

ООО «Виар Про»

392000, г Тамбов, Советская ул, д. 194и, помещ. 2 офис 47 б

ИНН: 6829164996

ОГРН 1226800005288

**ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ВИДЕОМОНИТОРИНГА И
КОНТРОЛЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ**

Документация, содержащая информацию, необходимую для эксплуатации экземпляра
программного обеспечения

2025

Содержание

1. Наименование изделия	3
2. Назначение и область применения.	4
3. Контактные данные производителя	5
4. Описание изделия и функциональные характеристики	6
5. Риски применения, противопоказания и побочные эффекты	7
6. Развертывание Системы	8
7. Указания по эксплуатации Системы.....	10
8. Техническое обслуживание.....	14
9. Требования безопасности и меры предосторожности	15
10. Словарь терминов	16

1. Наименование изделия

Интеллектуальная система видеомониторинга и контроля производственных процессов (далее Система) представляет собой монолитное программное обеспечение, реализованное на языках Python 3.11.8 для серверной логики, обработки видеопотоков, машинного обучения и работы с базой данных, а также JavaScript для клиентской части, обеспечивающей взаимодействие с веб-интерфейсом и отображение данных в реальном времени.

Компонент обработки видеопотока «Видеопроцессор» осуществляет чтение видеоданных из локальных MP4-файлов, веб-камер или RTSP-потоков, выполняет обнаружение объектов с использованием модели YOLO, отслеживание с помощью DeepSort и извлечение ключевых точек поз посредством Mediapipe Pose.

Компонент классификации состояний «Классификатор» анализирует изображения и ключевые точки с применением дообученной нейронной сети MobileNetV2, определяя состояния объектов, такие как ходьба, сортировка или нажатие кнопки.

Компонент управления базой данных «Менеджер базы данных» организует хранение информации о состояниях и видеофайлах в SQLite, поддерживая их добавление, редактирование и удаление.

Компонент веб-интерфейса «Веб-интерфейс», построенный на Flask, Flask-SocketIO и Bootstrap, отображает аннотированный видеопоток, логи событий с временными метками, графики частоты состояний и предоставляет инструменты для управления данными и дообучения модели.

Компонент обучения модели «Компонент обучения» отвечает за дообучение нейронной сети на пользовательских видеоданных с использованием двухэтапного подхода, включающего обучение новых слоев и тонкую настройку.

2. Назначение и область применения.

Назначение: Система предназначена для автоматизированного видеомониторинга и классификации состояний объектов (людей) на производственных площадках. Она выполняет задачи обнаружения объектов, отслеживания их движений, анализа поз с использованием компьютерного зрения и машинного обучения, а также управления данными состояний и их визуализации.

Область применения: Система используется в производственных средах для:

1. Мониторинга и контроля действий работников на производстве.
2. Обеспечения безопасности и оптимизации производственных процессов.
3. Обучения и анализа данных для улучшения производственных процедур.
4. Поддержки принятия решений на основе реального времени.

3. Контактные данные производителя

Для рекламаций и вопросов в сфере обращения Системы:

ООО «Виар Про»;

Юридический адрес: Россия, 392000, Тамбовская область, г.о. город Тамбов, г. Тамбов, ул. Советская, д. 194и, помещ. 2, офис 47 Б;

e-mail: info@vr-pro.ru

Интеллектуальная система видеомониторинга и контроля производственных процессов является интеллектуальной собственностью ООО «Виар Про», что подтверждается Свидетельством о государственной регистрации программы ЭВМ, выданными Федеральной службой по интеллектуальной собственности РФ: № 2025618458 от 03.04.2025 «Интеллектуальная система видеомониторинга и контроля производственных процессов».

Любое копирование и распространение Системы без разрешения правообладателя является незаконным.

