доля предпочтений дообученной модели (экспертная оценка); требуемый объем видеопамяти	76%; ≤ 5 ГБ

для оценки урожайности, разработка мобильного приложения и расширение базы знаний за счет открытых источников данных и публикаций.

Разработанный AI-агент закладывает основу для единой интеллектуальной системы управления садовыми насаждениями, объединяющей процессы сбора, анализа данных, принятия решений и их практической реализации.

Выводы

1. Разработанный программный комплекс «AI-агент поддержки принятия решений» для промышленного садоводства объединяет модули языковой модели (LLM) с механизмом RAG, компьютерного зрения (YOLO), мониторинга данных с датчиков (MQTT, HTTP), А/Б-тестирования и дообучения. Система, построенная на открытых библиотеках Python, предоставляет веб-интерфейс (Gradio) и REST API, что обеспечивает ее модульность, масштабируемость и адаптируемость под задачи промышленного садоводства.

2. Применение механизма генерации с дополненным поиском (RAG) статистически значимо повышает релевантность ответов LLM. В 84% случаев эксперты отдали предпочтение ответам, дополненным информацией из базы знаний (Z-тест: $|Z| = 3{,}21 > 1{,}96$. 95% ДИ: [78,2%; 89,8%], $p < 0{,}001$). Согласованность экспертов по каппе Коэна составила 0,72. Время генерации ответа при использовании RAG увеличивается в среднем на 0,8 с, что не нарушает интерактивности диалога.

3. Экспериментально подтверждена эффективность модуля компьютерного зрения на основе сверточной нейронной сети YOLO26n для детекции объектов яблоневого сада. Точность (метрика Precision) составила 0,82, полнота (метрика Recall) – 0,76, средняя точность (метрика mAP@0,5) – 0,78. Время инференса на GPU NVIDIA GeForce RTX 2080 Ti составляет 0,05 с на кадр, что позволяет использовать систему в режиме реального времени. Интеграция результатов детекции в языковую модель обеспечивает возможность мультимодального диалога о состоянии растений.

4. Разработанный модуль мониторинга обеспечивает прием данных от различных источников (через MQTT, HTTP, ручной ввод) с пропускной способностью до 500 записей/с, автоматическую проверку правил уведомлений и отправку оповещений со средней задержкой 2 с. Точность извлечения текста из загружаемых документов (PDF, DOCX, XLSX, HTML) объемом 10 МБ в течение 1,2 с составляет не менее 95%.

5. Возможность эффективного дообучения языковой модели методом QLoRA на диалогах из логов системы позволяет адаптировать AI-агента под условия конкретного хозяйства. В ходе тестирования в 76% случаев эксперты отдали предпочтение ответам дообученной модели в сравнении с базовой. В 85% тестовых режимов интерактивного диалога агент успешно собирал необходимую информацию и формировал корректные рекомендации.

6. Разработанный AI-агент может быть использован как исследовательский инструмент для проведения научных экспериментов и как основа для создания коммерческих решений в области точного садоводства: автоматизированных рабочих мест агронома, платформ мониторинга растений, сервисов дистанционной диагностики болезней, систем раннего оповещения о неблагоприятных событиях (заморозки, засуха, вредители) и модулей прогнозирования урожайности.

Список источников/ References

1. Soussi A., Zero E., Sacile R. et al. Smart sensors and smart data for precision agriculture: A review. *Sensors.* 2024;24(8):2647. https://doi.org/10.3390/s24082647 PMC11053448

2. Miller T., Mikiciuk G., Durlik I. et al. The IoT and AI in agriculture: The time is now-a systematic review of smart sensing technologies. *Sensors.* 2025;25(12):3583. https://doi.org/10.3390/s25123583 PMC12196926

3. Kalaany C.M.A., Kimaita H.N., Abdelmoneim A.A. et al. The potential of low-cost IoT-enabled agrometeorological stations: A systematic review. *Sensors.* 2025;25(19):6020. https://doi.org/10.3390/s25196020 PMC12526549

4. Awais M., Wang X., Hussain S. et al. Advancing precision agriculture through digital twins and smart farming technologies: A review. *AgriEngineering.* 2025;7(5):137. https://doi.org/10.3390/agriengineering7050137

5. Хорт Д.О., Кутырев А.И., Смирнов И.Г., Воронков И.В. Разработка системы автоматизированного управления агротехнологиями в садоводстве // Сельскохозяйственные машины и технологии. 2021. № 15 (2). С. 61-68. https://doi.org/10.22314/2073-7599-2021-15-2-61-68

Khort D.O., Kutyrev A.I., Smirnov I.G., Voronkov I.V. Development of an automated management system for agricultural technologies in horticulture. *Agricultural Machinery and Technologies.* 2021;15(2):61-68. (In Russ.) https://doi.org/10.22314/2073-7599-2021-15-2-61-68

