

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА КРОССПЛАТФОРМЕННОГО МОБИЛЬНОГО
ПРИЛОЖЕНИЯ ДЛЯ УЧЕТА И КОНТРОЛЯ УХОДА ЗА
КОМНАТНЫМИ РАСТЕНИЯМИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направления 02.03.02 — Фундаментальная информатика и информационные
технологии

факультета КНиИТ

Россахацкого Сергея Алексеевича

Научный руководитель

к. ф.-м. н., доцент

Ю. Н. Кондратова

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2026

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ	5
ЗАКЛЮЧЕНИЕ	9
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	10

ВВЕДЕНИЕ

Актуальность темы обусловлена устойчивым ростом интереса к комнатному растениеводству в условиях высокой урбанизации и интенсификации повседневной жизни. Вместе с тем значительная часть начинающих любителей сталкивается с отсутствием специализированных знаний, необходимых для долгосрочного содержания растений, что приводит к их гибели и разочарованию пользователей. Рынок мобильных приложений, связанных с растениеводством, характеризуется выраженной фрагментарностью: существующие решения, как правило, ориентированы на решение одной. Ни одно из доступных решений не предлагает целостного цикла взаимодействия от идентификации вида до регулярного трекинга ухода. Развитие технологий машинного обучения, компьютерного зрения и кросс-платформенной мобильной разработки открывает возможности для создания интеллектуальных помощников, способных объединить все перечисленные функции в едином решении.

Целью работы является теоретическое обоснование, проектирование и разработка функционального прототипа кросс-платформенного мобильного приложения «FloraMate», выступающего в роли интеллектуального помощника для ухода за комнатными растениями, с реализацией автоматического распознавания видов, трекинга коллекции и системы адаптивных напоминаний.

Для достижения поставленной цели необходимо было решить следующий комплекс **задач**: провести анализ предметной области и рынка существующих мобильных приложений для растениеводства; осуществить сравнительный анализ современных технологий и фреймворков для кросс-платформенной мобильной разработки и выбрать стек технологий; разработать архитектуру мобильного приложения и сопутствующего серверного обеспечения; собрать датасет, обучить свёрточную нейросеть и конвертировать модель в формат TensorFlow Lite; реализовать клиентскую часть приложения на Flutter; разработать серверную часть на языке C# с использованием ASP.NET Core и базы данных PostgreSQL; провести функциональное тестирование и оценить точность распознавания изображений.

Материалы исследования. В работе использованы открытый датасет House Plant Species с платформы Kaggle [1], содержащий 14 790 изображений 47 видов комнатных растений; фреймворк Flutter для кросс-платформенной разработки клиентского приложения; архитектура нейронной сети EfficientNet-B0

с предобучением на ImageNet; формат TensorFlow Lite [2] для развёртывания модели на мобильном устройстве; язык C# с фреймворком ASP.NET Core [3] и СУБД PostgreSQL [4] для серверной части.

Структура работы. Выпускная квалификационная работа состоит из введения, двух глав и заключения. Первая глава — «Теоретические и технологические основы разработки интеллектуальной системы для ухода за растениями» — посвящена анализу предметной области, сравнительному обзору технологических стеков и теоретическому обоснованию методов машинного обучения. Вторая глава — «Разработка и внедрение мобильного приложения FloraMate» — содержит описание практической реализации системы: обучение модели, разработку клиентской и серверной частей, описание пользовательских сценариев и интерфейса приложения.

1 КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические и технологические основы разработки интеллектуальной системы для ухода за растениями» посвящён анализу предметной области, исследованию существующих мобильных приложений для растениеводства, выбору технологий разработки и изучению методов машинного обучения для распознавания растений по изображениям.

Анализ предметной области и рынка приложений. Целевая аудитория разрабатываемого приложения разделена на три сегмента: начинающие пользователи с минимальным опытом, любители, владеющие коллекцией от пяти до двадцати растений, и коллекционеры-эксперты. Основным целевым сегментом являются начинающие пользователи и любители, где сосредоточен наибольший рыночный потенциал.