4. Описание изделия и функциональные характеристики

Интеллектуальная система видеомониторинга и контроля производственных процессов представляет собой монолитное программное обеспечение, предназначенное для анализа видеопотоков в реальном времени с целью обнаружения, отслеживания и классификации состояний объектов на производственных площадках (мониторинг действий работников, контроль безопасности, оптимизация процессов). Система реализована на языках Python 3.11.8 для серверной логики, обработки видео, машинного обучения и работы с базой данных, а также JavaScript для клиентской части, обеспечивающей динамическое взаимодействие с веб-интерфейсом.

Технические требования к компьютерному оборудованию и программному обеспечению для эксплуатации системы включают характеристики персонального компьютера: процессор с количеством ядер не менее 6, графический ускоритель с объемом видеопамати не менее 4 ГБ, объем оперативной памяти не менее 16 ГБ, объем свободной памяти твердотельного накопителя не менее 20 ГБ, поддержка беспроводных соединений через Wi-Fi и наличие постоянного IP-адреса. Характеристики Wi-Fi роутера предусматривают наличие не менее одного LAN-порта и скорость передачи данных по локальной сети не менее 100 Мбит/с. Требования к общесистемному программному обеспечению включают Python 3.11.8. Для корректного функционирования необходимы сторонние библиотеки TensorFlow, Ultralytics YOLO, Mediarpipe, OpenCV, Flask, Flask-SocketIO, Pandas и NumPy. Система функционирует на операционных системах Windows 10 и новее, Ubuntu 18 и новее, Debian 11 и новее, Astra Linux 1.8 и новее, обеспечивая стабильную работу при стандартных условиях эксплуатации персональных компьютеров без потерь данных при указанной скорости сети.

Система обеспечивает:

- реализацию удобного веб-интерфейса для управления видеопотоками и состояниями, универсального для различных источников видео (локальные MP4-файлы, веб-камеры, RTSP-потоки);
- классификацию состояний объектов с использованием нейронной сети MobileNetV2 и ключевых точек поз, извлеченных через Mediarpipe;
- интеграцию с видеочамерами для повышения точности анализа; централизованное хранение данных в базе SQLite с возможностью управления через веб-интерфейс.

5. Риски применения, противопоказания и побочные эффекты

Риски применения интеллектуальной системы видеомониторинга и контроля производственных процессов при надлежащем использовании в соответствии с инструкцией отсутствуют.

Для предотвращения ошибок и сбоев в работе системы необходимо строго следовать инструкциям по установке и развертыванию программного обеспечения.

Риски, связанные с использованием стороннего подключаемого оборудования, такого как видеокамеры или сетевые устройства, лежат на производителе оборудования и пользователе, его эксплуатирующем.

6. Развертывание Системы

Экземпляр ПО поставляется в виде развертывание через Docker.

Установите Docker и Docker Compose (если не установлены) и поместите в корень проекта следующие файлы:

- Dockerfile;
- docker-compose.yml;
- requirements.txt;
- код проекта.

Выполните команды:

```
docker-compose build
```

```
docker-compose up -d
```

Система будет доступна по адресу:

```
http://localhost:9000
```

Для развертывания экземпляра ПО в ручном режиме необходимо получить от ООО «Виар Про» набор файлов дистрибутива системы. Данный способ подходит для развертывания на физических серверах, в облаке или локально без использования контейнеризации.

Создайте рабочий каталог проекта, например: /opt/vidmon и поместите в него все файлы дистрибутива. Убедитесь, что в системе установлен Python версии 3.11, а также инструменты pip и virtualenv.

Выполните команды:

```
cd /opt/vidmon
```

```
python3.11 -m venv venv
```

```
source venv/bin/activate
```

После активации окружения установите зависимости:

```
pip install --upgrade pip
```

```
pip install -r requirements.txt
```

Разместите тестовое видео в папке test_data/, и при необходимости – загруженную ранее модель в файл model.keras. Если модель отсутствует, система сможет функционировать, но классификация состояний будет возвращать «Err».

