



Общество с ограниченной ответственностью
«Эндоскопическая видеоаналитика Лаб»

**Описание технической
архитектуры программного
обеспечения
Polyptron.AI**

Введение	3
Клиентское приложение	4
Детекция объектов	4
Принятие решений	4
Определения лучшего снимка	4
Визуализация данных	5
Работа с web api	5
Серверное приложение	5
Web server на основе Flask	5
Сервис для выполнения асинхронных задач на основе Celery	6
Брокер сообщений RabbitMQ	6
База данных под управлением СУБД Postgres	6
Графический клиент для СУБД Postgres - PGAdmin4	6

Введение

Система выполнена на клиент-серверной архитектуре. Клиентское приложение в режиме реального времени (не менее 30 кадров в секунду) анализирует видеопоток с эндоскопа, помечает для врача зоны интереса (на выходном изображении), отслеживает найденные патологии на протяжении всей процедуры и отправляет изображения отфильтрованные по качеству на сервер. Серверное приложение сохраняет данные по проведенной процедуре, формирует протокол обследования и отправляет уведомления со ссылкой на протокол обследования по спискам email рассылки.

Общая схема системы изображена на рисунке 1.

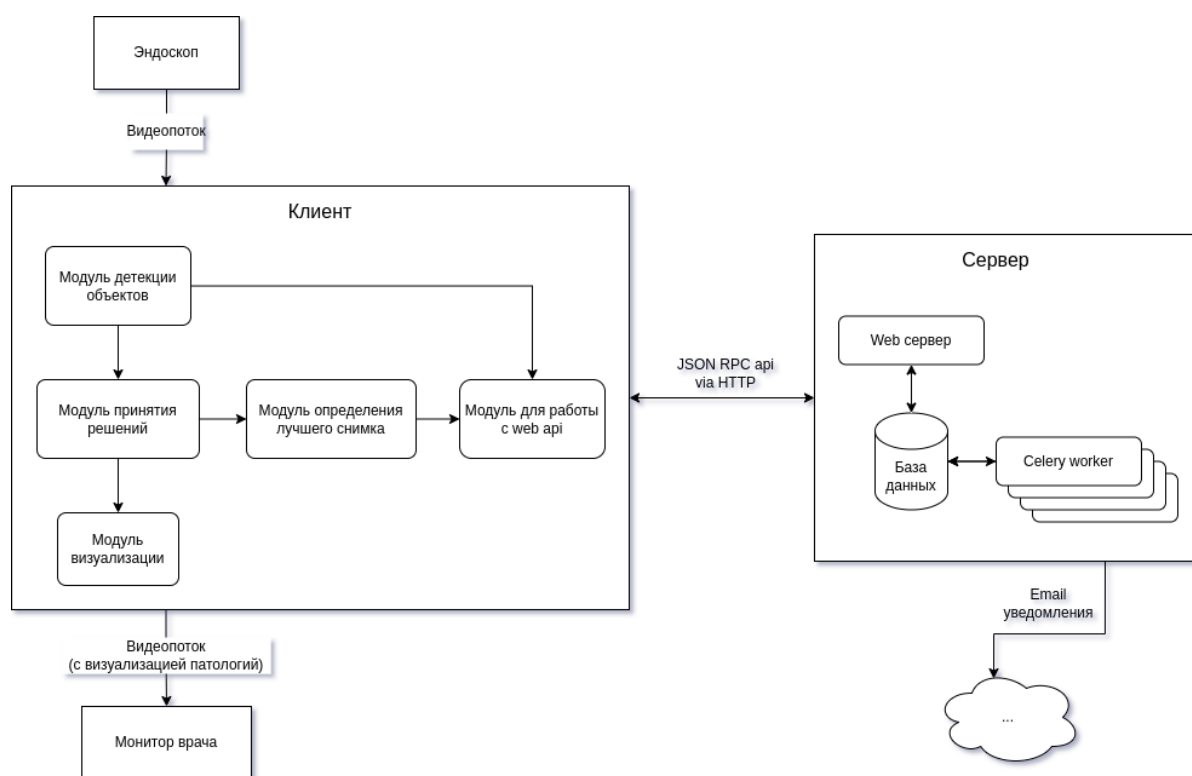


Рисунок 1 - Графическая схема архитектуры Polyptron.AI.

Клиентское приложение

Клиент выполнен в виде приложения на python. Приложение логически разделено на несколько модулей:

- Детекция объектов;
- Принятие решений;
- Определения лучшего снимка;
- Визуализация данных;
- Работа с web api.

Детекция объектов

Модуль содержит в себе нейронную сеть, реализованную в библиотеке PyTorch, средства предобработки изображения и постобработки выявленных объектов и принимает на вход изображения из видеопотока. Перед подачей изображения в нейронную сеть оно проходит предобработку инструментами opencv и torchvision: изменение размера на пригодное для подачи в нейросеть, нормализацию по цветовым каналам RGB. Затем подготовленное изображение подается в нейронную сеть, которая обнаруживает, локализует, классифицирует и возвращает обнаруженные на изображении патологии.

Принятие решений

Модуль принимает на вход информацию о выделенных объектах из модуля детекции объектов, хранит историю обнаруженных объектов на прошлых кадрах и с помощью инструментов библиотеки opencv осуществляет отслеживание объектов во времени, формирует треки.

Определения лучшего снимка

Модуль получает на вход исходный кадр и информацию о патологиях на нем из модуля принятия решений. С помощью инструментов из библиотек torchvision и opencv определяется качество кадра (отсутствие размытия, размер патологии на кадре). Если объект перестал отслеживаться в модуле принятия решений, то при прохождении определенного порога качества кадра информация об патологии передается в модуль сохранения данных на сервере.

Визуализация данных

Модуль получает на вход исходный кадр и информацию об обнаруженных на нем патологиях из модуля детекции объектов, отвечает за отрисовывание на изображении контуров и регионов интереса обнаруженных объектов с помощью библиотеки `opencv`.

Работа с web api

Модуль реализует коммуникацию с сервером посредством вызовов api по протоколу JSON RPC (`http(s)` в качестве транспорта). Задача данного модуля предоставлять остальной программе возможность сохранять и читать данные с удалённого сервера.

Серверное приложение

Сервер состоит из набора сервисов, каждый из которых изолирован внутри своего контейнера. В качестве инструмента контейнеризации используется `docker`. Для управления контейнерами используется `docker compose`.

Набор сервисов:

- Web server на основе Flask;
- Сервис для выполнения асинхронных задач на основе Celery;
- Брокер сообщений RabbitMQ;
- База данных под управлением СУБД Postgres;
- Графический клиент для СУБД Postgres - PGAdmin4.

Web server на основе Flask

Web Сервер реализует api для коммуникации с клиентским приложением. Через точку доступа, которую предоставляет сервер, происходит сохранение и чтение данных клиентом.

Сервис для выполнения асинхронных задач на основе Celery

Сервис реализует ту часть бизнес-логики, которую невозможно уместить в рамки http запроса, т.е. длительные по времени операции. Например формирование и рендеринг протокола обследования, или рассылку email уведомлений.

Брокер сообщений RabbitMQ

Брокер сообщений предназначен для обмена данными между сервисами.

База данных под управлением СУБД Postgres

База данных под управлением СУБД Postgres используется как основное хранилище данных системы.

Графический клиент для СУБД Postgres - PGAdmin4

Графический клиент PGAdmin4 используется для администрирования основной базы данных.