6. Gherțescu C., Manta A.G., Bădîrcea R.M. Smart agriculture and technological innovation: A bibliometric perspective on digital transformation and sustainability. *Agriculture.* 2025;15(13):1388. https://doi.org/10.3390/agriculture15131388

7. Krishna M.S., Machado P., Otuka R.I. et al. Plant leaf disease detection using deep learning: A multi-dataset approach. *Multidisciplinary Scientific Journal.* 2025;8(1):4. https://doi.org/10.3390/j8010004

8. Кутырёв А.И., Смирнов И.Г., Андриянов Н.А. Сравнительный анализ моделей нейронных сетей для распознавания плодов яблони на кроне дерева // Садоводство и виноградарство. 2023. № 5. С. 56-63. https://doi.org/10.31676/0235-2591-2023-5-56-63

Kutyrev A.I., Smirnov I.G., Andriyanov N.A. Neural network models of apple fruit identification in tree crowns: comparative analysis. *Horticulture and Viticulture.* 2023;(5):56-63. (In Russ.) https://doi.org/10.31676/0235-2591-2023-5-56-63

9. Lv M., Xu Y-X., Miao Y-H., Su W-H. A comprehensive review of deep Learning in computer vision for monitoring apple tree growth and fruit production. *Sensors.* 2025;25(8):2433. https://doi.org/10.3390/s25082433

10. Хорт Д.О., Кутырев А.И., Смирнов И.Г. и др. Разработка алгоритмов системы распознавания ягод земляники садовой при роботизированном сборе // Электротехнологии и электрооборудование в АПК. 2020. Т. 67, № 1. С. 133-141. https://doi.org/10.22314/2658-4859-2020-67-1-133-141

Khort D.O., Kutyrev A.I., Smirnov I.G. et al. Developing algorithms for a berry recognition system used in robotized harvesting of garden strawberry. *Electrical technology and equipment in the Agro-Industrial Complex.* 2020;67(1):133-141. (In Russ.) https://doi.org/10.22314/2658-4859-2020-67-1-133-141

11. Zhou C., Ge X., Chang Y. et al. A multimodal parallel transformer framework for apple disease detection and severity classification with lightweight optimization. *Agronomy.* 2025;15(5):1246. https://doi.org/10.3390/agronomy15051246

12. Kutyrev A., Andriyanov N., Khort D. et al. Adaptive CNN ensemble for apple detection: enabling sustainable monitoring orchard. *AgriEngineering.* 2025;7(11):369. https://doi.org/10.3390/agriengineering7110369

13. Benos L., Marinoudi V., Busato P. et al. Benchmarking large language models in evaluating workforce risk of robotization: Insights from agriculture. *AgriEngineering.* 2025;7(4):102. https://doi.org/10.3390/agriengineering7040102

14. Ravindran D.J.S., Skarga-Bandurova I., Awais M. et al. AgroLLM: Connecting farmers and agricultural practices through large language models for enhanced knowledge transfer and practical application. *AgriEngineering.* 2026;8(1):38. https://doi.org/10.3390/agriengineering8010038

15. Xiong J., Pan L., Liu Y. et al. Enhancing plant protection knowledge with large language models: A fine-tuned question-answering system using LoRA. *Applied Sciences.* 2025;15(7):3850. https://doi.org/10.3390/app15073850

16. Motamedi B., Villányi B. A reliable publish-subscribe mechanism for Internet of Things-enabled smart greenhouses. *Applied Sciences.* 2024;14(15):6407. https://doi.org/10.3390/app14156407

17. Dong H., Qiu X., Fan D. et al. LPDiag: LLM-Enhanced multimodal prototype learning framework for intelligent tomato leaf disease diagnosis. *Agriculture.* 2026;16(4):419. https://doi.org/10.3390/agriculture16040419

18. Islam M.B., Guerrieri A., Gravina R. et al. From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. *Sensors.* 2025;25(22):6903. https://doi.org/10.3390/s25226903

Информация об авторе

Кутырёв Алексей Игоревич, канд. техн. наук, ведущий научный сотрудник, Федеральный научный агроинженерный центр ВИМ; 109428, Российская Федерация, г. Москва, 1-й Институтский проезд, 5; alexeykutyrev@gmail.com; SPIN-код: 9793-5074; https://orcid.org/0000-0001-7643-775X

Author Information

Aleksei I. Kutyrev, CSc (Eng), Lead Research Engineer; Federal Scientific Agroengineering Center VIM; 5, 1st Institutskiy Proezd Str., Moscow, 109428, Russian Federation; alexeykutyrev@gmail.com; https://orcid.org/0000-0001-7643-775X; ResearcherID: I-3699-2018; Scopus Author ID: 57206770758

Статья поступила 02.03.2026, после рецензирования и доработки 23.06.2026; принята к публикации 26.06.2026

Received 02.03.2026; Revised 23.06.2026; Accepted 26.06.2026