**Отчет о проделанной работе по стажировке  
«Создание ИИ-системы по оценке исполнения ката шотокан карате-до»**

### **1. Подготовка и обработка видеоданных**

* **Калибровка камер**:  
  Проведена внутренняя и внешняя калибровка 4 камер с использованием шахматного шаблона. Для камеры 4 выявлены проблемы с определением углов шаблона (размерность [3,5] вместо [3,6]), что привело к повышенной ошибке калибровки (до 3.5 пикселей). Результаты сохранены в YAML-файлах с матрицами поворота, векторами трансляции и коэффициентами дисторсии.
  + Пример ошибки повторной проекции: 5.94–6.06 пикселей.
  + Выявлена сильная дисторсия у камеры 1 (k2=5.77, k3=-78.45), требующая коррекции.
* **Синхронизация видео**:  
  Реализована синхронизация видео по движению ключевых точек (запястья) с использованием перекрестной корреляции. Найдены временные сдвиги:
  + cam2 опережает cam1 на 57 кадров;
  + cam4 опережает cam1 на 1 кадр.  
    Видео объединены в сетку 2×2 с помощью FFmpeg для визуальной проверки.
* **Извлечение ключевых точек**:  
  Использованы MediaPipe Pose и YOLOv8 для трекинга поз. Координаты 33 ключевых точек сохранены в CSV-файлы. Для обработки применялся Google Colab с ноутбуками, доступными по ссылкам в отчете.
* **Триангуляция 3D-координат**:  
  Построены 3D-модели скелета на основе данных с 4 камер. Использован метод взвешенной триангуляции с учетом достоверности точек. Результаты визуализированы в Matplotlib и сохранены в формате NumPy.

### **2. Разработка модели классификации поз**

* **Подготовка датасета**:  
  Обработано 8148 изображений, распределенных по 46 классам поз. Для каждого изображения извлечены координаты ключевых точек с помощью MediaPipe. Проведена фильтрация данных по видимости точек.
* **Обучение модели**:  
  Создана модель на основе Random Forest с добавлением признаков углов между суставами.
  + Итоговая точность: 56.5% (для 9 основных классов, включая heiko-dachi, l-kokutsu-dachi, r-step-fwd).
  + Precision/Recall: Наивысшие показатели для heiko-dachi (precision=0.8, recall=0.92), низкие — для малочисленных классов (например, l-step-45: precision=0.18).
  + Сохранены артефакты: random\_forest\_model\_9\_no\_rei.pkl, scaler.pkl, label\_encoder.pkl.
* **Оптимизация**:  
  Для улучшения качества предложено:
  + Увеличить датасет для малочисленных классов;
  + Добавить признаки расстояний и углов между точками.

### **3. Система реального времени**

* **Настройка RTMP-стриминга**:  
  Развернута на виртуальной машине Yandex Cloud (8 vCPU, 16 ГБ RAM, 100 ГБ SSD). Использованы:
  + **Larix Broadcaster** для трансляции с мобильных устройств;
  + **Node Media Server** и **OBS Studio** для приема потоков;
  + **FFmpeg** для обработки и комбинирования видео.  
    Пропускная способность: 70–75 Мбит/с для 4 камер.
* **Интеграция с Flask**:  
  Реализован веб-интерфейс для управления процессом:
  + Запуск/остановка распознавания;
  + Просмотр видео через HLS.js;
  + Сохранение результатов в CSV и видеофайлы (например, ivan\_output.mp4).

### **4. Результаты и выводы**

* **Достижения**:
  + Создан конвейер от съемки видео до 3D-реконструкции и классификации поз.
  + Развернута система реального времени с точностью 56.5% для базовых поз.
  + Настроена трансляция на облачном сервере с поддержкой RTMP/HLS.
* **Проблемы и решения**:
  + Высокая дисторсия камеры 1 — предложена коррекция коэффициентов;
  + Низкая точность для редких классов — рекомендовано расширение датасета.
* **Дальнейшие шаги**:
  + Внедрение синхронизации через SEI Timecode в Larix Broadcaster;
  + Использование GPU для ускорения обработки (например, NVIDIA Tesla T4);
  + Разработка нейросетевой модели (например, LSTM) для анализа динамики движений.