

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**Разработка клиент-серверной системы распознавания маршрутных
номеров с использованием нейронных сетей**
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем
факультета компьютерных наук и информационных технологий
Протасова Ивана Леонидовича

Научный руководитель _____ Е.Е. Лапшева

Старший преподаватель

Зав. кафедрой

к.ф.-м.н., доцент _____ М.В. Огнева

ВВЕДЕНИЕ

Актуальность темы. Актуальность темы заключается в необходимости создания систем, облегчающих доступ слабовидящих людей к общественному транспорту. Существующие методы, такие как звуковые оповещения и дисплеи высокой контрастности, не всегда эффективны. Использование методов компьютерного зрения позволяет предложить современное и персонализированное решение. Однако, несмотря на успехи в области обработки изображений, специализированных решений для распознавания маршрутных номеров общественного транспорта в сложных условиях городской среды пока недостаточно.

Цель бакалаврской работы – разработка клиент-серверной системы с интеграцией нейронных сетей, предназначенной для распознавания маршрутного номера общественного транспорта в условиях городской среды.

Поставленная цель определила **следующие задачи:**

- 1) Проанализировать существующие программные решения и обосновать необходимость данной работы.
- 2) Выбрать наиболее подходящие технологии и инструменты для разработки клиент-серверной системы.
- 3) Изучить и описать предметную область проекта.
- 4) Разработать серверную часть проекта.
- 5) Интегрировать модели нейронных сетей в серверный модуль проекта.
- 6) Разработать клиентское Android-приложение, способное обмениваться данными с сервером.
- 7) Провести тестирование и оценку эффективности работы разработанной системы в приближённых к реальным условиям.

Методологические основы разработки клиент-серверной системы распознавания маршрутных номеров представлены в фундаментальных работах в области глубокого обучения и компьютерного зрения: ключевыми являются исследования Гудфеллоу И., Бенджио Й. и Курвилля А. по теории нейронных

сетей [1], а также труды Элгенди М. по применению глубокого обучения в задачах компьютерного зрения [2]. Методы детекции объектов и оптического распознавания текста в естественной среде рассмотрены в работах Бэка Дж. [3], Ляо М. [4] и Ду Й. [5]. Архитектурные подходы к построению микросервисных систем изложены Ричардсоном К. [6], практические методики обучения нейронных сетей — Ховардом Дж. и Гуггером С. [7].

Практическая значимость бакалаврской работы. Разработанная система предоставляет слабовидящим пользователям возможность самостоятельно идентифицировать маршрут приближающегося автобуса средствами обычного Android-смартфона без какого-либо специального инфраструктурного оснащения транспортных средств. Система может быть развёрнута на VPS и использована в реальных условиях городской среды; её микросервисная архитектура позволяет независимо обновлять и масштабировать компоненты, а открытый API — интегрировать систему в сторонние ассистивные приложения.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 15 приложений. Общий объём работы – 94 страницы, из них 46 страниц – основное содержание, включая 8 рисунков, список использованных источников информации – 27 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы разработки клиент-серверной системы распознавания маршрутных номеров» посвящён анализу предметной области, обоснованию технологических решений и теоретическому описанию применяемых методов.

Проблема идентификации маршрутного транспорта слабовидящими пользователями решается существующими подходами лишь частично. Анализ актуальных решений позволяет выделить три категории: инфраструктурные системы, универсальные ассистивные приложения и специализированные системы распознавания номеров. Инфраструктурные системы, в частности «Говорящий город», ограничены зависимостью от оснащения каждой единицы транспорта. Универсальные приложения, такие как Seeing AI и Google Lookout, не оптимизированы под специфику маршрутного табло и несут риски ограничения доступа со стороны зарубежных вендоров. Системы автоматического распознавания номерных знаков проектировались для принципиально иной задачи и не предусматривают интерфейса доступности. Таким образом, выявлена незаполненная ниша: отсутствует мобильное специализированное приложение, способное распознавать маршрутное табло в неконтролируемых условиях городской среды без инфраструктурного оснащения транспортных средств.

Обоснован выбор технологического стека: Kotlin и Jetpack Compose для мобильного клиента, язык Go для сетевого шлюза, Python с FastAPI, YOLO11 и PaddleOCR для ML-сервиса, Docker Compose для управления инфраструктурой.

