

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА И ВНЕДРЕНИЕ МАСШТАБИРУЕМОЙ
ВЕБ-ПЛАТФОРМЫ ДЛЯ ИДЕНТИФИКАЦИИ ВИДОВ
РАСТЕНИЙ С ИСПОЛЬЗОВАНИЕМ АНСАМБЛИРОВАНИЯ
ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ И МЕТОДОВ
АКТИВНОГО ОБУЧЕНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.04.01 — Информатика и вычислительная техника
факультета КНиИТ
Хворостухиной Натальи Сергеевны

Научный руководитель

должность, к. ф.-м. н.

И. Д. Сагаева

Заведующий кафедрой

доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы обусловлена растущей потребностью в точных, доступных и открытых системах идентификации растений по фотографиям. Существующие коммерческие приложения (PictureThis, PlantNet, PlantSnap) имеют ряд недостатков: ограниченная бесплатная функциональность, отсутствие механизмов обратной связи с пользователем, использование одиночных моделей машинного обучения без ансамблирования. Создание бесплатной веб-платформы, объединяющей гетерогенный ансамбль нейросетей с активным обучением, позволяет преодолеть эти ограничения.

Объект исследования – процесс автоматической классификации изображений растений с привлечением обратной связи пользователей.

Предмет исследования – ансамблевые нейросетевые модели и стратегии активного обучения в задаче идентификации видов цветов.

Цель работы – разработка веб-платформы для идентификации растений на основе ансамбля глубоких нейронных сетей с модулем активного обучения.

Задачи исследования:

1. Провести анализ архитектур свёрточных нейронных сетей и методов ансамблирования.
2. Подготовить и аугментировать набор данных Oxford 102 Flowers.
3. Реализовать гетерогенный ансамбль (VGG19, ResNet50, EfficientNet-B0).
4. Разработать модуль активного обучения с энтропийной стратегией отбора.
5. Создать серверную часть на FastAPI и пользовательский интерфейс.
6. Провести комплексное тестирование платформы.

Материалы исследования: датасет Oxford 102 Flowers (8189 изображений, 102 вида), предобученные модели VGG19, ResNet50, EfficientNet-B0.

Структура работы: введение, четыре главы, заключение, список литературы (22 источника) и приложения. В первой главе рассмотрены теоретические основы. Вторая глава посвящена проектированию архитектуры. Третья глава описывает реализацию ансамбля и активного обучения. Четвёртая глава содержит разработку веб-платформы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе проведён анализ современных методов компьютерного зрения. Рассмотрены трансформерные архитектуры (ViT, Swin Transformer), которые заменяют свёртки механизмом self-attention, позволяя моделировать глобальные зависимости между патчами изображения [1]. Гибридные модели (CvT, LeViT, CoAtNet) интегрируют свёрточные слои в блоки трансформеров, сохраняя способность извлекать локальные признаки [2]. Эффективные архитектуры (MobileViT, EdgeNeXt) сочетают depthwise-свёртки и трансформерные блоки, что делает их пригодными для работы на мобильных устройствах [3].

Описаны методы ансамблирования нейросетей: бэггинг (обучение моделей на разных подвыборках с последующим усреднением) [4], бустинг (последовательное исправление ошибок), стекинг (обучение мета-модели на выходах базовых моделей) [5], а также простое (среднее) голосование, которое использовано в данной работе как наиболее простой и распространённый способ комбинирования предсказаний [6].

Изучены стратегии активного обучения: на основе неопределённости (энтропия Шеннона, маргинальная разница, наименьшая уверенность) [7, 8], на основе комитета (query-by-committee) и ожидаемого изменения модели [9]. Энтропия Шеннона выбрана основной стратегией, так как она учитывает распределение вероятностей по всем 102 классам, что особенно важно для многоклассовой классификации.

Проведён сравнительный анализ существующих приложений для идентификации растений (PictureThis, PlantNet, Google Lens, iNaturalist, PlantSnap, Plantum). Исследования [10, 11] показывают, что даже лучшие приложения не всегда надёжны. Выявлены основные недостатки: ограниченная бесплатная функциональность, отсутствие механизмов обратной связи, использование одиночных моделей.

Во второй главе сформулированы функциональные требования (загрузка изображений, автоматическая классификация, обратная связь, просмотр статистики, сохранение истории) и нефункциональные требования (производительность, надёжность, удобство развёртывания, расширяемость).

Обоснован выбор модульного монолита как архитектурного стиля [12]. Этот подход обеспечивает простоту разработки и развёртывания на началь-

ном этапе, но позволяет в будущем перейти к микросервисной архитектуре благодаря модульной структуре кода. Разработана трёхуровневая архитектура: слой представления (HTML/CSS/JS), слой бизнес-логики (FastAPI + PlantModel), слой данных (SQLite + файловая система).