В активированном виртуальном окружении выполните команду:

```
python main.py
```

После запуска сервер будет доступен по адресу <http://localhost:9000>.

Через веб-интерфейс можно:

- добавлять и редактировать состояния;

- загружать обучающие видео в соответствующие каталоги;
- инициировать процесс дообучения модели.

Все данные сохраняются в локальной базе данных `states.db`, а загруженные видео – в подкаталогах `train/Sxxxx`.

Для запуска процесса дообучения выполните в активном окружении:

```
python train.py --train_dir train
```

По завершении работы будет создан или обновлён файл `model.keras`, который автоматически используется системой при следующих запусках.

7. Указания по эксплуатации Системы

При запуске программного обеспечения «Интеллектуальная система видеомониторинга и контроля производственных процессов» в браузере автоматически открывается страница, на которой в центре внимания находится выпадающий список источников видео. Пользователь видит там доступные файлы MP4 из папки test_data, опцию «webcam» для захвата с веб-камеры и поле для ввода RTSP-URL. Выбрав нужный источник и нажав кнопку «Сменить источник», он запускает потоковую трансляцию, которая отображается в большом окне с видеопотоком.

Интеллектуальная система видеомониторинга и контроля производственных процессов

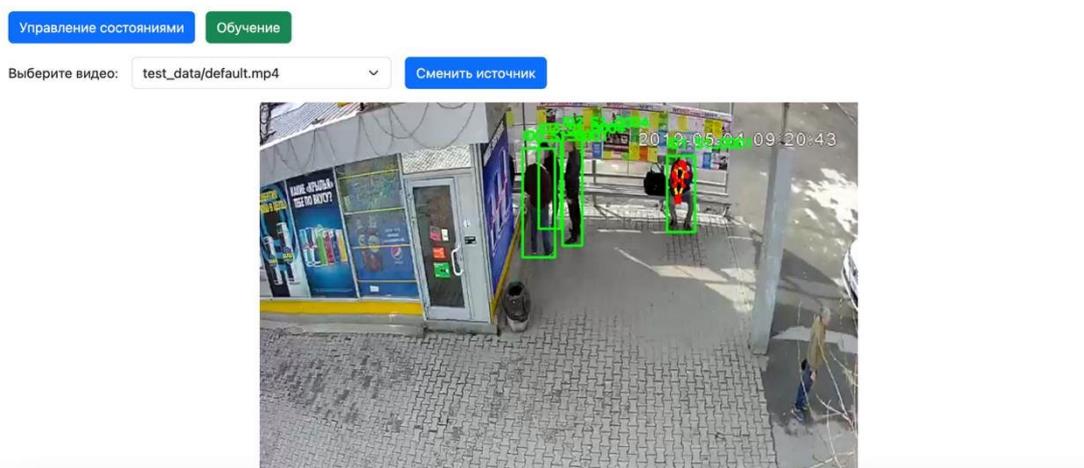


Рисунок 1. Главная страница

В том же окне, чуть выше кадра видео, расположены две кнопки: «Управление состояниями» и «Обучение». При нажатии первой открывается модальное окно со списком всех кодов и описаний состояний, хранящихся в базе данных SQLite.

