



Факультет Компьютерных Наук
Департамент Программной Инженерии

Умная кормушка

Smart Feeder

Выполнили: Студенты группы БПИ233
Хрулев Анатолий Сергеевич
Петров Константин Сергеевич

Руководитель: Кандидат технических наук, доцент департамента программной инженерии
Макаров Сергей Львович

Москва 2025

Проблема и актуальность

Цель

Разработать программный комплекс "Умная кормушка" (Android-клиент + ML-модель), обеспечивающий удаленный мониторинг кормушки для белок, управление доступом к корму на основе распознавания и просмотр медиаконтента.

Задачи

- Анализ аналогов и выбор технологий.
- Разработка Android-приложения (клиент).
- Обучение и оптимизация ML-модели распознавания белок.
- Реализация клиент-серверного взаимодействия.
- Тестирование системы.

Анализ аналогов

GreenBackyard



PerchMe



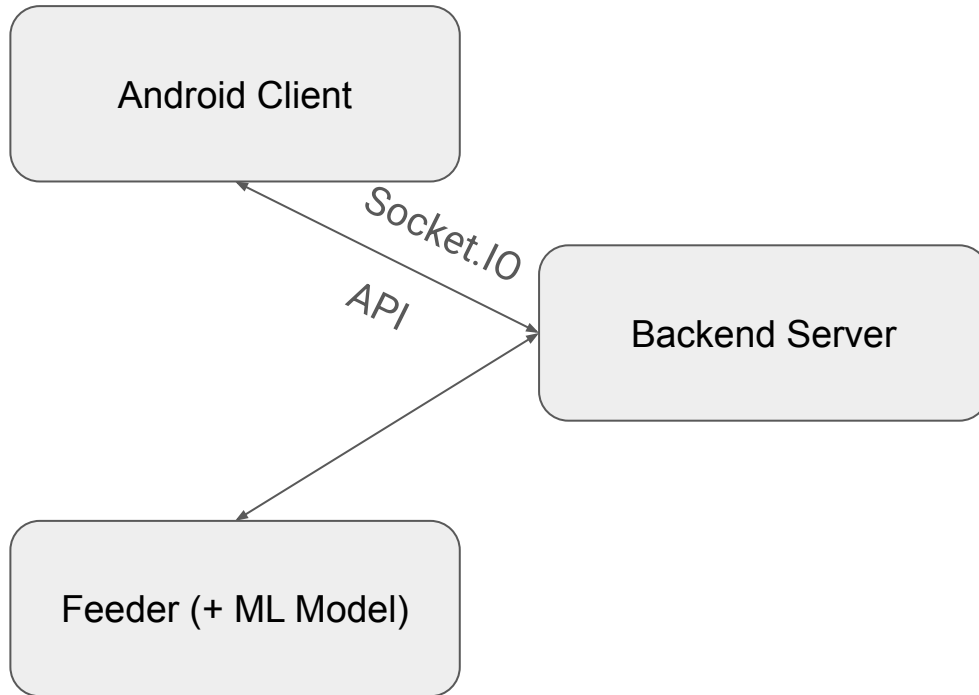
В ходе исследования были выявлены 4 косвенных аналога.

Существующие решения либо не сфокусированы на белках, либо не предоставляют автоматического управления доступом на основе ML, либо являются закрытыми системами.

Smart Squirrel Proof Bird Feeder

Group #10 : Linfei Jing, Christine Li, Yitian Xue

Архитектура системы



Android Client: Управление подключением, просмотр списка видео/кормушек, воспроизведение записей, просмотр live-стрима, скачивание видео, отображение статусов.

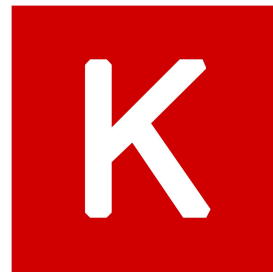
ML Model: Распознавание наличия белки в кадре для управления доступом. Интеграция в аппаратный модуль кормушки.

