

УТВЕРЖДАЮ

Генеральный директор
ООО «Алнисофт»



_____ А.С. Азаров

15 октября 2025 года

Программное обеспечение «Автоматизированное рабочее место врача-эндоскописта, эндоскопический архив “Legendo”»

Описание технической архитектуры программного обеспечения

На 15 листах

Москва
2025 год

Аннотация

Настоящий документ содержит описание технической архитектуры программного обеспечения «Автоматизированное рабочее место врача-эндоскописта, эндоскопический архив “Legendo”» (далее также – «ПО», «Legendo»), предоставленного для проведения экспертной проверки.

Документ содержит сведения о назначении ПО, схему технической архитектуры ПО, описание компонентов (сервисов и модулей) системы и технологий и средств разработки ПО.

Содержание

Перечень сокращений.....	4
1 Общие сведения	8
1.1 Область применения средства автоматизации.....	8
1.2 Краткое описание возможностей средства автоматизации	8
1.3 Характеристика объекта автоматизации	8
1.3.1 Назначение	8
1.3.2 Потенциальные пользователи Решения	8
1.3.3 Основные процессы, подлежащие автоматизации	8
2 Описание технической архитектуры системы.....	10
2.1 Схема технической архитектуры	10
2.2 Описание компонентов (сервисов и подсистем) системы.....	10
2.3 Технологии и средства разработки	13
2.3.1 Внешнее окружение и технологии в части бэк-энда	13
2.3.2 Внешнее окружение и технологии в части фронт-энда	14
2.3.3 Внешнее окружение и технологии в части средств сопряжения с оборудованием и обработки видеопотока Подсистемы сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем ...	14
2.3.4 Внешнее окружение и технологии в части искусственного интеллекта для Подсистемы сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем.....	15

Перечень сокращений

Термин/ сокращение	Определение
API	(англ. Application programming interface) – программный интерфейс приложения – набор инструкций, который позволяет разным приложениям (частям приложения) общаться между собой.
Backend	Серверная часть приложения, отвечающая за логику, обработку данных и управление ими, но скрытая от прямого взаимодействия с пользователем.
Background	Сервис фоновых задач. Микросервис для выполнения асинхронных задач и задач по расписанию.
CPU	(англ. Central Processing Unit) – центральный процессор, Основной вычислительный компонент компьютера
CSS	(англ. Cascading Style Sheets) – каскадные таблицы стилей – язык, который определяет внешний вид веб-страницы, задавая цвета, шрифты, отступы, фоны и другие декоративные элементы
DOCX	Формат файлов Open XML для документов Microsoft Word
Frontend	Клиентская часть приложения. Пользовательский веб-интерфейс, работающий в браузере.
HTTP	(англ. Hypertext Transfer Protocol) – протокол, который определяет правила обмена данными между клиентом (например, браузером) и сервером в интернете, позволяя передавать веб-страницы, изображения, видео и другие файлы
HTTP(s)	(англ. HyperText Transfer Protocol (Secure)) – безопасное расширение протокола HTTP, которое использует шифрование для защиты передаваемых данных между веб-браузером и сайтом
iGPU	(англ. integrated Graphics Processing Unit) – интегрированный графический процессор, встроенный непосредственно в центральный процессор или на материнскую плату компьютера

