

# Обучение YOLOv8 на своём датасете

YOLOv8 — это последняя версия модели для детекции объектов от Ultralytics, которая позволяет обучить модель на своём собственном датасете. В этой статье рассмотрим весь процесс — от подготовки данных до запуска предсказаний с использованием обученной модели.

## Шаг 1: Подготовка данных

Для обучения YOLOv8 на своём датасете, структура данных должна соответствовать следующим требованиям:

- Создайте папки для изображений и аннотаций, разделив их на тренировочную и валидационную части.
- Каждый файл аннотации для изображения должен быть в формате `.txt`, где каждый объект описывается одной строкой:

```
class_id x_center y_center width height
```

- `class_id` — индекс класса.
- `x_center`, `y_center`, `width`, `height` — координаты центра и размеры объекта, нормализованные в диапазоне от 0 до 1 относительно размеров изображения.

Для создания аннотаций можно использовать такие инструменты, как [Roboflow](#) или [Labellmg](#).

## Шаг 2: Установка YOLOv8

YOLOv8 разработана в фреймворке Ultralytics, поэтому сначала установим его:

```
pip install ultralytics
```

Это автоматически установит YOLOv8 и необходимые зависимости.

## Шаг 3: Настройка конфигурационного файла

Для обучения модели необходимо создать конфигурационный файл `config.yaml`, который укажет пути к данным, количество классов и их названия.

Пример `config.yaml`:

```
# config.yaml

path: ../path/to/your/dataset # Путь к папке с вашим датасетом
train: images/train           # Путь к папке с тренировочными изображениями
val: images/val               # Путь к папке с валидационными изображениями

nc: <количество_классов>     # Укажите количество классов в вашем датасете
names: ['class1', 'class2', ...] # Список с названиями классов
```

Замените `<количество_классов>` на число классов в вашем датасете и перечислите их названия в `names`.

## Шаг 4: Запуск обучения

Когда данные и конфигурационный файл готовы, можно приступить к обучению. Выполните следующий код для запуска процесса:

```
from ultralytics import YOLO

# Загружаем предобученную модель YOLOv8
model = YOLO('yolov8n.pt') # Выберите размер модели: n, s, m, l, x

# Запускаем обучение
model.train(data='config.yaml', epochs=100, batch=16, imgsz=640)
```

### Пояснения к параметрам:

- `data`: путь к конфигурационному файлу.
- `epochs`: количество эпох, на протяжении которых модель будет обучаться. Чем больше эпох, тем лучше обучение, но это требует больше времени.
- `batch`: размер батча — количество изображений, обрабатываемых за один шаг. Оптимальный размер зависит от мощности вашей GPU.
- `imgsz`: размер изображения, на котором будет обучаться модель. YOLOv8 по умолчанию использует 640x640, но можно изменить при необходимости.

## Шаг 5: Оценка и тестирование модели

После завершения обучения вы можете оценить модель на валидационном наборе, чтобы понять её производительность.

```
# Оценка модели на валидационном наборе
metrics = model.val()
print(metrics)
```

Команда `model.val()` возвращает метрики точности, полноты и mAP (среднее значение точности). Эти метрики позволяют оценить, насколько эффективно модель обнаруживает объекты на изображениях из тестового набора.

## Шаг 6: Использование обученной модели для предсказаний

---

Когда модель обучена, её можно использовать для предсказаний на новых изображениях.

```
# Предсказание на новом изображении
results = model.predict(source="path/to/your/image.jpg", conf=0.5)
results.show() # Показываем изображение с предсказанными объектами
```

### Пояснения к параметрам:

- `source` : путь к изображению или видео, на которых нужно выполнить предсказание.
- `conf` : порог уверенности, ниже которого объекты не будут отображаться (от 0 до 1).

### Сохранение результатов

Чтобы сохранить предсказанные изображения, можно воспользоваться параметром `save` :

```
results = model.predict(source="path/to/your/image.jpg", conf=0.5, save=True)
```

При этом YOLO автоматически создаст папку для сохранения изображений с предсказаниями.

## Заключение

---

С помощью этих шагов можно обучить YOLOv8 на своём собственном датасете и применять её для обнаружения объектов на новых изображениях. Если модель не достигла нужной точности, попробуйте увеличить количество эпох или оптимизировать гиперпараметры. Надеюсь, что этот процесс поможет вам в создании мощной модели для ваших задач!

Удачи в обучении YOLOv8!