Анализ существующих решений позволил выделить три категории приложений. Цифровые трекеры-календари (Planta [5], Vera) обеспечивают планирование ухода через систему напоминаний, однако требуют ручной идентификации вида и не адаптируют рекомендации к реальным условиям содержания. Системы распознавания видов (PictureThis [6], PlantNet [7], iNaturalist [8]) решают задачу идентификации по фотографии, но не предлагают интегрированных инструментов ухода. Диагностические системы (Plantix [9]) ориентированы на нужды сельского хозяйства и не подходят для городского растениеводства. Проведённый анализ выявил рыночный разрыв: отсутствие интегрированной платформы, объединяющей автоматическую идентификацию, персональный трекинг и адаптивные рекомендации в рамках единого пользовательского опыта [10].

Выбор технологического стека. По результатам сравнительного анализа фреймворков Flutter, React Native и Kotlin Multiplatform для реализации клиентской части выбран Flutter. Решающими факторами стали: собственный движок рендеринга, обеспечивающий стабильную частоту кадров 60–120 Гц без зависимости от нативных компонентов платформы [11]; технология Stateful Hot Reload, существенно ускоряющая итеративную разработку [12]; развитая экосистема стабильных плагинов [13]; единая кодовая база для iOS и Android. Серверная часть реализована на языке C# с использованием фреймворка ASP.NET Core, что обусловлено высокой производительностью асинхронной модели обработки запросов и строгой типизацией языка. В качестве СУБД выбрана PostgreSQL.

Принципиально важным является локальное выполнение инференса на устройстве без передачи изображений на сервер, что обеспечивает конфиденциальность данных и работу в офлайн-режиме [14, 2].

Методы машинного обучения. Задача распознавания вида растения по фотографии относится к классу многоклассовой классификации изображений. Стандартом для данного класса задач являются сверточные нейронные сети (CNN) [15], извлекающие иерархические визуальные признаки. Применяется метод трансферного обучения на базе весов, предобученных на датасете ImageNet (более 1,2 млн изображений 1000 классов) [16], что радикально сокращает объём требуемых размеченных данных. По результатам сравнительного анализа архитектур MobileNetV3, ResNet50 и EfficientNet выбрана архитектура EfficientNet-B0: она обеспечивает наилучшее соотношение точности и вычислительной сложности, допускает эффективную конвертацию в формат TensorFlow Lite с временем инференса 30–80 мс на современном смартфоне [17].

Второй раздел «Разработка и внедрение мобильного приложения FloraMate» посвящён проектированию, реализации и тестированию программного комплекса для учёта и контроля ухода за комнатными растениями.

Архитектура системы FloraMate. Приложение организовано в виде трёх взаимодействующих уровней: мобильный клиент (Flutter), локальное хранилище (Shared Preferences) и удалённый сервер (C# / ASP.NET Core / PostgreSQL). Ключевая операция — локальный инференс модели EfficientNet-B0 в формате TensorFlow Lite — выполняется непосредственно на устройстве пользователя. Сервер обеспечивает исключительно аутентификацию пользователей и синхронизацию коллекций между устройствами, не участвуя в обработке изображений.

Обучение модели распознавания. В качестве обучающей выборки использован датасет House Plant Species [1], содержащий 14 790 изображений 47 классов комнатных растений. Дисбаланс классов (от 66 изображений для юкки до 547 для монстеры деликатесной) компенсировался взвешиванием функции потерь. Перед обучением выполнялась предобработка изображений: приведение размера к 224×224 пикселей и нормализация значений к диапазону [0, 1]. Датасет разделён на обучающую (80%), валидационную (10%) и тестовую (10%) выборки.

Для повышения обобщающей способности модели применялась аугментация данных [18]: горизонтальное отражение, поворот до 20°, масштабиро-

вание до 20%, изменение контрастности и яркости, сдвиги до 10%. Обучение выполнялось в два этапа. На первом этапе веса базовой модели EfficientNet-V0 были заморожены и обучалась только классификационная голова (25 эпох, скорость обучения 10^{-3}). На втором этапе выполнялась тонкая настройка (fine-tuning) последних 50 слоёв базовой сети (50 эпох, скорость обучения 10^{-5}). Для управления обучением применялись колбэки EarlyStopping, ReduceLROnPlateau и ModelCheckpoint.