Термин/ сокращение	Определение
Jetson	Серия комплектов для разработки от NVIDIA, предназначенных для развертывания ML приложений на периферийных устройствах.
JSON	(англ. JavaScript Object Notation) – текстовый формат обмена данными, удобный для чтения человеком и анализа машиной.
NVIDIA	Компания, разработчик графических процессоров и технологий для игр, профессиональной визуализации, машинного обучения и др.
OCR	(англ. Optical Character Recognition) – технология преобразования изображений текста в машиночитаемый текст.
ONNX	(англ. Open Neural Network Exchange) – открытый стандарт формата файлов для представления моделей машинного обучения
OpenVINO	Открытый бесплатный набор инструментов для разработки высокопроизводительных решений компьютерного зрения на основе машинного обучения на различных платформах Intel.
Open XML	Открытый и свободный стандарт формата файлов для офисных документов, таких как текстовые документы, электронные таблицы и презентации
Python	Высокоуровневый язык программирования общего назначения
PyTorch	Открытая библиотека машинного обучения для Python, основанная на фреймворке Torch. Широко используется для компьютерного зрения и обработки естественного языка.
REST API	(англ. Representational State Transfer Application Programming Interface) – архитектурный стиль для построения веб-сервисов, определяющий принципы взаимодействия между клиентом (например, браузером или мобильным приложением) и сервером для обмена данными через протокол HTTP.
RTS	(англ. Real Time Streaming) – потоковое вещание в реальном времени – непрерывная передача данных через интернет, позволяющая пользователям получать контент, такой как

Термин/ сокращение	Определение
	видео, аудио или другие данные, практически мгновенно по мере их создания без ожидания их полной загрузки
RTSP	(англ. Real Time Streaming Protocol) – протокол потокового вещания в реальном времени – сетевой протокол для управления передачей мультимедиа в реальном времени
S3	(англ. Simple Storage Service) – Простой сервис хранения – REST API протокол для облачного объектного хранилища, разработанный
S3-совместимое хранилище	Объектное хранилище, поддерживающее S3 API
TensorRT	Набор средств разработки от NVIDIA для высокопроизводительного вывода (инференса) моделей глубокого обучения
TFT монитор	(англ. Thin-Film Transistor – тонкоплёночный транзистор) – жидкокристаллический дисплей с активной матрицей, в котором каждый пиксель управляется отдельным тонкоплёночным транзистором.
Vision Transformers	Модель глубокого обучения, которая применяет архитектуру Transformer, изначально разработанную для обработки естественного языка, к задачам компьютерного зрения, обрабатывая изображения как последовательности фрагментов.
web-интерфейс	Набор средств (кнопок, форм, меню) на веб-странице, который позволяет пользователям взаимодействовать с сайтом через браузер, без необходимости установки программ.
yaml	Удобочитаемый формат сериализации данных, часто используемый для конфигурационных файлов.
БД	База данных
Медиафайлы	Файлы, содержащие фото- или видеоинформацию.

Термин/ сокращение	Определение
Микросервис	Небольшой независимый программный модуль, который выполняет одну конкретную бизнес-задачу и взаимодействует с другими микросервисами через сеть
ПК	Персональный компьютер
ПО	Программное обеспечение

1 Общие сведения

1.1 Область применения средства автоматизации

Программное обеспечение «Legendo» (далее Система, ПО, Legendo) предназначено для использования в медицинских учреждениях для автоматизации деятельности эндоскопических подразделений.

1.2 Краткое описание возможностей средства автоматизации

Программное обеспечение Legendo представляет собой веб-приложение, работающее в контуре медицинского учреждения для централизованного сбора, хранения, просмотра и документирования эндоскопических фото- и видеоизображений. Legendo обеспечивает:

- Автоматизированный сбор и анализ медицинских фото и видео изображений с эндоскопических видеосистем;
- Получение, централизованное хранение, отображение, выгрузку и аннотирование фото и видео изображений эндоскопических исследований;
- Автоматизированное заполнение протоколов исследований, документирование результатов исследований и формирование статистической отчетности.

1.3 Характеристика объекта автоматизации

1.3.1 Назначение

Программное обеспечение «Legendo» является автоматизированным рабочим местом врача-эндоскописта и предназначено для использования в медицинских учреждениях для автоматизации деятельности эндоскопических подразделений. Система предназначена для централизованного сбора, хранения, просмотра и документирования эндоскопических фото- и видеоизображений.