Выбран технологический стек: Python 3.12 как основной язык, FastAPI как веб-фреймворк (асинхронный, высокая производительность, автоматическая генерация OpenAPI-документации), Uvicorn как ASGI-сервер, PyTorch и torchvision для машинного обучения, Pillow для обработки изображений, SQLite для хранения метаданных.

Спроектирована структура базы данных: таблица identifications с полями id, image, plant_name, confidence, predictions (JSON-сериализованный список топ-5 предсказаний), timestamp, feedback. Такая структура позволяет восстанавливать детальную информацию о предсказании без повторного обращения к модели и собирать обратную связь для активного обучения. Разработаны REST API эндпоинты: GET /, POST /identify, POST /feedback, GET /stats.

В третьей главе описана реализация ансамбля. Использован датасет Oxford 102 Flowers [13], содержащий 8189 изображений 102 видов цветов. Данные разделены на обучающую (6552 изображения), валидационную (818) и тестовую (819) выборки. Для обучающей выборки применены аугментации: случайное кадрирование (RandomResizedCrop(224)), горизонтальное отражение (RandomHorizontalFlip), поворот до 30 градусов (RandomRotation(30)). Для валидационной и тестовой выборок используется фиксированная последовательность: масштабирование до 256 пикселей, центральное кадрирование 224×224 , нормализация с параметрами ImageNet ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$). Загрузка данных реализована через torchvision.datasets.ImageFolder и DataLoader (batch_size=64).

Для создания гетерогенного ансамбля выбраны три архитектуры: VGG19 (2014, 143 млн параметров, глубокая сеть с малыми ядрами 3×3), ResNet50 (2015, 25.6 млн параметров, остаточные связи) и EfficientNet-B0 (2019, 5.3 млн параметров, составное масштабирование) [3,14,15]. Все модели предварительно обучены на ImageNet. Свёрточные слои заморожены, классификаторы заменены под 102 класса: у VGG19 заменён блок classifier на последовательность $25088 \Rightarrow 4096 \Rightarrow 1024 \Rightarrow 102$ с Dropout(0.5), у ResNet50 заменён fc-слой

2048 \Rightarrow 102, у EfficientNet-B0 заменён выходной слой classifier 1280 \Rightarrow 102. Обучение проводилось независимо с оптимизатором Adam (lr=0.001) и функцией потерь CrossEntropyLoss [16].

Реализовано мягкое голосование (soft voting) с равными весами ($w1 = w2 = w3 = 1/3$). Финальное предсказание вычисляется как среднее арифметическое вероятностей трёх моделей [6]. Результаты на 5-й эпохе: VGG19 — 71.1

Разработан модуль ActiveLearning с энтропийной стратегией отбора. Энтропия Шеннона [7] выбрана как наиболее информативная метрика для многоклассовой классификации. Реализованы также альтернативные стратегии: маргинальная разница и выбор по наименьшей уверенности. Проведён экспериментальный цикл из трёх итераций с добавлением 50 образцов на каждой итерации. Результаты показали повышение точности с 86.8

В четвертой главе представлена разработка веб-платформы. Веб-приложение построено на асинхронном фреймворке FastAPI [17]. Структура проекта включает директории MODELS_DIR (модели), UPLOADS_DIR (загруженные изображения), DATA_DIR (SQLite). При инициализации создаются директории, инициализируется БД, загружается ансамбль через класс PlantModel. Загрузка модели включает восстановление архитектур трёх сетей, загрузку весов из контрольной точки plant_ensemble_improved.pth, загрузку словаря cat_to_name.json. Предобработка изображений стандартизирована: масштабирование 256×256 , центральный кроп 224×224 , нормализация.

Реализованы эндпоинты: GET / (HTML-интерфейс), POST /identify (загрузка изображения и классификация с возвратом топ-5 предсказаний и показателя неуверенности), POST /feedback (приём обратной связи), GET /stats (статистика использования).

Реализован на чистом HTML/CSS/JS без внешних фреймворков. Включает зону drag-and-drop, предпросмотр загруженного изображения, кнопку «Определить», блок результатов с топ-5 предсказаниями и цветной шкалой уверенности. Система индикации уверенности (функция getConfidenceMessage) определяет пять уровней: ≥ 90

Данные сохраняются в SQLite (таблица identifications). При успешной идентификации создаётся запись с уникальным id, именем файла, названием

растения, уверенностью, JSON-сериализованным топ-5 и временной меткой. При отправке обратной связи поле `feedback` обновляется. Собранные данные формируют пул кандидатов для дообучения ансамбля.

