



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**



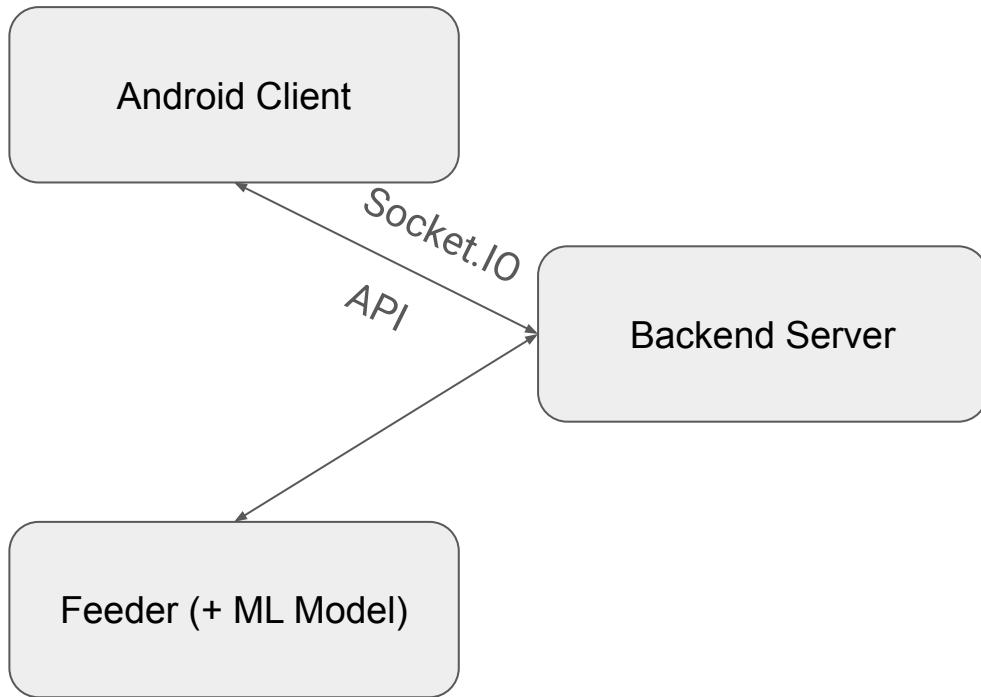
В ходе исследования были выявлены 4 косвенных аналога.

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

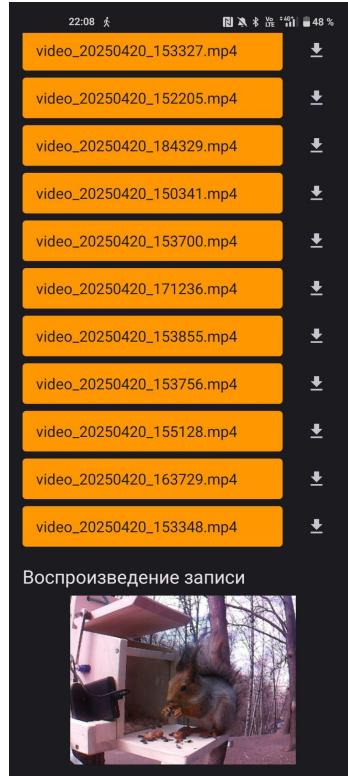
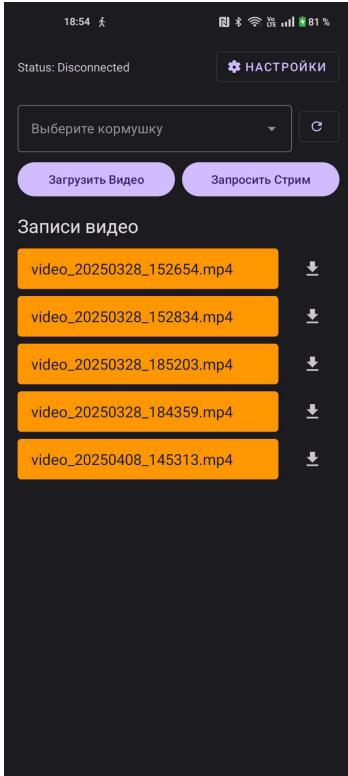
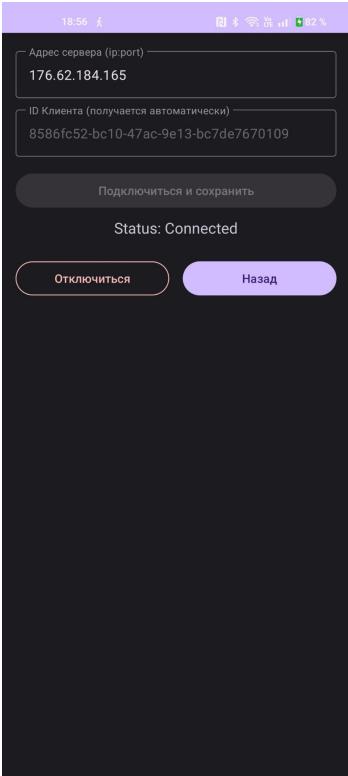
## Android