1.3.2 Потенциальные пользователи Решения

Ключевыми пользователями Системы являются врачи-эндоскописты, выполняющие эндоскопические исследования на цифровом эндоскопическом оборудовании различного уровня, лечащие врачи, назначающие эндоскопические исследования, руководители эндоскопических отделений.

1.3.3 Основные процессы, подлежащие автоматизации

Программное обеспечение «Legendo» является автоматизированным рабочим местом врача-эндоскописта и предназначено для использования в медицинских учреждениях для автоматизации деятельности эндоскопических подразделений в рамках следующих процессов:

1) Проведение эндоскопических исследований (выполняет врач-эндоскопист):

- Фотофиксация эндоскопического исследования,
- Видеозапись исследования,
- Документирование исследования,
- Поиск и просмотр предыдущих записей пациента, сохраненных в системе,
- Хранение фото- и видеозаписей и протоколов исследований,
- Просмотр фото- и видеозаписей исследований,
- Демонстрация трансляции исследования коллегам для получения второго мнения, в образовательных или научных целях.

2) Пересмотр исследований для постановки диагноза и назначения лечения (выполняет лечащий врач):

- Просмотр фото- и видеозаписей исследований,
- Сравнение двух исследований для определения динамики.

3) Управление отделением эндоскопии (выполняет руководитель):

- Просмотр статистики по отделению,
- Выборочный пересмотр исследований в рамках контроля качества исследований,
- Просмотр статистики по использованию эндоскопов.

2 Описание технической архитектуры системы

Система построена с использованием сервис-ориентированной структуры, в основе которой лежит применение микросервисов – слабосвязанных модулей (компонент) ПО для инкапсулирования ожидаемой логики работы. Микросервисная архитектура подразумевает использование широко известных стандартов взаимодействия, таких как HTTP(s), что позволяет включать в систему в качестве вспомогательных модулей готовые компоненты и решения.

2.1 Схема технической архитектуры

Схема технической архитектуры представлена на рис. 1.

2.2 Описание компонентов (сервисов и подсистем) системы

- 1) Пользовательский web-интерфейс (**Frontend**) – клиентская часть приложения, предоставляющего пользователю графический веб-интерфейс с помощью браузера (Яндекс.Браузер, Chrome). Является основным способом взаимодействия пользователей с ПО.
- 2) Сервис проксирования **Nginx** – выполняет роль обратного прокси, принимает запросы от пользователя и перенаправляет их к нужным микросервисам.
- 3) Сервис авторизации **Keycloak** – выполняет аутентификацию и разграничение уровней доступа пользователей. Информация об учетных записях пользователей содержится в сервисной базе данных PostgreSQL (*artincol-keycloak-postgresql*).
- 4) **Prometheus** используется для агрегации метрик микросервиса **Backend**.
- 5) **Loki** используется для агрегации логов микросервиса **Backend**.
- 6) Агент **Node-exporter** – собирает метрики сервера, такие как использование процессора, памяти, дисков и сети.
- 7) **Grafana** – предоставляет доступ к метрикам и логам микросервиса **Backend**.
- 8) Сервис бизнес-логики **Backend** – реализует бизнес-логику системы, предоставляет REST API для **Frontend** части ПО.
- 9) **sts** – сервис генерации миниатюр снимков.
- 10) Сервис фоновых задач **Background** – микросервис выполнения асинхронных задач и задач по расписанию.

