

Инструкция по установке экземпляра программного обеспечения, предоставленного для проведения экспертной проверки

Наименование ПО: Программный комплекс распознавания и оценки физической активности «AI Цифровая физкультура».

Правообладатель: ООО «ЭЙАЙ СОЛЮШНС», Сахалинская область, М.О. Южно-Курильский, Тер. ТОР Курилы; ОГРН 1256500006388 от 26.12.2025; ИНН/КПП 6500027812/650001001.

Документ подготовлен в соответствии с подпунктами «а» и «д» пункта 11 Правил формирования и ведения единого реестра российских программ для ЭВМ и баз данных.

1. Способ предоставления проверочного экземпляра

Проверочный экземпляр предоставлен **в виде набора необходимых файлов** (исходные тексты и файлы сборки) с автоматизированным развёртыванием средствами Docker Compose.

Ссылка на экземпляр программного обеспечения: <URL архива с дистрибутивом> (файл `ai-fizkultura-<версия>.zip`, контрольная сумма SHA-256: <хеш>).

Дополнительно, по запросу эксперта, может быть предоставлен **удалённый доступ с правами root** (SSH) к инфраструктуре с уже развёрнутым экземпляром ПО — см. раздел 7.

2. Требования к системе

2.1. Аппаратные требования

Параметр	Минимально	Рекомендуется
Процессор	4 ядра x86-64	8 ядер
Оперативная память	8 ГБ	16 ГБ и более
Дисковое пространство	20 ГБ свободно	40 ГБ (SSD)
Графический ускоритель	не требуется (режим CPU)	дискретный NVIDIA с поддержкой CUDA (например, RTX 3050 и выше)
Камера	веб-камера USB (UVC)	веб-камера USB либо IP-камера с поддержкой RTSP

Без графического ускорителя ПО полностью работоспособно в режиме CPU (`RTMPOSE_DEVICE=cpu`), скорость обработки кадра снижается примерно в 10 раз (~400 мс против ~30 мс). Для функциональной проверки режима CPU достаточно.

2.2. Программные требования

- **Операционная система:** Linux x86-64 с ядром 5.x и выше. Проверено на Ubuntu 22.04 LTS; поддерживаются отечественные ОС Astra Linux, РЕД ОС.
- **Docker Engine** версии 24.0 и выше.
- **Docker Compose** версии 2.20 и выше (плагин `docker compose`).
- **Для режима GPU (опционально):** проприетарный драйвер NVIDIA версии 535 и выше, **NVIDIA Container Toolkit**.
- **Браузер** на базе Chromium (Google Chrome, Яндекс.Браузер, Chromium) актуальной версии — для доступа к клиентскому интерфейсу.

Доступ в сеть Интернет требуется **только на этапе сборки образов** (загрузка базовых образов и зависимостей). В процессе работы ПО обращение к внешним сервисам не требуется: веса нейросетевых моделей включены в состав образа.

3. Установка и настройка (пошагово)

Шаг 1. Установить Docker и Docker Compose

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

# проверка
docker --version && docker compose version
```

Шаг 2. (Опционально, для режима GPU) Установить NVIDIA Container Toolkit

```
# драйвер NVIDIA должен быть уже установлен: проверить командой
nvidia-smi

curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey \
  | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-
toolkit.list \
  | sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-
keyring.gpg] https://#g' \
  | sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
sudo apt-get update && sudo apt-get install -y nvidia-container-toolkit
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker

# проверка доступа контейнера к GPU
docker run --rm --gpus all nvidia/cuda:12.2.0-base-ubuntu22.04 nvidia-smi
```

Шаг 3. Распаковать дистрибутив

```
unzip ai-fizkultura-<версия>.zip -d ai-fizkultura
cd ai-fizkultura
```

Шаг 4. Настроить параметры запуска

Создать файл `.env` в корне проекта:

```
cat > .env <<'EOF'
# Устройство инференса нейросетей: cuda (при наличии GPU) либо cpu
RTMPOSE_DEVICE=cuda
# Режим точности моделей: lightweight | balanced | performance
RTMPOSE_MODE=balanced
# Порог детекции человека (0.05 – устойчивая детекция на киоск-камере)
RTMPOSE_DET_SCORE=0.05
# Порог достоверности позы (отсечение ложных скелетов)
RTMPOSE_MIN_POSE_SCORE=0.45

# Порты обратного прокси (при конфликте с другими сервисами – изменить)
NGINX_HTTP_PORT=80
NGINX_HTTPS_PORT=443

# База данных
POSTGRES_USER=ais5
POSTGRES_PASSWORD=<задать пароль>
POSTGRES_DB=ais5_admin

# Административный доступ
ADMIN_LOGIN=admin
ADMIN_PASSWORD=<задать пароль>
JWT_SECRET=<задать случайную строку>
EOF
```

Если GPU отсутствует: установить `RTMPOSE_DEVICE=cpu` и удалить (закомментировать) в файле `docker-compose.yml` секции `runtime: nvidia` и `deploy.resources.reservations.devices` у сервиса `rtmpose_server`.

Шаг 5. Создать общую сеть и собрать образы

```
docker network create ais5_net 2>/dev/null || true
docker compose build
```

Сборка занимает от 15 до 40 минут в зависимости от канала связи (загружаются базовые образы и веса нейросетевых моделей).

Шаг 6. Запустить систему

```
docker compose up -d
docker compose ps          # все сервисы должны быть в состоянии Up
```

4. Проверка работоспособности установки

4.1. Проверка ИИ-ядра

```
curl http://localhost:10290/health
```

Ожидаемый ответ (поле `device` показывает используемое устройство инференса):

```
{"ok": true, "device": "cuda", "mode": "balanced"}
```

Проверка результата детекции и оценки поз:

```
curl http://localhost:10290/debug/detect
```

4.2. Проверка прочих сервисов

```
curl -I http://localhost:4575/      # клиентское приложение упражнений → 200 OK
curl -I http://localhost/          # обратный прокси → 200 OK
docker compose logs --tail=50 rtmpose_server
```

4.3. Функциональная проверка (сценарий эксперта)

1. Открыть в браузере Chromium адрес `http://<адрес сервера>/`.
2. Разрешить сайту доступ к камере.
3. На главном экране выбрать раздел «Школа», затем упражнение «Приседания».
4. Указать число участников: 1. Дождаться обратного отсчёта (5 → 1).

5. Встать в кадр в полный рост так, чтобы корпус и ноги попадали в поле зрения камеры.
6. Выполнить 3–5 приседаний.

Ожидаемый результат: на экране отображается распознанный скелет; счётчик повторений увеличивается на каждое приседание; по завершении занятия формируется отчёт с числом повторений, временем и оценкой качества выполнения.

Аналогично проверяются разделы «ГТО», «Фитнес» и «Детский раздел».

5. Развёртывание проверочного экземпляра на сервере

Порядок действий полностью соответствует разделу 3. Дополнительно для промышленной установки:

```
# автозапуск при перезагрузке сервера обеспечивается политикой restart: unless-stopped,
# заданной в docker-compose.yml для всех сервисов

# просмотр журналов
docker compose logs -f

# остановка
docker compose down

# обновление версии
git pull && docker compose build && docker compose up -d
```

Настройка TLS (HTTPS). Поместить файлы сертификата и приватного ключа в каталог `./ssl/`, после чего перезапустить обратный прокси:

```
docker compose restart nginx
```

Подключение IP-камеры (RTSP). Указать источник в файле `rtsp_gateway/go2rtc.yaml`, затем перезапустить медиашлюз:

```
docker compose restart go2rtc
```

6. Состав сервисов после развёртывания

Сервис	Назначение	Адрес для проверки
nginx	Обратный прокси, единая точка входа	http://<сервер>/
gesture_web_demo	Клиентское приложение упражнений	http://<сервер>:4575/
rtmpose_server	Ядро ИИ: инференс нейросетей, трекинг	http://<сервер>:10290/health

Сервис	Назначение	Адрес для проверки
gesture_ws_server	Прикладная логика, события занятия	ws://<сервер>:10282/ws
admin_api	Административный API	через прокси, /api/
postgres	База данных	внутренняя сеть
go2rtc	Медиашлюз RTSP → WebRTC	http://<сервер>:1984/api
camera_configurator	Мастер настройки камер	http://<сервер>:2105/
frog_pond , games , arcade , runner	Детские игровые модули	/frog/ , /games/ , /arcade/ , /runner/

7. Альтернативный вариант: удалённый доступ к развёрнутому экземпляру

По запросу эксперта правообладатель предоставляет доступ с правами `root` к серверу с развёрнутым экземпляром ПО:

Протокол: SSH
Адрес: <IP-адрес или доменное имя>
Порт: <22>
Логин: root
Пароль: <пароль> (либо приватный ключ, передаётся отдельно)

Каталог с ПО: /opt/ai-fizkultura
Просмотр состояния сервисов: cd /opt/ai-fizkultura && docker compose ps
Просмотр журналов: cd /opt/ai-fizkultura && docker compose logs -f
Веб-интерфейс: http://<адрес>/

Расположение файлов ПО и его компонентов:

Путь	Содержание
/opt/ai-fizkultura/docker-compose.yml	Описание всех сервисов, портов, томов
/opt/ai-fizkultura/rtpose_server/	Ядро ИИ: server.py (инференс), tracker.py (трекинг), ball_endpoint.py (детекция мяча)
/opt/ai-fizkultura/rtpose_server/models_cache/	Веса нейросетевых моделей в формате ONNX
/opt/ai-fizkultura/gesture_web_demo/	Клиентское приложение упражнений
/opt/ai-fizkultura/admin_server/	Административный API
/opt/ai-fizkultura/frog_pond/ , games/ , arcade/ , runner/	Игровые модули
/opt/ai-fizkultura/rtsp_gateway/	Конфигурация медиашлюза и обратного прокси

Томы Docker с данными: pgdata (база данных), admin_uploads (загруженные файлы).

8. Устранение типичных затруднений

Признак	Причина	Решение
<code>/health</code> возвращает <code>"device": "cpu"</code> при наличии GPU	Не установлен NVIDIA Container Toolkit	Выполнить шаг 2, перезапустить Docker
Сервис <code>rtmpose_server</code> не стартует	Нет доступа к GPU	Установить <code>RTMPOSE_DEVICE=cpu</code> , убрать секции <code>runtime: nvidia</code> и <code>deploy.resources</code>
Браузер не запрашивает доступ к камере	Открыт <code>http://</code> не на <code>localhost</code>	Настроить HTTPS (раздел 5) — доступ к камере требует защищённого соединения
Скелет не распознаётся	Недостаточное освещение либо человек не помещается в кадр	Обеспечить равномерное освещение; встать в кадр в полный рост
Повторения не считаются	Слишком высокий порог детекции	Установить <code>RTMPOSE_DET_SCORE=0.05</code> и перезапустить <code>rtmpose_server</code>
Порт 80 занят	Конфликт с другим веб-сервером	Изменить <code>NGINX_HTTP_PORT</code> в <code>.env</code>
IP-камера не открывается («go2rtc: ошибка 502»)	Медиашлюз недоступен	<code>docker compose restart go2rtc</code> , проверить <code>rtsp_gateway/go2rtc.yaml</code>

9. Контакты технических специалистов

По вопросам развёртывания, настройки и функционирования проверочного экземпляра консультируют технические специалисты правообладателя:

Ответственный специалист	Воронов Роман Владиленович, <должность>
Телефон	+7 (999) 596-09-66
Электронная почта	info@aisolus.ru
Время консультаций	рабочие дни, 09:00–18:00 (по московскому времени)

Документация, содержащая описание функциональных характеристик ПО и информацию, необходимую для установки и эксплуатации, размещена по адресу: <https://ваша-школа.рф/docs/>.

Руководитель ООО «ЭЙАЙ СОЛЮШНС»

<должность> ____ / <Фамилия И.О.> /

«_» _ **20** г. М.П.