Оценка качества модели. Тестирование на выборке из 1 480 изображений показало общую точность классификации (Accuracy) 83,8%, Precision — 84,2%, Recall — 83,5%, F1-score — 83,8% (macro avg) [19]. Дополнительное тестирование на 200 фотографиях в реальных условиях дало точность 85,5%: при хорошем освещении — 93,3%, среднем — 85,0%, слабом — 77,5%, при частичном перекрытии листьев — 82,5%. Размер итоговой модели в формате TensorFlow Lite составил 18 МБ, что позволяет встроить её непосредственно в пакет приложения без загрузки при первом запуске.

Разработка клиентского приложения. Проект Flutter организован по функциональному принципу: каждый модуль выделен в отдельный пакет со своими экранами, виджетами, сервисами и моделями данных. Управление состоянием реализовано посредством пакета Provider [13]. Сервис PlantRecognizer выполняет предобработку изображения и инференс модели TFLite, возвращая класс с максимальной вероятностью и уровень уверенности. Функциональность распознавания реализована в виде последовательности трёх экранов: экран камеры, экран результата распознавания, экран добавления растения в коллекцию. Если уровень уверенности модели не превышает порогового значения, пользователю предлагается повторить съёмку.

Разработка серверной части. База данных PostgreSQL включает четыре таблицы: users (учётные данные), plant_species (справочник видов с интервалами полива), user_plants (коллекции пользователей) и plant_events (история событий ухода). Пароли хранятся в хешированном виде с применением алгоритма bcrypt [3]. REST API [20] включает эндпоинты регистрации и аутентификации с выдачей JWT-токена, получения и пополнения коллекции растений, регистрации событий полива и получения задач по уходу на неделю. Все защищённые эндпоинты требуют передачи JWT-токена в заголовке Authorization.

Пользовательский интерфейс. Интерфейс приложения разработан в со-

ответствии с принципами Material Design [21] и организован в виде трёх основных экранов, доступных через нижнюю навигационную панель. Главный экран отображает персональную коллекцию растений с карточками, включающими фотографию, название и индикатор дней до следующего полива; растения, требующие срочного полива, выделяются анимацией. Дополнительно реализована вкладка энциклопедии со справочными сведениями об уходе за 47 поддерживаемыми видами. Экран календаря позволяет планировать и отслеживать задачи по уходу с привязкой к конкретному растению из коллекции. Экран профиля обеспечивает управление учётной записью, настройку push-уведомлений и выбор темы оформления (светлой или тёмной).

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы были получены следующие основные результаты.

Проведён анализ предметной области и рынка мобильных приложений для растениеводства, выявлена рыночная ниша — отсутствие интегрированной платформы, объединяющей автоматическую идентификацию, персональный трекинг и адаптивные рекомендации в рамках единого пользовательского опыта.

Осуществлён сравнительный анализ технологических стеков; обоснован выбор Flutter для клиентской части, ASP.NET Core и PostgreSQL — для серверной, архитектуры EfficientNet-B0 и формата TensorFlow Lite — для модуля распознавания.

Разработан и реализован локальный модуль распознавания видов комнатных растений на основе архитектуры EfficientNet-B0, обученной на датасете из 14 790 изображений 47 видов с применением двухэтапного трансферного обучения и аугментации данных. Общая точность классификации на тестовой выборке составила 83,8%, точность в реальных условиях съёмки — 85,5%. Инференс выполняется локально за 30–80 мс, что обеспечивает конфиденциальность данных и работу функции распознавания в офлайн-режиме.

Реализована клиентская часть приложения на Flutter с функциями распознавания растений по фотографии, формирования и ведения личной коллекции, планирования ухода через экран календаря, push-уведомлений и встроенной энциклопедии 47 видов.

Разработана серверная часть на C# / ASP.NET Core с базой данных PostgreSQL, обеспечивающая аутентификацию пользователей на основе JWT-токенов с хешированием паролей алгоритмом bcrypt и синхронизацию коллекций между устройствами через REST API.