Изложены теоретические основы применяемых методов. Описаны принципы одностадийной детекции объектов на базе архитектуры YOLO: трёхкомпонентная схема backbone–neck–head, безъякорная парадигма предсказания, алгоритм NMS. Изучены методы оптического распознавания текста в естественной среде: архитектура PaddleOCR на базе Differentiable Binarization и

SVTR. Также рассмотрены подходы к организации серверного инференса моделей глубокого обучения в мобильных системах и обоснован выбор server-side инференса с декомпозицией на шлюз и ML-сервис.

Второй раздел «Практическая реализация клиент-серверной системы распознавания маршрутных номеров» посвящён реализации всех компонентов системы, тестированию и развёртыванию.

Разработана трёхкомпонентная микросервисная архитектура, включающая мобильный клиент на Android, сетевой шлюз на Go и инференс-сервис на Python. Каждый компонент инкапсулирует строго определённую область задач и не осведомлён о внутреннем устройстве соседних, что позволяет независимо обновлять, масштабировать и тестировать компоненты. Определены контракты межсервисного REST API: внешний (POST /upload, GET /health) и внутренний (POST /predict, GET /health).

На рисунке 1 приставлена общая схема взаимодействия компонентов системы:

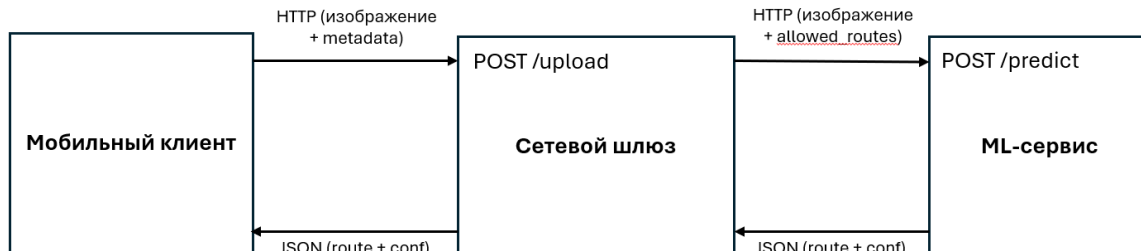


Рисунок 1 — схема взаимодействия компонентов системы

Спроектирован двухстадийный конвейер распознавания. На первой стадии модель YOLO локализует маршрутное табло на изображении. На второй стадии PaddleOCR извлекает текст из вырезанной области, применяя стратегию последовательных попыток с тремя схемами предобработки. Результаты проходят многоступенчатую постобработку: фильтрацию по порогу уверенности, нормализацию, синтаксический анализ регулярным выражением и сопоставление с белым списком эталонных маршрутов посредством взвешенного расстояния Левенштейна с матрицей омоглифов.

Для обучения детектора сформирован и размечен в инструменте CVAT специализированный датасет фотографий автобусов в реальных городских условиях. Дообучение модели YOLO11s в течение 100 эпох достигло значений $mAP@50$ — 0.944, $mAP@50-95$ — 0.637, Precision — 0.900, Recall — 0.920 на валидационной выборке.

На рисунке 2 представлен график кривых обучения:

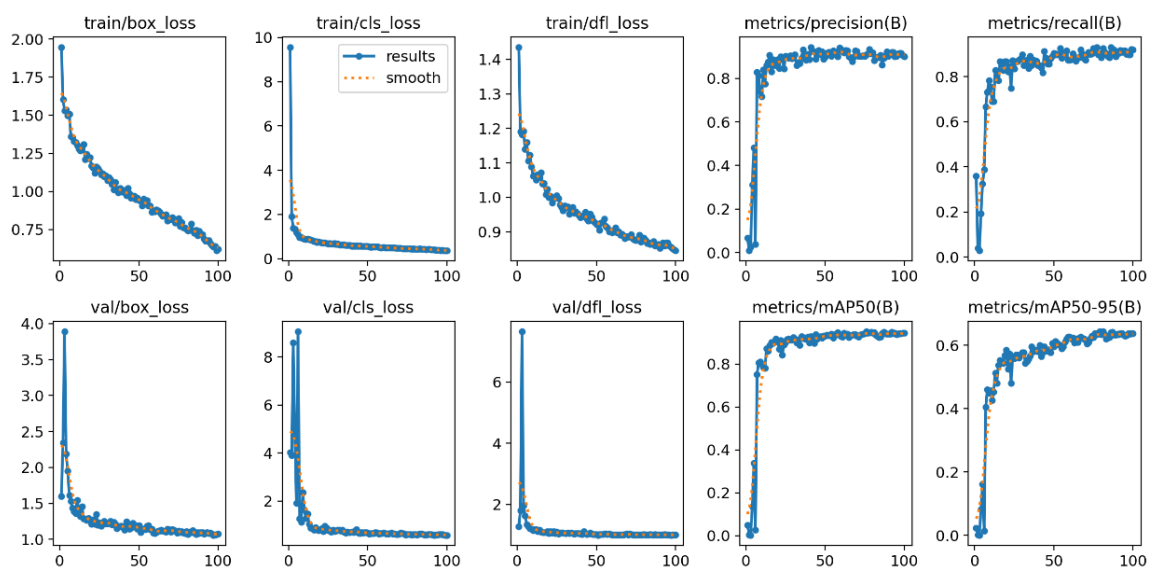


Рисунок 2 — график кривых обучения

Реализован отказоустойчивый сетевой шлюз на языке Go с механизмом Circuit Breaker, семафорным ограничением параллелизма с кодом ответа 429 при заполнении, сквозной трассировкой через X-Request-ID и поддержкой graceful shutdown. Реализован ML-сервис на базе FastAPI с однократной загрузкой моделей через механизм lifespan и параметризуемыми порогами уверенности на уровне API.

Мобильное приложение построено по архитектурному паттерну MVVM. Захват и предобработка изображения реализованы через CameraX с учётом EXIF-метаданных ориентации. Доступность обеспечивается двумя взаимодополняющими каналами: компонентом TtsManager на базе Android Text-to-Speech API и интеграцией с экраным диктором TalkBack через семантический слой Jetpack Compose.

На рисунке 3 представлен интерфейс мобильного приложения:

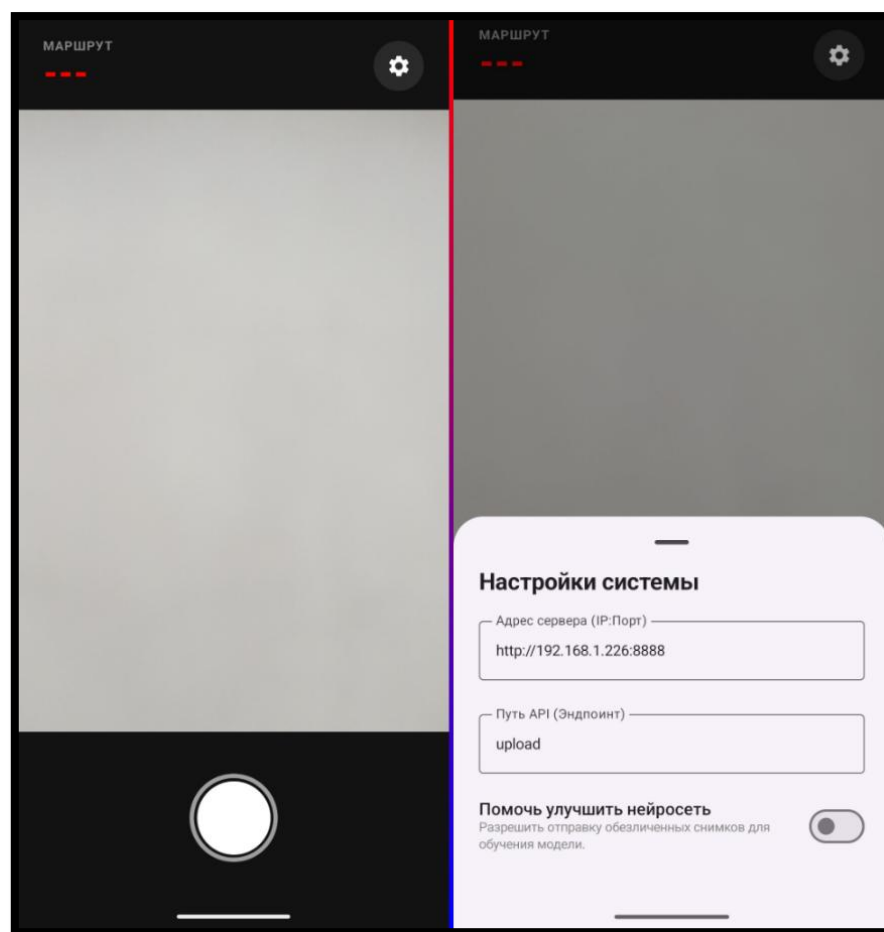


Рисунок 3 — интерфейс мобильного приложения

Тестирование проводилось на двух уровнях. Юнит-тесты сетевого шлюза покрывают HTTP-поведение, поведение Circuit Breaker и логику валидации изображений. Юнит-тесты ML-сервиса покрывают функции постобработки OCR и HTTP-слой FastAPI. Интегральное бенчмаркирование на тестовом датасете из 25 изображений показало точность распознавания 68% при средней задержке инференса 511.2 мс. Все компоненты развёрнуты в изолированных Docker-контейнерах под управлением Docker Compose на VPS с публичным IP-адресом.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы была разработана клиент-серверная система распознавания маршрутных номеров общественного транспорта, ориентированная на слабовидящих пользователей. Все задачи, поставленные во введении, были последовательно решены.