## ML



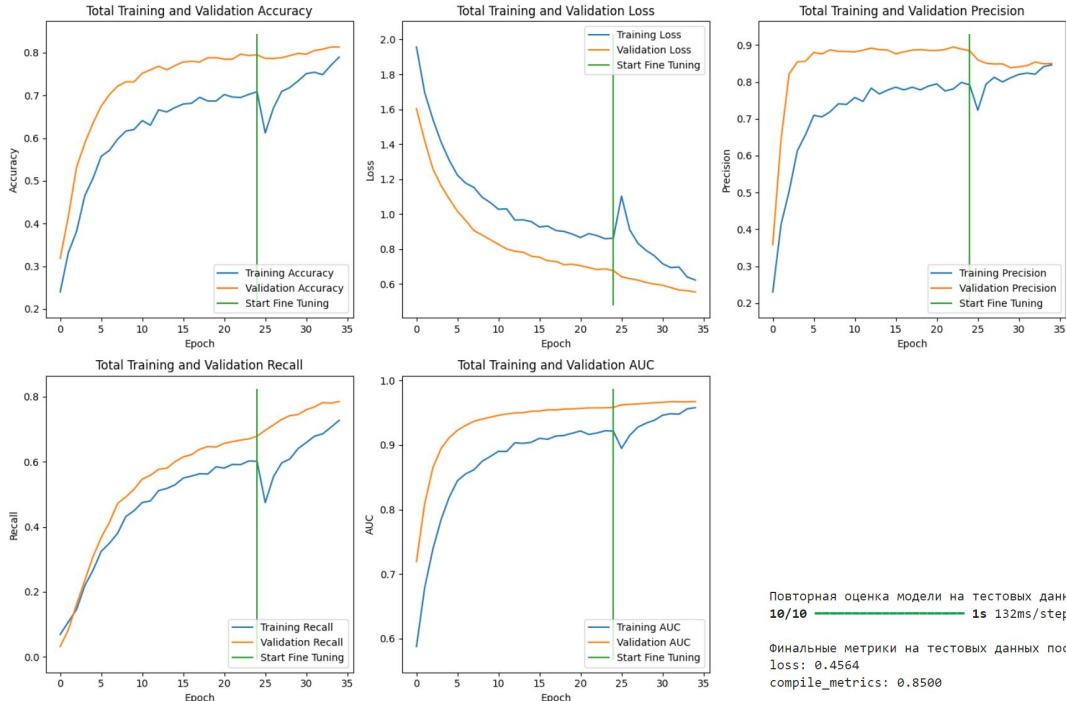
# Реализация: Android-приложение



## 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



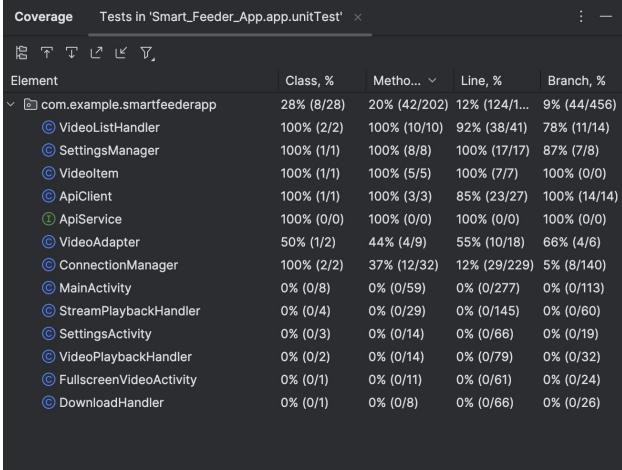
## Основные идеи:

- 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).

Повторная оценка модели на тестовых данных после fine-tuning:  
10/10 1s 132ms/step - accuracy: 0.8043 - auc: 0.6565 - loss: 0.5020 - precision: 0.8491 - recall: 0.7518

Финальные метрики на тестовых данных после fine-tuning:  
loss: 0.4564  
compile\_metrics: 0.8500

# Тестирование



Unit-тесты

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