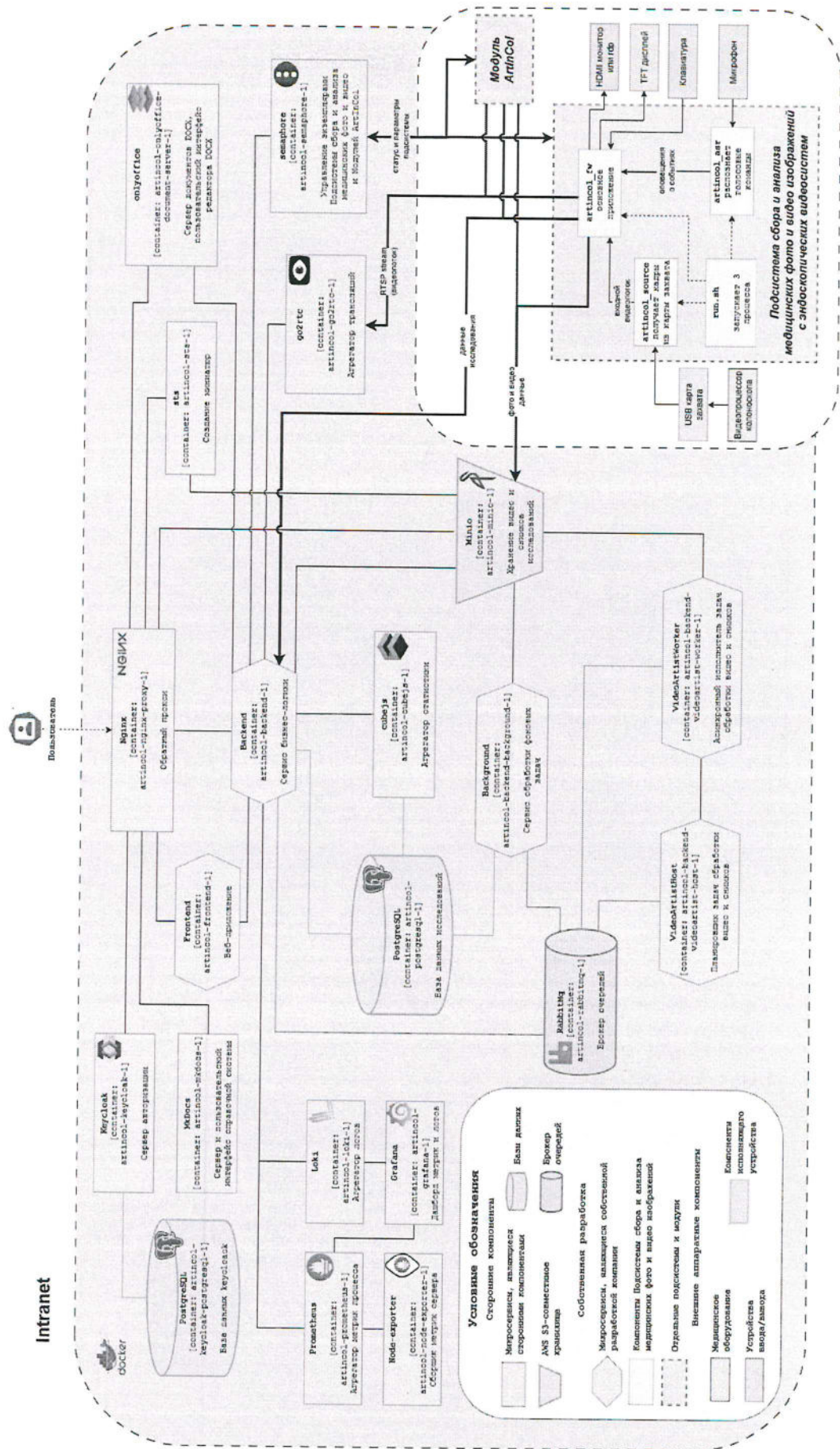


Рисунок 1. Схема технической архитектуры решения

- 11) Микросервис **semaphore** – сервис управления (настройка параметров, обновление) экземплярами Подсистемы сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем и экземплярами модуля ArtInCol.
- 12) Хранилище файлов **Minio** представляет собой S3-совместимое хранилище, в рамках ПО хранит снимки и видео исследований, печатные формы.
- 13) Сервис управления видеопотоком **go2rtc** – микросервис управления RTC потоками видео из Подсистемы сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем и модуля ArtInCol.
- 14) **Onlyoffice** – сервер документов DOCX, также предоставляет веб-интерфейс редактора печатной формы.
- 15) **MkDocs** – сервер справочной системы, содержащий информационные материалы и предоставляющий пользовательский интерфейс для их просмотра.
- 16) Сервис **cubejs** – агрегатор статистики из БД, предоставляет статистическую информацию для отображения в подсистеме статистики.
- 17) **RabbitMq** – брокер очередей.
- 18) **VideoArtistHost** – планировщик асинхронных задач обработки видео и снимков.
- 19) **VideoArtistWorker** – исполнитель задач обработки видео и снимков.
- 20) База данных **PostgreSQL** (*artincol-keycloak-postgresql*) - база данных исследований.
- 21) Подсистема сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем разворачивается промышленным на ПК, подключаемом непосредственно к эндоскопической видеосистеме.

Подсистема автоматически запускается скриптом **run.sh** при включении компьютера. **run.sh** запускает 3 процесса: **artincol_source**, **artincol_fw**, **artincol_asr**.

- Процесс **artincol_source** передает кадры из карты захвата основному процессу **artincol_fw**. Карта захвата принимает видеопоток от видеопроцессора эндоскопа.

- Процесс **artincol_fw** выполняет обработку кадров с помощью OpenVino интерфейса и реагирует на входные события. Модель state (определение состояния исследования «внутри»/«снаружи») работает на iGPU. OCR модель для распознавания текста (PaddleOCR) работает на CPU.

Результатом работы этого процесса являются:

- транслирование rtsp стрима – передача видеопотока в подсистему трансляции,
 - обработка результатов работы state модели,
 - отрисовка индикаторов работы подсистемы и кадров видео на hdmi выходе устройства,
 - отрисовка индикаторов работы подсистемы на TFT дисплее,
 - отправка записанных видео в хранилище,
 - отправка отчета об исследовании на бэкенд,
 - оповещение бэкенда о статусе подсистемы.
- Процесс **artincol_asr** - выполняет прием аудиосигнала и управляет распознаванием голосовых команд. Для детекции слова-команды для включения голосового управления используется модель openWakeWord. Для основных команд голосового управления используется модель Whisper.

22) **Модуль ArtInCol** (ранее зарегистрирован).

2.3 Технологии и средства разработки

Компоненты системы разработаны с использованием языков программирования/ библиотек/ фреймворков, перечисленных ниже.

2.3.1 Внешнее окружение и технологии в части бэк-энда

Стек технологий C#: ASP.NET Core, EntityFramework Core, AutoFixture, Bogus, FFMpegCore, FluentValidation, Flurl, Handlebars.Net, HtmlToOpenXml.dll, Humanizer.Core.ru, MassTransit, MediatR, MiniExcel, Minio, MockQueryable.Moq, Moq, NUnit, Polly, prometheus-net.AspNetCore, RecurrentTasks, Refit, Scrutor, Serilog, Swashbuckle.AspNetCore.

Стек технологий python: faststream, celery, pydantic-settings, minio, aio-pika, psycpg2-binary, pillow, opencv-python-headless, sts-python.

Дополнительные технологии: FFMpeg

Реализация бэк-энд части ПО выполнена на языках программирования C# и python. Для создания HTTP-сервера используется фреймворк ASP.NET Core. Реализация HTTP-запросов и описание REST-API подготовлена с использованием библиотек FluentValidation, Refit и Flurl, Swashbuckle.AspNetCore.

Взаимодействие с базой данных PostgreSQL обеспечивает EntityFramework Core и psycpg2-binary, а библиотека Minio применяется для работы с файловым хранилищем minio.

Библиотека FFMpegCore предоставляет удобный интерфейс к утилите FFMpeg для обработки видеофайлов. Для работы с видеофайлами также используется библиотека opencv-python-headless, а для работы с изображениями – библиотека pillow.

