

the factor being analyzed). Conclusions were drawn based on the results obtained. The developed predictive models can be used to support prevention and treatment strategies.

Keywords: *data mining, tuberculosis, J48 algorithm, Simple Logistic algorithm, machine learning.*

Об авторах:

АЛЬХАРЕРЕ Тариг Ахмед Яхья – магистрант, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: tarig.alharere6498@mail.ru

АБДУРОЗИК Самар – преподаватель кафедры информационных технологий, Иордано-Суданский колледж науки и технологий, Хартум, Судан. E-mail: samarabdelrazig9@gmail.com

About the authors:

ALHARERE Tarig Ahmed Yahya – Master's Student, Tver State Technical University, Tver. E-mail: tarig.alharere6498@mail.ru

ABDUROSIK Samar – Lecturer at the Department of Information Technology, Jordanian-Sudanese College of Science and Technology, Khartoum, Sudan. E-mail: samarabdelrazig9@gmail.com

УДК 336.226.212.1

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ФОТОГРАФИЯМИ С ЛОКАЛЬНЫМ ХРАНИЛИЩЕМ И МЕЖПЛАТФОРМЕННОЙ СИНХРОНИЗАЦИЕЙ

П.А. Громова, Е.И. Крутских

© Громова П.А., Крутских Е.И., 2026

Аннотация. *Статья посвящена разработке программного комплекса для организации и синхронизации личных фотоархивов между мобильными и десктопными устройствами. Рассмотрена архитектура системы, использующая локальное хранилище данных и не требующая облачной инфраструктуры. Описаны процессы подключения устройств через QR-код, синхронизации медиафайлов, а также автоматической классификации изображений и распознавания лиц с применением моделей машинного обучения. Приведены результаты проектирования и реализации кросс-платформенного решения.*

Ключевые слова: *управление фотографиями, локальное хранилище, синхронизация, машинное обучение, распознавание лиц, классификация*

изображений, кросс-платформенность, микросервисная архитектура, LiteDB, ONNX.

Введение

Современные пользователи сталкиваются с проблемой организации растущих архивов фотографий и видео. Существующие облачные решения требуют постоянной оплаты, могут изменять условия предоставления услуг и не всегда обеспечивают достаточный уровень конфиденциальности [1, 2]. Локальные системы хранения, такие как NextCloud и PhotoPrism, часто нуждаются в сложной настройке, что может вызвать затруднения у неопытных пользователей [3]. Цель данной работы заключается в создании удобного кросс-платформенного приложения, которое обеспечивает синхронизацию медиафайлов между устройствами через локальную сеть без необходимости использования облачных сервисов.

Архитектура системы

Программный комплекс построен по клиент-серверной архитектуре и включает следующие компоненты:

1. Серверную часть. Реализована на платформе ASP.NET Core [4]. Выполняет функции центрального хаба для хранения метаданных и медиафайлов. Генерирует QR-коды для подключения клиентских устройств.

2. Desktopное приложение. Разработано на WPF (Windows Presentation Foundation) [5] с использованием библиотеки Lero для реализации современного пользовательского интерфейса. Построено по шаблону Model-View-ViewModel (MVVM). Интегрировано с серверной частью в единое приложение.

3. Мобильное приложение. Реализовано на Kotlin с использованием фреймворка Jetpack Compose [6] для декларативного UI. Включает сканер QR-кодов для подключения к серверу. Разработка велась в среде Android Studio [6].

4. Базу данных. В качестве системы управления базами данных используется LiteDB [7] – документоориентированная база данных NoSQL, не требующая отдельного сервера и работающая с единым файлом.

5. Модели машинного обучения. Для классификации изображений и распознавания лиц используются предобученные модели YOLO (You Only Look Once) [8] и FaceONNX [9] соответственно.

Обмен данными между компонентами системы осуществляется по протоколу HTTP с использованием типа содержимого form-data для передачи файлов.

Методы и подходы

Ключевой особенностью системы является использование машинного обучения для автоматической организации фотографий. Процесс обработки новой фотографии включает следующие этапы:

1. Классификацию изображения. С помощью модели YOLO фотография анализируется на наличие объектов из предопределенных категорий (например, «животные», «город», «природа», «спорт») [8].

2. Обнаружение лиц. Модель FaceONNX обнаруживает лица на изображении [9].

3. Построение эмбедингов. Для каждого обнаруженного лица вычисляется векторное представление (эмбединг), которое уникально характеризует черты лица.

4. Идентификацию личности. Вычисленные эмбединги сравниваются с существующими в базе данных с использованием метрик схожести. При превышении порогового значения фотография автоматически добавляется в коллекцию соответствующего человека. В противном случае создается новый профиль.

Для подключения мобильного устройства к системе применяется механизм QR-кодов. Сервер генерирует код, содержащий URL для подключения, который сканируется мобильным приложением. Это обеспечивает простоту настройки без необходимости ручного ввода сетевых адресов.

Результаты и обсуждение

Созданный программный комплекс предоставляет пользователю следующий функционал:

1. Подключение устройств. Процесс подключения мобильного устройства к десктопному приложению занимает менее 30 секунд за счет использования QR-кодов.

2. Синхронизацию медиафайлов. Фотографии и видео, загруженные с любого устройства, автоматически становятся доступны на всех подключенных устройствах в локальной сети.

3. Автоматическую организацию. Тематические подборки создаются на основе категорий, определенных моделью YOLO [8].

4. Группировку фотографий по людям с использованием FaceONNX [9].

