

Программный продукт «Helmetnet»

Описание функциональных характеристик



Оглавление

Описание программного продукта «Helmetnet»	3
Основные функциональные характеристики	3

Описание программного продукта «Helmetnet»

Программный продукт «Helmetnet» представляет собой серверное приложение (микросервис) для автоматического обнаружения средств индивидуальной защиты — защитных касок — на цифровых изображениях с использованием технологий компьютерного зрения и нейронных сетей. Продукт принимает кадры от внешних информационных систем заказчика (систем видеонаблюдения, интеграционных платформ) через REST API, выполняет нейросетевое детектирование людей и защитных касок и возвращает структурированный результат в формате JSON. Продукт применяется для автоматизированного контроля соблюдения требований охраны труда и промышленной безопасности на производственных объектах.

Основные функциональные характеристики

1. Приём и первичная обработка входных данных

- a. Приём POST-запросов, содержащих цифровые изображения (кадры) в кодировке base64, через программный интерфейс REST API (эндпоинт /cvpredict).
- b. Приём изображений файлом (multipart/form-data, поле frame) через альтернативный эндпоинт POST /; при включённом режиме трекинга (IS_TRACKER) — трекинг объектов между кадрами с привязкой к идентификатору камеры (параметр cam_id).
- c. Автоматическое декодирование и предварительная обработка изображений:
 - o изменение размера (resize) под входной формат нейросетевой модели;
 - o нормализация и формирование вычислительных тензоров.

2. Нейросетевое детектирование средств индивидуальной защиты

- a. Обнаружение людей в кадре и определение наличия/отсутствия защитной каски с использованием нейросетевой модели архитектуры YOLO.
- b. Выполнение инференса нейросетевой модели на выделенном сервере инференса (Triton Inference Server) с обменом данными по высокоскоростному протоколу gRPC.
- c. Алгоритм подавления немаксимумов (NMS) интегрирован непосредственно в граф модели, что обеспечивает перенос вычислительной нагрузки на сервер инференса.
- d. Возможность выполнения вычислений как на CPU, так и на GPU.

3. Формирование результата

- a. Возврат структурированного ответа в формате JSON, содержащего перечень обнаруженных объектов, где для каждого объекта указаны:
 - o класс объекта (например, «person», «helmet»);
 - o вероятность (уверенность модели);
 - o координаты ограничивающей рамки [xmin, ymin, xmax, ymax].

4. Интеграция со смежными системами

- a. Взаимодействие с внешними информационными системами заказчика через REST API для передачи результатов детектирования другим модулям.
- b. Поддержка работы в режиме, близком к реальному времени, за счёт асинхронной обработки запросов.

5. Сервисные функции и мониторинг

- a. Программный интерфейс проверки работоспособности сервиса (эндпоинт /health — liveness/readiness probe).
- b. Получение служебной информации о состоянии сервиса: версия, статус подключения к серверу инференса, время работы (эндпоинт /status).
- c. Журналирование обработки запросов с настраиваемым уровнем детализации.

6. Лицензирование и защита

- a. Ограничение доступа к вычислительным функциям на основе срочной (time-based) лицензии; при истечении срока действия лицензии сервис возвращает HTTP-ошибку об отказе в доступе.
- b. API-методы проверки статуса лицензии (/license/status) и её активации с помощью лицензионного ключа (/license/activate).
- c. Встроенная веб-страница управления лицензией (/license/ui).
- d. Защита исходного кода: код поставляется в обфусцированном и зашифрованном виде.

7. Архитектурные особенности

- a. Поставка и развёртывание в виде изолированного Docker-образа в инфраструктуре заказчика.
- b. Модульная архитектура: бизнес-логика (API-шлюз) отделена от нейросетевого вычислительного контура.
- c. Конфигурирование параметров работы через переменные окружения без пересборки образа.