

Экзамен по дисциплине «Компьютерное зрение»

- Автор: **Лобан Константин Михайлович**, группа: М08-402ПА
- Дата экзамена: 13.01.2026
- Полное содержимое проекта содержится в [репозитории на GitHub](#)
- Ссылка на рабочий [ноутбук в Google Colab](#)

Комментарии автора:

В ноутбуке **FinalWork_LobanKM.ipynb** содержится полный пайплайн обучения моделей детекции и генерации в соответствии с заданием.

Описание функций приложения:

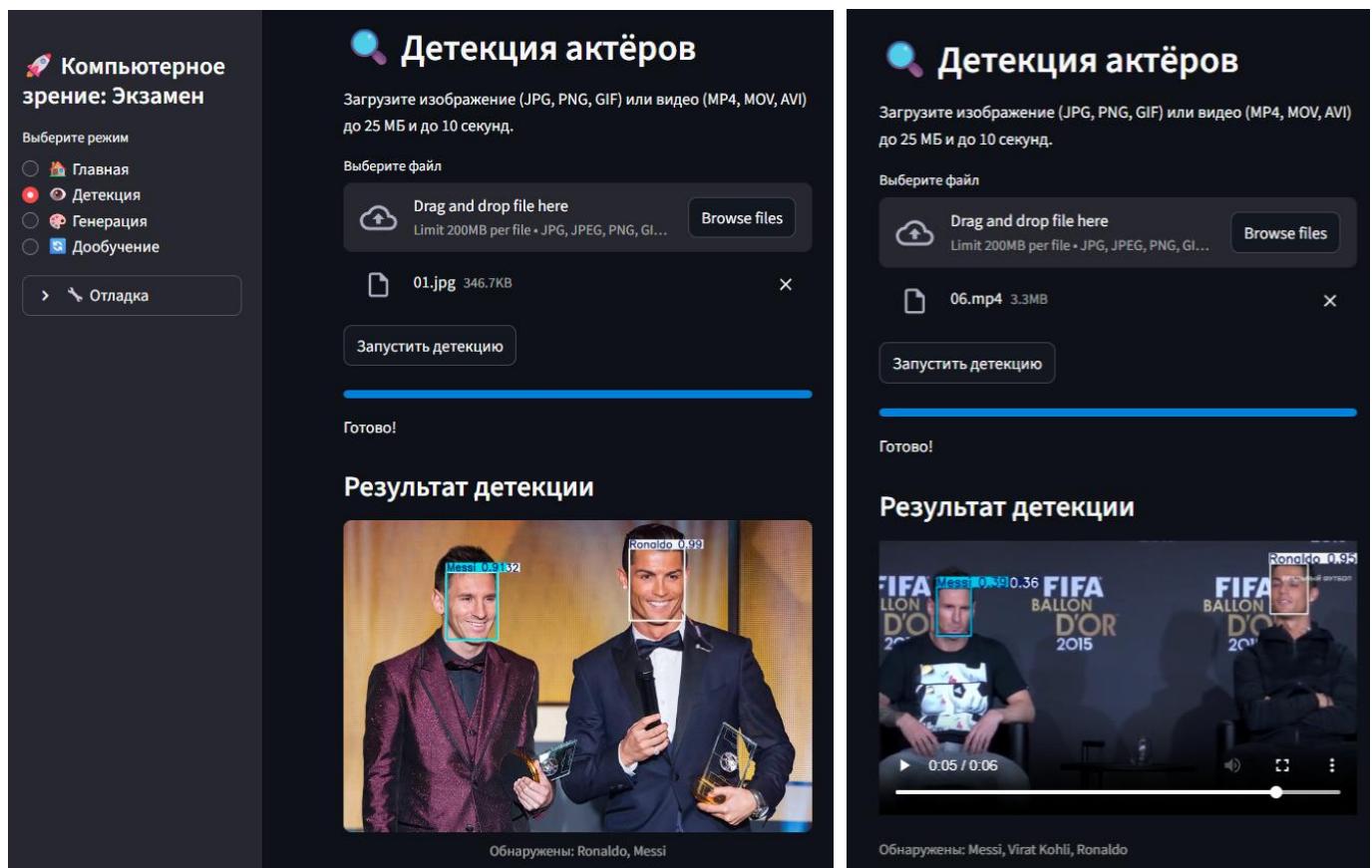
Локальный запуск приложения совершается исполнением следующего кода в терминале:

```
git clone https://github.com/Aeverandi/mipt_cv_exam  
cd mipt_cv_exam  
pip install -r requirements.txt  
streamlit run app.py
```

Далее происходит запуск интерфейса (первый раз длительный, связанный с загрузкой моделей, прежде всего Stable Diffusion). Интерфейс реализован с помощью библиотеки **Streamlit**, так как она мне показалась наиболее подходящей для этой цели и относительно простой для новичка в этом. После этого пользователю становятся доступны следующие режимы использования:

Детекция

Пользователю предлагается загрузить фото или короткое видео, после чего программа отображает размеченное фото и указывает, кого она «узнала» на этих фото. Просмотр логов доступен прямо в интерфейсе приложения в каждом разделе.