Проведено функциональное тестирование загрузки изображений, тестирование точности классификации (подтверждена согласованность с результатами главы 3), тестирование целостности БД, оценка времени отклика (1.5–2.5 секунды на CPU).

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР достигнуты следующие результаты:

1. Разработана веб-платформа для идентификации растений на основе ансамбля глубоких нейронных сетей с модулем активного обучения.
2. Ансамбль из трёх моделей (VGG19, ResNet50, EfficientNet-B0) достиг точности 92.9
3. Активное обучение (3 цикла по 50 образцов) повысило точность на тесте с 86.8
4. Создана масштабируемая архитектура: асинхронный FastAPI, замена SQLite на PostgreSQL без изменения кода, возможность контейнеризации, кэширования и выноса модели в отдельный микросервис.
5. Проведено комплексное тестирование, подтвердившее корректность работы всех компонентов.
6. Практическая значимость: платформа может использоваться ботаниками для полевых исследований, садоводами-любителями, токсикологическими центрами для распознавания ядовитых растений.
7. Направления дальнейшего развития: расширение базы видов, инференс на GPU, автоматизация цикла дообучения, разработка мобильного клиента.

Основные источники информации:

1. Dosovitskiy A., Beyer L., Kolesnikov A. и др. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [Электронный ресурс] URL: <https://arxiv.org/abs/2010.11929> (дата обращения: 28.04.2026).
2. Maaz, M. et al. EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications. In: Karlinsky, L., Michaeli, T., Nishino, K. (eds) Computer Vision – ECCV 2022 Workshops // Lecture Notes in Computer Science. 2023. vol 13807. Springer, Cham. https://doi.org/10.1007/978-3-031-25082-8_1
3. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // arXiv:1409.1556. — 2015.
4. Типы машинного обучения: ансамблевые методы и их преимущества: [Электронный ресурс] URL: <https://sky.pro/wiki/analytics/typy-mashinnogo-obucheniya-ansamblevye-metody-i-ih-preimushchestva/> (дата обращения: 20.04.2026).
5. Ансамблевые методы: Градиентный бустинг, случайные леса, бэггинг, метод голосования, стекинг: [Электронный ресурс] URL: <https://scikit-learn.ru/stable/modules/ensemble.html> (дата обращения: 23.04.2026)
6. Sagi O., Rokach L. Ensemble learning: A survey // WIREs Data Mining and Knowledge Discovery. – 2018. – Vol. 8, No. 4. – e1249.
7. Лаптева А.В. Теория информации. — Екатеринбург: УрГЭУ, 2023. — 98 с.
8. Активное обучение: [Электронный ресурс] URL: <https://nec.itmo.ru/wiki/index> (дата обращения: 01.05.2026)
9. Li D. et al. A Survey on Deep Active Learning: Recent Advances and New Frontiers // IEEE Transactions on Neural Networks and Learning Systems. — 2024. — DOI: 10.1109/TNNLS.2024.3396463.
10. McNeill, I., Knoeckel, R., Goggin, A., Wylie, C., & Isoardi, K. Accuracy of plant identification applications to identify plants in suspected poisoning cases referred to the Queensland Poisons Information Centre // Toxicology Communications. 2024. № 8(1). <https://doi.org/10.1080/24734306.2024.2377523>
11. Long K, Townesmith A, Overmiller A, Applequist W, Scalzo A, Buchanan P, Bitter CC. Plant identification applications do not reliably identify toxic and edible plants in the American Midwest // Clin Toxicol (Phila). 2023. № 61(7):524-528. <https://doi.org/10.1080/24734306.2024.2377523>

12. Топалов Н. К. Сравнительный анализ монолитной и микросервисной архитектуры: выбор оптимальной стратегии для программных систем / Н. К. Топалов // Молодой ученый. — 2024. — № 48 (547). — С. 12–15. — URL: <https://moluch.ru/archive/547/119718> (дата обращения: 05.05.2026).
13. Oxford 102 Flower Dataset: [Электронный ресурс] URL: <https://www.kaggle.com/challenge-flower-dataset> (дата обращения: 12.04.2026)
14. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. — Pp. 770–778.
15. Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. — Pp. 6105–6114.
16. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А.А. Слинкина. — Москва : ДМК Пресс, 2017. — 652 с. — ISBN 978-5-97060-554-7.
17. Ramirez S. FastAPI: [Электронный ресурс] URL: <https://fastapi.tiangolo.com/> (дата обращения: 10.04.2026)