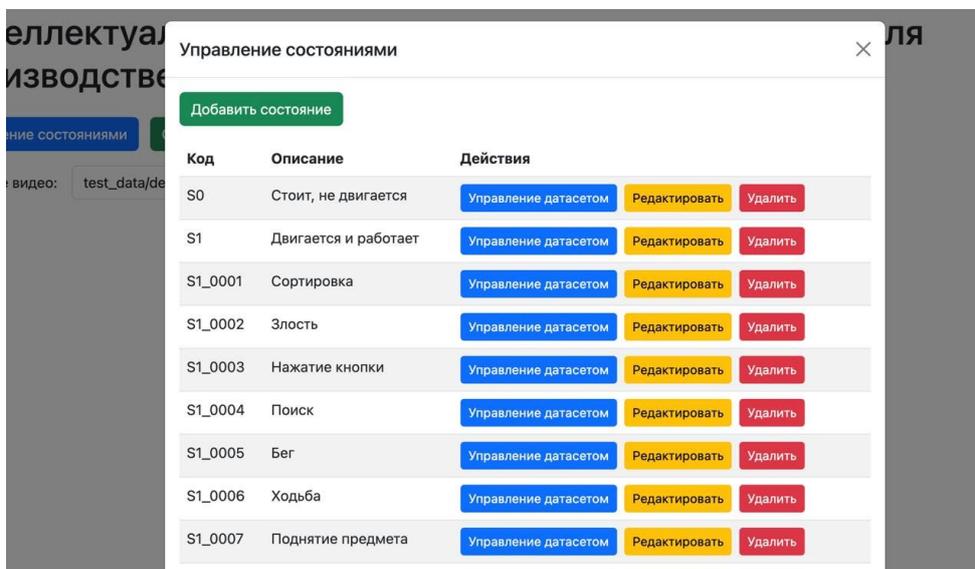


Рисунок 2. Окно состояний

Там же можно добавить новое состояние, отредактировать имеющееся или удалить ненужное, а также выбрать «Управление датасетом» для конкретного кода.