Генерация

Пользователю предлагается знаменитость из списка, которую можно сгенерировать. Имеется спойлер с более тонкими настройками генерации и уже готовыми промптами.

The screenshot shows the 'Generation' interface. On the left, a sidebar lists modes: Главная (Main), Детекция (Detection), Генерация (Generation), and Дообучение (Fine-tuning). The 'Генерация' mode is selected. A central panel displays the title 'Генерация изображений' (Image Generation) with a painter palette icon. Below it, a message says 'Выберите актёра из списка, чтобы сгенерировать его изображение' (Select an actor from the list to generate his image). A dropdown menu shows 'Messi'. To the right is a detailed configuration panel titled 'Настройки генерации' (Generation Settings). It includes fields for a positive prompt ('a high-quality professional portrait photograph of messi, looking directly at camera, natural expression, cinematic lighting 4k resolution detailed skin texture professional'), a negative prompt ('blurry, low quality, distorted face, extra limbs, disfigured, bad anatomy, duplicate, morbid, mutilated, out of frame, extra fingers, mutated hands, poorly drawn hands, poorly'), a generation step count slider set to 50, a resolution dropdown set to 512, and a 'Генерировать изображение' (Generate Image) button. To the right of the configuration is a preview area titled 'Сгенерированное изображение: messi' (Generated image: messi) with a star icon. It shows a portrait of Lionel Messi in a Barcelona jersey. Below the image are status messages: 'Размер: 512x512 px' (Size: 512x512 px), a 'Скачать изображение' (Download image) button, and a green message 'Изображение сгенерировано и сохранено в лог' (Image generated and saved to log). At the bottom, there's a link to 'Последние записи лога' (Recent log entries).

Дообучение

Пользователю предлагается загрузить архив с фото одного актера, которые затем автоматически размечиваются (*MTCNN* в случае наличия GPU на запускаемом устройстве, либо *HaarCascade* из *OpenCV* – в случае CPU).

The screenshot shows the 'Fine-tuning' interface. The sidebar shows the same mode selection as the previous screen. The main panel has a title 'Дообучение модели' (Fine-tune model) with a person icon. Below it, a section titled 'Шаг 1: Загрузка данных' (Step 1: Load data) shows a message 'Загрузите ZIP-архив с фотографиями одного человека для дообучения модели' (Upload a ZIP archive with photos of one person for fine-tuning the model). There is a 'Drag and drop file here' input field containing 'for_train.zip 412.2KB' and a 'Browse files' button. Below this is a field 'Имя актёра (латиницей, например: ben_afflek)' (Actor name (in Latin letters, e.g., ben_afflek)) with 'ben_afflek' entered. A red arrow points to the 'ben_afflek' entry. At the bottom is a large red 'Выполнить разметку' (Perform annotation) button. To the right, a progress bar shows 'Найдено изображений: 19' (Found images: 19) and 'Обработка изображения 19/19' (Image processing 19/19). Below these are green status messages: 'Обнаружено лиц на 16 из 19 изображений (84.2%)' (Detected faces on 16 of 19 images (84.2%)) and 'Использован метод детекции: opencv_cpu' (Detection method used: opencv_cpu).

Пользователю остается только исключить плохо размеченные на его взгляд фото и затем, указав количество эпох обучения и батч, запустить обучение.

Шаг 2: Проверка разметки

Проверьте разметку на каждом изображении. Отклоните изображения с некорректной разметкой.

Всего изображений	Принято	Процент принятых
19	19	100%

Изображение 12 из 19

Разметка для: ben_afflek

Принять разметку Отклонить разметку

Статус изображения: принято

Дообучение модели

Шаг 3: Обучение модели

Обучение модели на данных актёра: ben_afflek

Принято для обучения: 19 изображений

Количество эпох обучения: 50

Размер батча: 8

💡 Рекомендации:

- Для небольшого количества изображений (3-10) используйте 15-25 эпох
- Для большого количества изображений (>10) используйте 30-50 эпох
- Размер батча 8 оптимален для большинства GPU

🔥 Запустить обучение

> Последние записи лога

По завершении обучения выводятся метрики и новая дообученная модель YOLOv8 загружается в кэш. В дальнейшем новый актер будет доступен в режиме «Детекция».

Компьютерное зрение: Экзамен

Выберите режим:

- Главная
- Детекция
- Генерация
- Дообучение

Отладка

Текущая директория: /root/root/mipt_cv_exam

YOLO модель: загружена (models/YOLO/new.pt)

SD модель: загружена (runwayml/stable-diffusion-v1-5)

Загрузка новой модели...

🎉 Модель успешно дообучена!

Precision	Recall	mAP@0.5	mAP@0.5-0.95
1.000	0.983	0.995	0.830

Точность детекции Полнота детекции Средняя точность при IoU=0.5 Средняя точность при IoU=0.5-0.95

Графики обучения

Результаты обучения YOLO

✓ Новая модель загружена и готова к использованию в детекции!

> Последние записи лога