Ключевые технологии и решения

Android

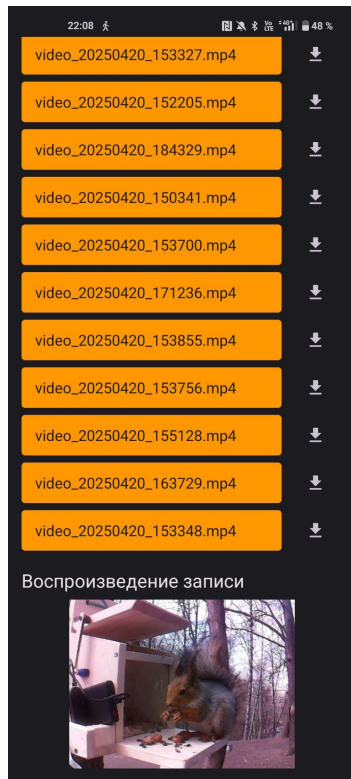
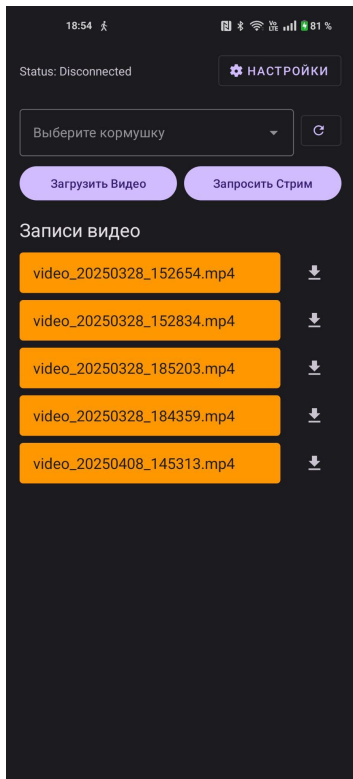
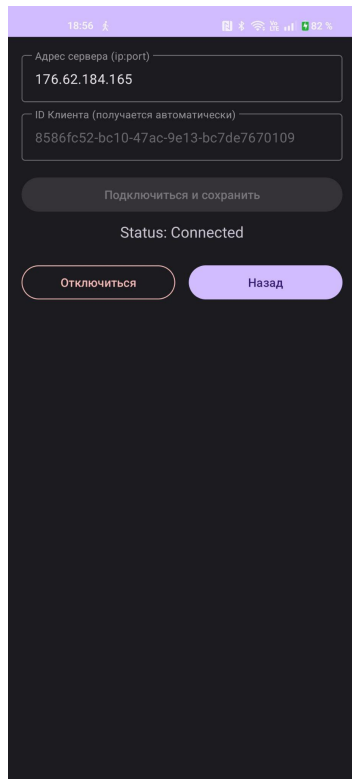


ML



TensorFlow

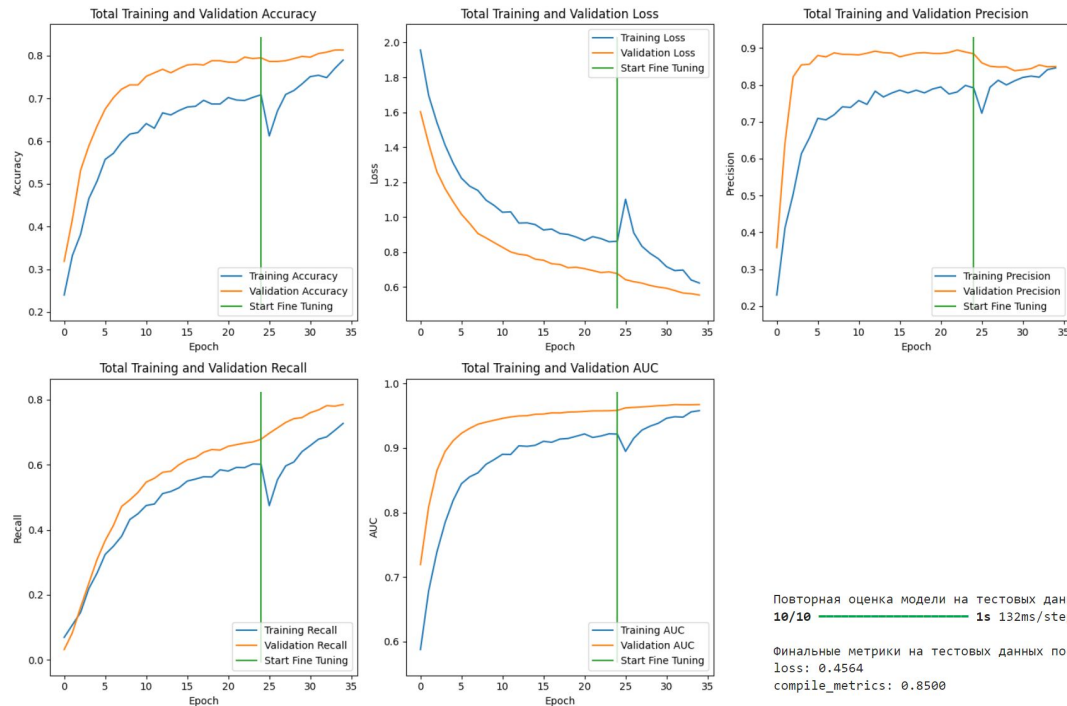
Реализация: Android-приложение



Основной функционал:

- Настройка соединения (адрес сервера, получение ID клиента).
- Загрузка списков (кормушек, видео) через API (Retrofit).
- Воспроизведение записей (ExoPlayer).
- Запрос/остановка/просмотр live-стрима (Socket.IO + ExoPlayer).
- Скачивание видео (DownloadManager).
- Обработка статусов и ошибок (LiveData + Toasts).

Реализация: ML Модель



Основные идеи:

- Модель обучена классифицировать 6 видов белок.
- Подход: Transfer Learning на базе MobileNetV2 (ImageNet).
- Использование Fine-tuning.
- Оптимизация для развертывания: итоговый размер модели стал ~2.4 МБ (был ~9.3 МБ).

Тестирование

Coverage Tests in 'Smart_Feeder_App.app.unitTest' x

Element	Class, %	Metho... ▾	Line, %	Branch, %
com.example.smartfeederapp	28% (8/28)	20% (42/202)	12% (124/1...	9% (44/456)
VideoListHandler	100% (2/2)	100% (10/10)	92% (38/41)	78% (11/14)
SettingsManager	100% (1/1)	100% (8/8)	100% (17/17)	87% (7/8)
VideoItem	100% (1/1)	100% (5/5)	100% (7/7)	100% (0/0)
ApiClient	100% (1/1)	100% (3/3)	85% (23/27)	100% (14/14)
ApiService	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
VideoAdapter	50% (1/2)	44% (4/9)	55% (10/18)	66% (4/6)
ConnectionManager	100% (2/2)	37% (12/32)	12% (29/229)	5% (8/140)
MainActivity	0% (0/8)	0% (0/59)	0% (0/277)	0% (0/113)
StreamPlaybackHandler	0% (0/4)	0% (0/29)	0% (0/145)	0% (0/60)
SettingsActivity	0% (0/3)	0% (0/14)	0% (0/66)	0% (0/19)
VideoPlaybackHandler	0% (0/2)	0% (0/14)	0% (0/79)	0% (0/32)
FullscreenVideoActivity	0% (0/1)	0% (0/11)	0% (0/61)	0% (0/24)
DownloadHandler	0% (0/1)	0% (0/8)	0% (0/66)	0% (0/26)

Unit-тесты

Status 11 passed 11 tests, 45 s 514 ms

Filter tests: [Icons]

Tests	Duration	OnePlus NE2213
Test Results	21s	11/11
ExampleInstrumentedTest	22 ms	1/1
useAppContext	22 ms	✓
MainActivityFlowServerTest	10 s	5/5
A_testNavigateToSettingsAndBack	4 s	✓
B_testMainActivityButtonsDisplayed	1 s	✓
C_testStreamControlsInitiallyHidden	1 s	✓
D_testLoadVideos_populatesRecyclerView	1 s	✓
E_testClickVideoItem_showsPlayer	2 s	✓
MainActivityFlowTest	11 s	5/5
testNavigateToSettingsAndBack	1 s	✓
testMainActivityButtonsDisplayed	1 s	✓
testStreamControlsInitiallyHidden	1 s	✓
testLoadVideosAndClickItem	3 s	✓
testSettingsInteraction	3 s	✓

Инструментальные тесты

Результаты

При разработке проекта выполнены следующие задачи:

- Проведен анализ аналогов.
- Разработано Android-приложение.
- Обучена и оптимизирована ML-модель распознавания белок.
- Реализовано клиент-серверное взаимодействие.
- Система протестирована программно и в реальных условиях.

Направления дальнейшей работы

- Улучшить точность ML-модели.
- Добавить уведомления о появлении белки в кадре доступной кормушки.
- Добавить облачное хранение.
- Изменить UI/UX составляющую приложения.

Демонстрация и доступность

Стабильная версия приложения
доступна для скачивания по QR-
коду из репозитория проекта



Пример работы приложения
доступен по QR-коду на Google
Диске