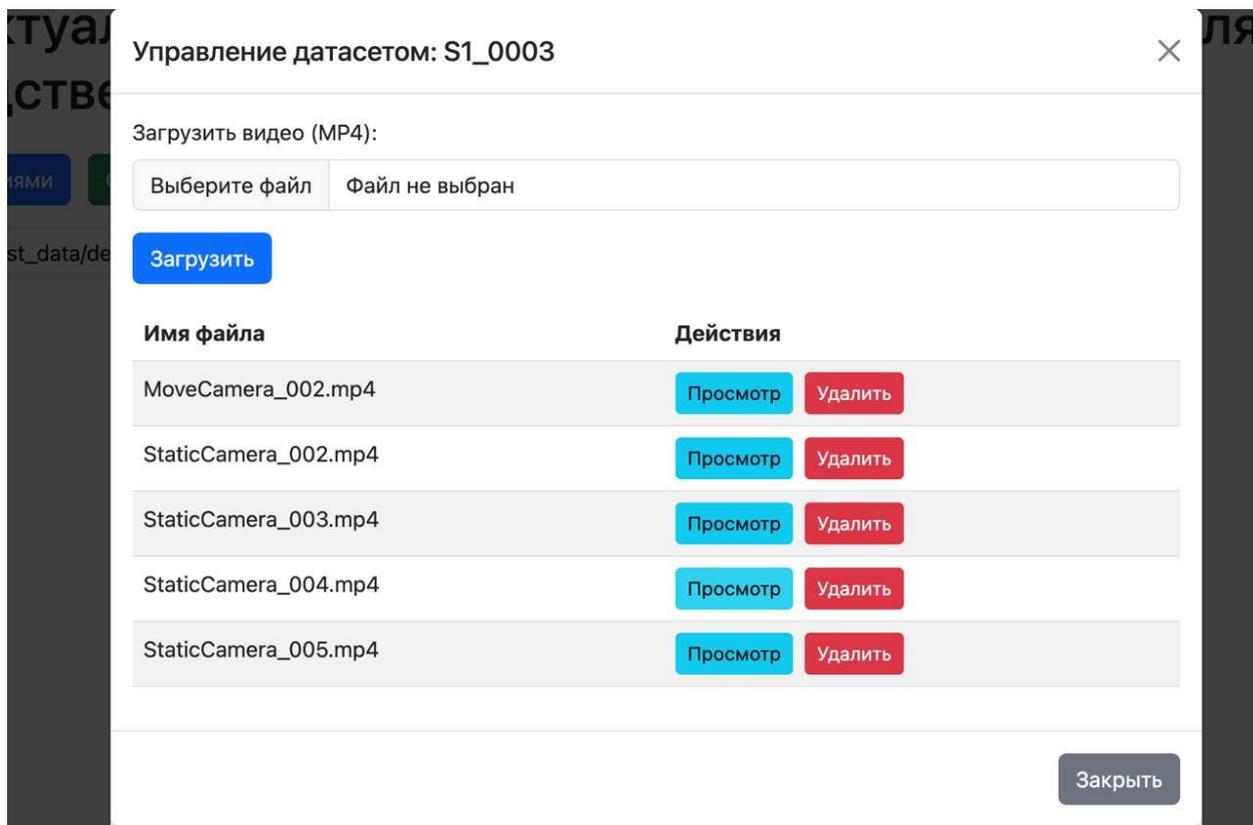


Рисунок 3. Управление датасетом

При выборе «Обучение» появляется другое модальное окно, где выводится пошаговый текстовый лог процесса обучения модели и кнопка для его запуска.

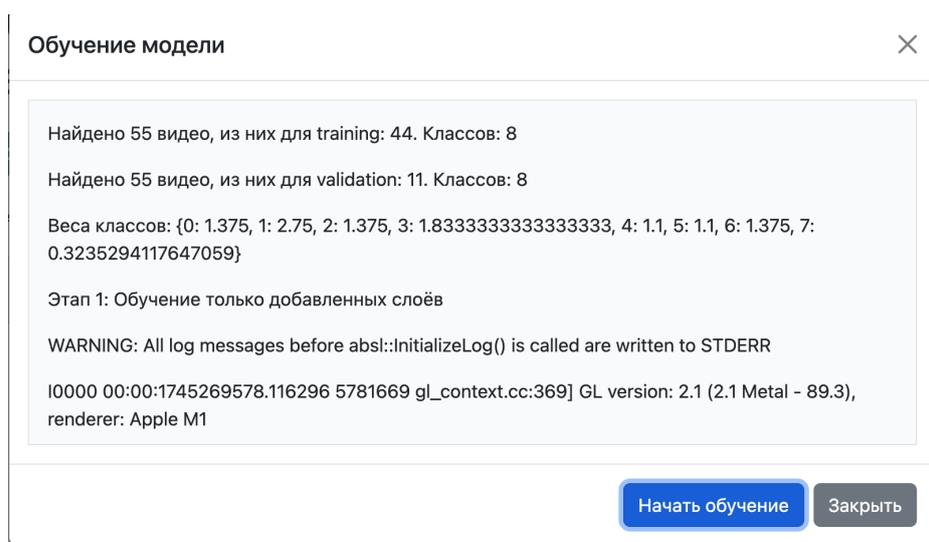


Рисунок 4. Обучение

Снизу под панелью управления видеопотоком находится раздел «Лог событий», представляющий собой таблицу с тремя колонками: временем события, идентификатором трека и текстовым описанием. Пользователь может ввести фрагмент временной метки или номер трека в поля над таблицей, и содержимое автоматически фильтруется по этим параметрам.

Лог событий

Фильтр по времени (YYYY-MM-DD HH:MM:SS) Фильтр по ID объекта

Время	ID объекта	Сообщение
2025-04-22 00:04:42	0	2025-04-22 00:04:42 - Track 0: Нажатие кнопки
2025-04-22 00:04:42	1	2025-04-22 00:04:42 - Track 1: Нажатие кнопки
2025-04-22 00:04:43	2	2025-04-22 00:04:43 - Track 2: Бег
2025-04-22 00:04:43	3	2025-04-22 00:04:43 - Track 3: S9
2025-04-22 00:04:43	4	2025-04-22 00:04:43 - Track 4: Ходьба
2025-04-22 00:04:45	0	2025-04-22 00:04:45 - Track 0: Нажатие кнопки
2025-04-22 00:04:45	1	2025-04-22 00:04:45 - Track 1: Ходьба
2025-04-22 00:04:45	2	2025-04-22 00:04:45 - Track 2: Ходьба
2025-04-22 00:04:46	0	2025-04-22 00:04:46 - Track 0: Транспортировка
2025-04-22 00:04:46	1	2025-04-22 00:04:46 - Track 1: S9
2025-04-22 00:04:46	2	2025-04-22 00:04:46 - Track 2: Ходьба
2025-04-22 00:04:46	3	2025-04-22 00:04:46 - Track 3: Ходьба
2025-04-22 00:04:46	4	2025-04-22 00:04:46 - Track 4: Транспортировка