В рамках анализа предметной области проведён обзор существующих решений: инфраструктурных систем, универсальных ассистивных приложений и систем автоматического распознавания номерных знаков. Установлено, что ни один из рассмотренных классов решений не удовлетворяет совокупности требований: мобильности, специализации под задачу идентификации маршрутного табло и работы в неконтролируемых условиях городской среды. Изучены теоретические основы применяемых методов: принципы одностадийной детекции объектов на базе архитектуры YOLO, методы оптического распознавания текста в естественной среде, а также подходы к организации серверного инференса моделей глубокого обучения в мобильных системах.

Обоснован технологический стек всех компонентов системы. Для мобильного клиента выбраны Kotlin, Jetpack Compose и CameraX; для сетевого шлюза — язык Go с его моделью конкурентности на основе горутин; для сервиса машинного обучения — Python с фреймворком FastAPI, детектором YOLO11 и библиотекой PaddleOCR. Управление инфраструктурой реализовано средствами Docker Compose и Poetry.

Спроектирована трёхкомпонентная микросервисная архитектура системы с чётким разделением ответственности между мобильным клиентом, сетевым шлюзом и ML-сервисом. Определены контракты межсервисного REST API, спроектирован двухстадийный конвейер распознавания и архитектура мобильного приложения с поддержкой доступности.

В ходе практической реализации разработан отказоустойчивый сетевой шлюз на языке Go, оснащённый механизмом Circuit Breaker для изоляции недоступного ML-сервиса, семафорным ограничением параллелизма и сквозной трассировкой запросов через идентификатор X-Request-ID. Реализован ML-

сервис на базе FastAPI, инкапсулирующий двухстадийный конвейер детекции и распознавания текста с устойчивой стратегией последовательных попыток предобработки OCR-кroppов. Для обучения детектора собран и размечен специализированный датасет; дообучение модели YOLO11s достигло значений mAP@50 — 0.944 и Recall — 0.920 на валидационной выборке. Разработано мобильное приложение на Kotlin с поддержкой TalkBack и Android TTS, обеспечивающее полноценный доступный пользовательский опыт без необходимости зрительного контакта с экраном. Все компоненты покрыты юнит-тестами, проведено интегральное бенчмаркирование ML-конвейера на реальных изображениях, показавшее точность распознавания 68% при средней задержке инференса 511.2 мс. Система развёрнута в контейнеризированной среде на VPS и доступна для подключения мобильного клиента.

Таким образом, в результате работы создан функционирующий программный комплекс, реализующий полный цикл от захвата изображения мобильным устройством до голосового озвучивания распознанного маршрута пользователю, и решающий задачу повышения транспортной доступности для слабовидящих людей средствами современного компьютерного зрения и мобильной разработки.

Основные источники информации:

1. Goodfellow I., Bengio Y., Courville A. Deep Learning [Электронный ресурс] : книга / MIT Press. — URL: <https://www.deeplearningbook.org> (дата обращения: 31.05.2026).
2. Elgendy M. Deep Learning for Vision Systems. — Manning Publications, 2020. — 512 с.
3. Baek J., Kim G., Lee J. et al. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis [Электронный ресурс] : статья / arXiv. — URL: <https://arxiv.org/abs/1904.01906> (дата обращения: 31.05.2026).
4. Liao M., Wan Z., Yao C. et al. Real-time Scene Text Detection with Differentiable Binarization [Электронный ресурс] : статья / arXiv. — URL: <https://arxiv.org/abs/1911.08947> (дата обращения: 31.05.2026).
5. Du Y., Chen Z., Jia C. et al. SVTR: Scene Text Recognition with a Single Visual Model [Электронный ресурс] : статья / arXiv. — URL: <https://arxiv.org/abs/2205.00159> (дата обращения: 31.05.2026).
6. Richardson C. Microservices Patterns. — Manning Publications, 2018. — 520 с.
7. Howard J., Gugger S. Deep Learning for Coders with fastai and PyTorch [Электронный ресурс] : книга / O'Reilly Media. — URL: <https://www.amazon.com/Deep-Learning-Coders-fastai-PyTorch/dp/1492045527> (дата обращения: 31.05.2026).