

Домашнее задание. Детекция и классификация домашних животных на основе YOLO

Цель

Закрепить понимание свёрточных нейронных сетей как обучаемых операторов обработки изображений на примере практической задачи детекции объектов. В рамках домашнего задания необходимо реализовать **мини-систему детекции и классификации домашних животных** на изображениях с использованием архитектуры **YOLO**.

Задание ориентировано на прохождение **полного инженерного цикла**: от данных и разметки → к обучению модели → к анализу качества и ошибок.

Постановка задачи

Необходимо построить систему, которая по входному изображению:

- обнаруживает домашних животных;
- определяет **класс животного**;
- возвращает bounding box и метку класса.

Система должна быть реализована на базе **YOLO** (версия YOLOv5 / YOLOv8 / YOLOv9 / YOLOv11 — на выбор, указать явно).

Задание

1. Выбор классов

Необходимо выбрать **не менее трёх классов** домашних животных, например:

- cat
- dog
- bird

Состав классов должен быть зафиксирован и описан в отчёте.

2. Подготовка датасета

Необходимо подготовить датасет изображений с аннотациями bounding box:

1. Использовать готовый датасет (COCO subset, Open Images, Roboflow) **или** собрать собственный датасет.
2. Минимальный объём:
 - не менее **200–300 изображений суммарно**;

- каждый класс должен быть представлен достаточным числом примеров.

3. Все изображения должны быть размечены bounding box.

4. Датасет необходимо привести к **формату YOLO**:

- корректная структура директорий;
 - корректные **.txt**-аннотации;
 - разбиение на train / val.
-

3. Обучение YOLO-модели

1. Выбрать YOLO-модель и указать:

- версию;
- базовую конфигурацию;
- используются ли предобученные веса.

2. Обучить модель на подготовленном датасете.

3. Зафиксировать:

- параметры обучения;
- количество эпох;
- размер изображений;
- batch size.

4. Сохранить лучшие веса модели.

4. Инференс и визуализация

1. Выполнить инференс модели на тестовых изображениях.

2. Визуализировать результаты:

- bounding box;
- метки классов;
- confidence score.

3. Подготовить набор примеров:

- корректные детекции;
 - ошибки модели.
-

5. Анализ качества и ошибок

Необходимо провести **качественный анализ**, включающий:

1. Примеры пропусков объектов.

2. Примеры ложных детекций.
 3. Примеры перепутанных классов.
 4. Анализ причин ошибок:
 - масштаб объекта;
 - освещение;
 - сложный фон;
 - недостаток данных;
 - архитектурные ограничения YOLO.
-

Требования к реализации

Модель

- Используется архитектура YOLO.
 - Обучение на собственном датасете обязательно.
 - Чистая классификация без bounding box **не допускается**.
-

Код

- Python.
 - Разрешено использование:
 - PyTorch;
 - Ultralytics YOLO API;
 - OpenCV для визуализации.
 - Код должен быть воспроизводимым.
-

Формат сдачи

GitHub-репозиторий, содержащий:

- **data/** — описание датасета или скрипты его подготовки.
- код обучения YOLO-модели.
- код инференса.
- **README .md**, включающий:
 - описание задачи и классов;
 - источник и объём датасета;
 - параметры обучения;
 - примеры корректных детекций;
 - примеры ошибок;

- аналитические выводы.
-

Критерии оценки

| Баллы | Критерий |
|-------|----------------------------------------------------------------|
| 0–3 | Полнота: реализован полный пайплайн от данных до инференса. |
| 0–3 | Корректность: корректная разметка, обучение и работа YOLO. |
| 0–2 | Анализ: осмысленный разбор ошибок и ограничений модели. |
| 0–2 | Репозиторий: аккуратность, воспроизводимость, понятный README. |

Максимум: 10 баллов.

Примечание

Оценивается не достижение максимальной метрики, а **понимание архитектуры YOLO как свёрточной модели детекции**, умение работать с данными и анализировать поведение модели.