5. Управление альбомами. Пользователь может создавать, редактировать и удалять альбомы, а также вручную добавлять в них фотографии.

Результаты сравнительного анализа функциональных возможностей разработанной системы с NextCloud и Google Photos приведены ниже:

Функциональность	Разработанная система	NextCloud	Google Photos
Self-hosted	+	+	–
Локальное хранение	+	+	–
Бесплатное использование	+	+	Ограничено
Распознавание лиц	+	Через плагины	+
Автоматическая классификация	+	–	+
Простота настройки	+	Средняя	+
Синхронизация между устройствами	+	+	+

Тестирование системы проводилось на наборе фотографий знаменитостей и изображений различных категорий. Модель YOLO продемонстрировала точность классификации, сопоставимую с заявленными характеристиками для предобученных моделей на датасете СОСО. Модель распознавания лиц успешно идентифицировала личности при различных условиях освещения и ракурсах.

Заключение

Разработанный программный комплекс решает актуальную проблему организации и синхронизации личных фотоархивов, предлагая альтернативу облачным сервисам. Применение локального хранилища данных обеспечивает конфиденциальность информации, а использование машинного обучения для автоматической классификации и распознавания лиц значительно упрощает процесс управления большими коллекциями фотографий.

К перспективным направлениям развития проекта можно отнести улучшение точности ML-моделей, распознавание сцен и действий (например, «прогулка», «праздник»), разработку встроенного редактора фотографий и расширение списка поддерживаемых платформ (например, macOS, iOS).

Библиографический список

1. Google One: Планы хранения. URL: <https://one.google.com/about/plans> (дата обращения: 16.01.2025).
2. iCloud+: Обзор. URL: <https://www.apple.com/ru/icloud/> (дата обращения: 16.01.2025).
3. PhotoPrism: AI-Powered Photos App for the Decentralized Web. URL: <https://www.photoprism.app/> (дата обращения: 16.01.2025).
4. Microsoft. Введение в ASP.NET Core // Документация Microsoft. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-9.0> (дата обращения: 22.05.2025).

5. Microsoft. Windows Presentation Foundation (WPF) // Документация Microsoft. URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/> (дата обращения: 22.05.2025).

6. Android Developers. Jetpack Compose // Документация Android. URL: <https://developer.android.com/jetpack/compose> (дата обращения: 21.04.2025).

7. LiteDB Documentation // LiteDB. 2023. URL: <https://www.litedb.org/docs/> (дата обращения: 17.04.2025).

8. YOLOv12 Documentation // Ultralytics. 2024. URL: <https://docs.ultralytics.com/ru/models/yolo12/> (дата обращения: 17.04.2025).

9. FaceONNX // GitHub. 2024. URL: <https://github.com/FaceONNX/FaceONNX> (дата обращения: 17.04.2025).

DEVELOPMENT OF A PHOTO MANAGEMENT SYSTEM WITH LOCAL STORAGE AND CROSS-PLATFORM SYNCHRONIZATION

P.A. Gromova, E.I. Krutskikh

Abstract. *The article is devoted to the development of a software complex for organizing and synchronizing personal photo archives between mobile and desktop devices. The system architecture is considered, which uses local data storage and does not require cloud infrastructure. The processes of connecting devices via QR code, synchronization of media files, as well as automatic image classification and face recognition using machine learning models are described. The results of designing and implementing a cross-platform solution are presented.*

Keywords: *photo management, local storage, synchronization, machine learning, face recognition, image classification, cross-platform, microservice architecture, LiteDB, ONNX.*

Об авторах:

ГРОМОВА Полина Александровна – магистрант, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: pa.gromova@gmail.com

КРУТСКИХ Елена Игоревна – старший преподаватель кафедры программного обеспечения, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: yelena.korneeva@ya.ru

About the authors:

GROMOVA Polina Aleksandrovna – Undergraduate Student, Tver State Technical University, Tver. E-mail: pa.gromova@gmail.ru

KRUTSKIKH Elena Igorevna – Senior Lecturer of the Department of Software, Tver State Technical University, Tver. E-mail: yelena.korneeva@ya.ru

УДК 336.226.212.1

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ САДОВЫМ УЧАСТКОМ С ИСПОЛЬЗОВАНИЕМ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ И МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Т.В. Латыпов, Е.И. Крутских

© Латыпов Т.В., Крутских Е.И., 2026

***Аннотация.** Статья посвящена разработке программного комплекса для автоматизации процессов садоводства. Рассмотрена архитектура системы, включающая мобильное приложение, серверную часть на основе микросервисов и сервис садового ассистента. Описаны процессы сбора и обработки данных о растениях с применением веб-скреппинга и языковых моделей, а также реализация функций распознавания растений по фотографии упаковки семян. Приведены результаты проектирования и реализации основных модулей системы.*

***Ключевые слова:** садоводство, мобильное приложение, микросервисная архитектура, искусственный интеллект, машинное обучение, распознавание образов, веб-скреппинг, Flutter, ASP.NET Core, FastAPI.*

Введение

Современное садоводство, как любительское, так и профессиональное, связано с необходимостью учета большого количества данных: видов растений, графика работ, планов участка и т. д. Существующие на рынке приложения зачастую характеризуются ограниченным функционалом в бесплатных версиях, низким качеством распознавания, отсутствием локализации или недостаточной автоматизацией процессов [1, 2]. Целью данной работы является создание комплексного мобильного приложения, не имеющего указанных недостатков и предоставляющего пользователю удобный инструментарий для полного цикла управления садовым участком.