

# Зачётные задачи по курсу компьютерного зрения

---

Ниже представлены зачётные задания. Все задания приближены к реальности и предполагают практическую реализацию.

Каждое задание включает:

- использование модели (обученной или предобученной),
- обёртку в виде микросервиса (FastAPI, Flask и др.),
- деплой в виде Docker Compose, разворачиваемого на учебном кластере,
- UI или REST-интерфейс для пользователя.

Выберите **одну задачу**, которая ближе к вашим интересам: кому-то будет ближе обучение модели, кому-то — деплой и реализация микросервиса.

---

## Важные даты

- **Чекпоинты:** Конкретные даты будут позже.
- **Предзащита:** 07.05 (вторник) — проверка прогресса, готовности микросервиса, кода и архитектуры. Обязательное участие с демонстрацией текущего результата (или мокапа).
- **Финальная защита:** 12.05 (воскресенье) — демонстрация законченного решения, ответы на вопросы. Необходимо показать работу всего пайплайна: от загрузки до вывода результата.

Каждый студент должен быть готов коротко рассказать:

- идею и цель проекта,
  - выбранную архитектуру модели и обоснование,
  - как реализован микросервис и API,
  - как происходил деплой.
- 



## Работа с кластером

Каждому студенту будет предоставлен доступ к учебному кластеру (информация о доступе будет опубликована дополнительно).

На кластере можно:

- обучать и дообучать модели,
- выполнять инференс на CPU или GPU,
- тестировать микросервисы.

**Каждый студент обязан подготовить `docker-compose.yml` файл**, который описывает развёртывание микросервиса. Контейнер должен включать:

- модель или ссылку на веса,
- REST API (на FastAPI или Flask),
- все необходимые зависимости,

- инструкции по запуску (например, `start.sh` или `CMD`).

Файл `docker-compose.yml` необходимо отправить DevOps-специалисту — [Алексею Васильевичу](#) — для развёртывания на сервере.

## Нулевая задача (обязательная для всех)

Перед началом реализации проекта студент должен:

- найти и обосновать выбор модели (или весов),
- найти и подготовить датасет,
- указать формат входных и выходных данных,
- составить схему пайплайна (в виде текста или схемы),
- предоставить короткое описание API.



### 1. Сегментация 2D изображения таза с редактированием

**Фокус:** интерфейс + обработка результата

Чекпоинты:

- **Неделя 1:** прототип интерфейса и описание pipeline (UI → модель → UI)
- **Неделя 2:** реализована сегментация изображения
- **Неделя 3:** реализация редактирования сегмента (движение точек/контуров)
- **Неделя 4:** микросервис в Docker Compose + демонстрация работы на кластере



### 2. Сегментация 3D бедренной кости и экспорт результата

**Фокус:** загрузка и обработка медицинских данных

Чекпоинты:

- **Неделя 1:** подбор и загрузка модели (предобученной) + минидатасет
- **Неделя 2:** вывод сегментации на одном объёме, базовая визуализация
- **Неделя 3:** экспорт результата (Nifti → STL или PNG срезы)
- **Неделя 4:** развёрнутый REST-интерфейс в Docker Compose на кластере

**Дополнительно:** использовать MONAI или nnUNet, MedSAM. Визуализация результата — через MedPy или скриншоты срезов.



### 3. Трекинг людей в видео + API с Live-визуализацией, подсчет статистики

**Фокус:** реалтайм + инженерия

Чекпоинты:

- **Неделя 1:** базовый трекинг (напр., YOLOv8 + DeepSORT) на видео

- **Неделя 2:** API с трекингом и возвратом координат
- **Неделя 3:** визуализация треков в реальном времени
- **Неделя 4:** развёртывание через Docker Compose + работа на кластере

**Дополнительно:** опционально — WebSocket для real-time визуализации; клиентская часть на HTML/JS.

---