Рисунок 5. Лог событий

Справа от лога расположена секция «График состояний объекта», в которой сначала отображается выпадающий список доступных треков, а под ним — столбчатая диаграмма. Диаграмма отражает количество попаданий выбранного объекта в каждое из состояний за всё время работы текущей сессии.

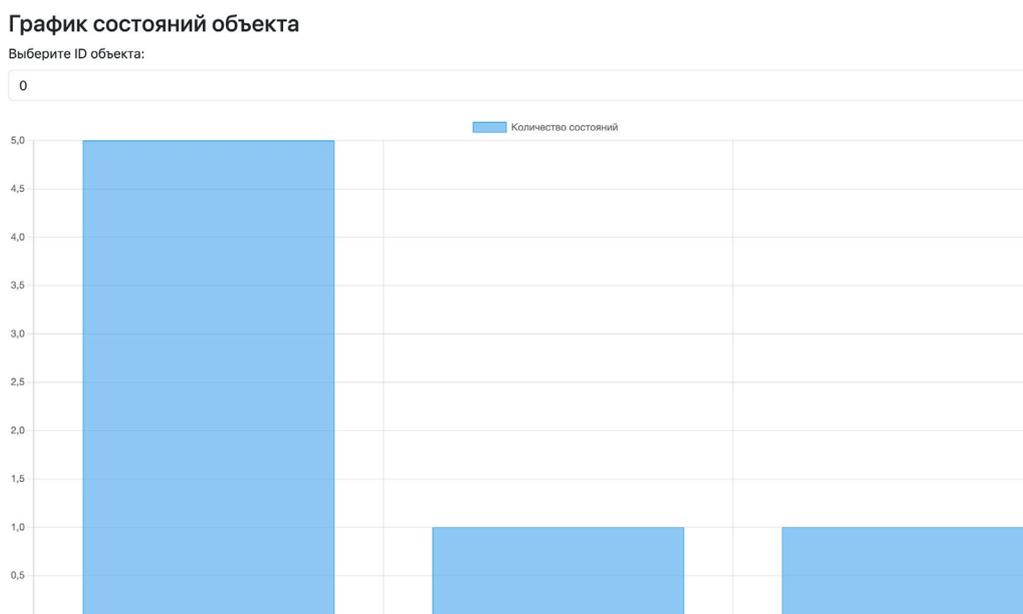


Рисунок 6. График состояний объекта

Когда пользователь открывает модальное окно «Управление состояниями», система отправляет AJAX-запрос на эндпоинт `/get_states`. Получив ответ в формате JSON, скрипт динамически формирует таблицу с кнопками для редактирования или удаления каждой строки. При добавлении или обновлении состояния отправляется POST-запрос на `/add_state` или `/update_state`, и в файловой системе автоматически создаётся (или переименовывается) папка `train/<код>` для хранения видеопримеров.

Если же пользователь нажимает «Управление датасетом», то открывается окно, в котором сначала загружается список уже имеющихся видео для выбранного состояния через GET-запрос к `/get_videos/<код>`. В этом списке каждое видео можно либо предварительно просмотреть, либо удалить. Загрузка нового файла происходит путем отправки multipart-form данных на эндпоинт `/upload_video/<код>`, при этом прогресс отображается в виде бегущей полосы.