Приложение успешно прошло функциональное тестирование и готово к эксплуатации на платформах iOS и Android. Таким образом, все поставленные задачи выполнены в полном объёме, цель работы достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 House Plant Species [Электронный ресурс]. — URL: <https://www.kaggle.com/datasets/kaspergregorowicz/house-plant-species> (Дата обращения 13.01.2026). Загл. с экр. Яз. англ.
- 2 TensorFlow [Электронный ресурс]. — URL: <https://www.tensorflow.org/lite/guide> (Дата обращения 11.03.2026). Загл. с экр. Яз. англ.
- 3 ASP.NET Core [Электронный ресурс]. — URL: <https://learn.microsoft.com/en-us/aspnet/core/> (Дата обращения 14.02.2026). Загл. с экр. Яз. англ.
- 4 PostgreSQL [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/> (Дата обращения 14.02.2026). Загл. с экр. Яз. англ.
- 5 Planta [Электронный ресурс]. — URL: <https://getplanta.com> (Дата обращения 15.01.2026). Загл. с экр. Яз. англ.
- 6 PictureThis [Электронный ресурс]. — URL: <https://www.picturethisai.com> (Дата обращения 25.03.2026). Загл. с экр. Яз. англ.
- 7 PlantNet [Электронный ресурс]. — URL: <https://plantnet.org/en/> (Дата обращения 12.04.2026). Загл. с экр. Яз. англ.
- 8 iNaturalist [Электронный ресурс]. — URL: <https://www.inaturalist.org> (Дата обращения 23.02.2026). Загл. с экр. Яз. англ.
- 9 Plantix [Электронный ресурс]. — URL: <https://plantix.net> (Дата обращения 06.03.2026). Загл. с экр. Яз. англ.
- 10 *Петров, И. А.* Обзор архитектурных паттернов и фреймворков для разработки кроссплатформенных мобильных приложений / И. А. Петров, Е. В. Сидоров // *Программная инженерия*. — 2023. — № 4. — С. 12–21.
- 11 *Фролов, С. М.* Оптимизация производительности мобильных приложений на flutter: методы и инструменты / С. М. Фролов // *Информационные технологии и вычислительные системы*. — 2023. — № 3. — С. 78–89.
- 12 *Нидхэм, Д.* Flutter и Dart. Путь от новичка до профессионала / Д. Нидхэм. — Санкт-Петербург: Питер, 2022. — С. 544.
- 13 Flutter. Provider [Электронный ресурс]. — URL: <https://pub.dev/packages/provider> (Дата обращения 11.12.2025). Загл. с экр. Яз. англ.

- 14 *Kowalski, T.* Offline-first architectures for mobile applications: Patterns and best practices // 2022 IEEE International Conference on Software Architecture (ICSA). — 2022. — Pp. 120–130.
- 15 *Шатов, Д. В.* Глубокое обучение и компьютерное зрение на Python / Д. В. Шатов. — Москва: ДМК Пресс, 2021. — С. 892.
- 16 *Sharan, R. V.* Deep Learning with Transfer Learning for Computer Vision: A Practical Guide using Python / R. V. Sharan, G. S. B. Rao. — Apress, 2022. — P. 300.
- 17 *Li, C.* Efficient on-device deep learning: A transfer learning approach for resource-constrained mobile devices // Proceedings of the 21st International Conference on Mobile Systems, Applications, and Services (MobiSys '23). — ACM, 2023. — Pp. 45–57.
- 18 *Иванов, К. П.* Эффективные методы аугментации данных для улучшения обучения спп на небольших наборах данных / К. П. Иванов // *Искусственный интеллект и принятие решений*. — 2024. — № 1. — С. 55–66.
- 19 *Рейдер, Е.* Машинное обучение на Python: Практическое руководство / Е. Рейдер. — Санкт-Петербург: БХВ-Петербург, 2023. — С. 736.
- 20 JSON Web Token [Электронный ресурс]. — URL: <https://jwt.io/introduction> (Дата обращения 13.04.2026). Загл. с экр. Яз. англ.
- 21 Material Design [Электронный ресурс]. — URL: <https://m3.material.io/> (Дата обращения 23.12.2025). Загл. с экр. Яз. англ.