Handlebars.Net является реализацией шаблонизатора текстовых файлов handlebars и используется для генерации печатных форм протоколов. Библиотека HtmlToOpenXml.dll используется для генерации DOCX файлов на основе HTML файлов. Библиотека Humanizer.Core.ru используется для интернационализации текстовых значений в печатной форме. Библиотека MiniExcel используется для генерации Excel файлов.

Библиотеки MassTransit, aio-pika предоставляет взаимодействие с брокером сообщений RabbitMq. Фреймворк faststream используется для создания асинхронного сервиса. RecurrentTasks и фреймворк celery позволяют настроить и запускать периодические задачи в фоне и управлять ими. Библиотека pydantic-settings используется для описания, валидации и чтения конфигураций микросервиса.

Библиотеки MockQueryable.Moq, Moq, NUnit, AutoFixture и Bogus используются для юнит-тестирования, а библиотеки prometheus-net.AspNetCore, Serilog – для работы с логами.

2.3.2 Внешнее окружение и технологии в части фронт-энда

Стек технологий: TypeScript, React, Zustand, Axios, React Query, Keycloak-js, React Hook Form, Vite, Zod, MUI, mkdocs.

Реализация выполнена на языке программирования TypeScript с использованием библиотеки React. Для построения форм применяется библиотека React Hook Form, а для их валидации — Zod. Визуальная часть приложения реализована с помощью CSS-фреймворка MUI. Авторизация в приложении выполнена с помощью библиотеки Keycloak-js. Для хранения состояния приложения используются React Query и Zustand. Взаимодействие с бэкендом посредством HTTP-запросов осуществляется с помощью библиотек Axios и React Query. Для управления зависимостями (установки и отслеживания версий) применяется пакетный менеджер Yarn.

Справочная система реализована с использованием фреймворка mkdocs.

2.3.3 Внешнее окружение и технологии в части средств сопряжения с оборудованием и обработки видеопотока Подсистемы сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем

Стек технологий: C++, OpenCV, gstreamer, jetson-utils, sycl, httpLib, Boost DI, aws-sdk-cpp, licensecc, nlohmann, yaml-cpp, PaddleOCR, onnxruntime, openWakeWord, whisper, whisper.cpp, Visual Studio Code.

Реализация на языке программирования C++. Ядром приложения, обрабатывающим видеопоток является мультимедийный фреймворк

gstreamer. За обработку кадров и отвечает модифицированная jetson-utils на базе sycl и OpenCV.

HTTP клиент и сервер для реализации взаимодействия с бэкендом реализуются на основе библиотеки httpplib. Для интеграции зависимостей используется Boost DI. Для отправки медиафайлов в хранилище minio используется aws-sdk-cpp. Лицензирование реализовано с помощью licensecc. Парсинг json и yaml файлов конфигурации реализуется с использованием библиотек nlohmann и yaml-cpp.

Модель для распознавания текста (PaddleOCR) запускается с помощью кроссплатформенный акселератора onnxruntime, при голосовом распознавании используются модели openWakeWord и whisper. Интеграция голосового управления реализована с использованием whisper.cpp на базе openvino.

2.3.4 Внешнее окружение и технологии в части искусственного интеллекта для Подсистемы сбора и анализа медицинских фото и видео изображений с эндоскопических видеосистем

Стек технологий: Python 3.9, PyTorch, Torchvision, pandas, tqdm, TensorRT, ONNX, pyyaml, imageio, imageio-ffmpeg, shapely, numpy, scikit-image, scikit-learn, filterpy, OpenCV, checksumdir, click, CUDA, PyCUDA, transformers, natsort, psutil, pyarrow, ClearML, jupyter-notebook, poetry, vscode, WSL

Реализация на языке программирования Python с использованием фреймворка pytorch для обучения модели на архитектуре Vision Transformers. Трекинг экспериментов - ClearML. Модель переводится в формат ONNX и затем в формат engine для работы с библиотекой TensorRT на устройствах класса Jetson. Инициализация модели и предсказание по кадру по запросу из приложения, разрабатываемого на c++, происходит через инструменты OpenVINO.