При клике на кнопку «Обучение» внутри модального окна выполняется асинхронный POST-запрос на `/train_model`. Сервер сначала проверяет, что для каждого состояния в папке `train` есть хотя бы одно видео, и при отсутствии какого-либо видеопримера отправляет предупреждение клиенту и закрывает окно. Если все папки заполнены, запускается внешний процесс `python train.py --train_dir train`, а каждая строка его вывода передается клиенту через Socket.IO и отображается в реальном времени в блоке лога обучения. По завершении процесса клиент получает сообщение об успешном окончании или об ошибке и окно обучения закрывается.

На стороне сервера класс `VideoProcessor` читает кадры в отдельном потоке, кладет их в очередь, а затем обрабатывает: сначала детектирует людей при помощи YOLO, затем применяет трекинг (DeerSort) и оценивает позу в кадре с помощью MediaPipe. Далее, если загружена модель Keras (`model.keras`), происходит классификация каждого найденного объекта на одно из известных состояний; при отсутствии модели состояние помечается как «Err». К каждому кадру добавляются `bounding-box`'ы, скелетная разметка и текстовые метки, после чего изображение кодируется в JPEG и отправляется клиенту через событие `video_frame`.

Параллельно каждые две секунды другой поток собирает последние сто записей журнала и актуальную статистику по каждому треку из внутренних структур `VideoProcessor` и посылает их клиенту через события `log_update` и `chart_update`. На стороне клиента эти данные автоматически обновляют содержимое таблицы лога и график состояний.

8. Техническое обслуживание

Интеллектуальная система видеомониторинга и контроля производственных процессов не требует специального обслуживания, выходящего за рамки руководства пользователя и технической документации, поэтому обслуживание и сопровождение изделия производителем не предусмотрены.

Для технической поддержки или при обнаружении ошибок работы программы, не описанных в технической документации, рекомендуется обратиться по указанным контактными данным

9. Требования безопасности и меры предосторожности

Перед эксплуатацией аппаратного обеспечения, используемого с интеллектуальной системой видеомониторинга и контроля производственных процессов, необходимо проводить визуальный осмотр видеокамер и стороннего оборудования на наличие повреждений (целостность линз и других компонентов).

Отчеты о классификации состояний объектов сохраняются на компьютере пользователя. Пользователь обязан соблюдать все требования по защите данных в соответствии с законодательством Российской Федерации. Для обеспечения целостности отчетов рекомендуется периодическое резервирование с использованием функций операционной системы.

Рекомендуется устанавливать камеры на безопасном расстоянии от людей и предметов.

Эксплуатация системы должна осуществляться в помещении с умеренным освещением, так как слишком яркий или темный свет может затруднить работу видеокамер.

Использование системы должно производиться в соответствии с руководством пользователя.

Ограничение срока службы не применимо к данному программному обеспечению.

10. Словарь терминов

Интеллектуальная система видеомониторинга. Совокупность программных решений и технологий, позволяющая осуществлять анализ видеопотоков в реальном времени, обнаружение, отслеживание и классификацию состояний объектов на производственных площадках, а также управление данными и их визуализацию через веб-интерфейс.

Операционная система (ОС). Комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

Программное обеспечение (ПО). Программа или множество программ, используемых для управления электронными вычислительными устройствами (персональный компьютер, видеокамеры).

Система видеомониторинга. Группа устройств и программ, обеспечивающих сбор, обработку и анализ видеоданных для контроля производственных процессов.

Видеокамера. Устройство, фиксирующее видеопоток для последующей обработки и анализа системой.

Монолитная архитектура. Структура программного продукта, при которой все функциональные компоненты объединены в единое приложение с высокой внутренней связностью.

Модуль. Функционально законченный фрагмент программы, реализующий определенные задачи, такие как обработка видео, классификация состояний или управление базой данных.