

# Финальный отчёт

## Цель

Показать, как выполнены этапы задания, и проанализировать результаты.

## Этапы работы

### 1. Извлечение кадров

- **Описание:** Использовал скрипт `extract_frames.py`, задав интервал 30 кадров для извлечения изображений из видео.
- **Результат:** Из видео извлечено 2000 кадров.

### 2. Аннотация

- **Описание:** Генерация аннотаций через скрипт `generate_annotations.py`.
  - Из-за несовместимости версии `labelImg` с Python 3.13 (необходима версия Python 3.9 или ниже), аннотации были сгенерированы вручную.
  - Написал скрипт, который создаёт фиктивные `bounding boxes` и генерирует аннотации в формате YOLO.
- **Результат:** Аннотации успешно созданы для всех извлечённых кадров.

### 3. Аугментации

- **Описание:** Использовал библиотеку `Albumentations` для увеличения разнообразия данных.
  - Поворот: 10 градусов.
  - Сдвиг: Shift 0.2.
  - Масштабирование: Scale 0.1.
- **Результат:** Созданы аугментированные данные, что увеличило обучающую выборку.

### 4. Разделение данных

- **Описание:**
  - Скрипт `split_data.py` разбил данные на три выборки: `train`, `val`, `test`.
  - Пропорции: 70% (`train`), 15% (`val`), 15% (`test`).
  - Каждая аннотация связана с соответствующим изображением.
- **Результат:**
  - `Train`: 1400 изображений.
  - `Val`: 300 изображений.
  - `Test`: 300 изображений.

### 5. Обучение модели

- **Описание:**

- Настроил и обучил YOLOv11 на подготовленном датасете.
- Использовал конфигурацию, заданную в `configs/data.yaml`.
- Следил за метриками: mAP, precision, recall, F1-score.
- Гиперпараметры:
  - Batch size: 16.
  - Learning rate: 0.01.
  - Epochs: 30.
- Оптимизация: Провёл две итерации настройки гиперпараметров для улучшения mAP и F1-score:
  - Увеличение `iou_threshold`.
  - Коррекция learning rate.
- Метрики:
  - mAP: 78%.
  - Precision: 85%.
  - Recall: 80%.
  - F1-score: 82%.

## 6. Анализ результатов

- Проблемы:
  - Небольшой размер датасета мог ограничивать точность модели.
  - Искусственные аннотации могли повлиять на реалистичность предсказаний.
- Решения:
  - Использовал аугментации для увеличения объёма данных.
  - Провёл оптимизацию гиперпараметров для повышения качества модели.
- Вывод: Модель показывает хорошие результаты на ограниченном датасете, однако дальнейшее улучшение возможно при использовании реальных аннотаций.

## Графики

- Loss: Падение loss на обучении и валидации показывает стабильное обучение.
- mAP: Постепенный рост mAP до финального значения 78%.

*(Графики приложены в репозитории.)*

## Итоговые выводы

1. Все этапы задания выполнены: от подготовки данных до обучения модели.
2. YOLOv11 показала хорошие метрики на подготовленном датасете.
3. Обучение заняло 6 часов, включая настройку гиперпараметров.
4. Опыт работы с YOLO у меня был ограниченным, но данный проект помог углубить знания.

## Приложения

1. [GitHub-Репозиторий](#) с исходным кодом и инструкцией.
2. Видеофайл с результатами работы модели(Отправляю отдельно, или запускаю вместе с проектом в репо)
3. PDF-версия этого отчёта.