

Image Models Example

Image Models Example

Файлы: `examples/image_models/train.py` и `predict.py`. Демонстрируют `YoloImageModel` на подвыборке COCO (4 изображения, `examples/image_models/data/`).

“**Проверено на локальном Docker-стеке:** оба скрипта отработывают до конца (`train.py` дообучает все 3 модели, `predict.py` строит предсказания и сохраняет визуализации). Датасет и Docker-образ были сломаны — оба фикса описаны ниже.

`train.py` — ?????????? ????
pretrained-????????? ? ??????

```
model_files = list(pretrained_dir.glob("*.pt")) #
examples/image_models/pretrained_models/*.pt

for model_path in model_files:
    base_model = YOLO(str(model_path))
    wrapper = YoloImageModel(base_model)
    wrapper.train(data=str(dataset_yaml), epochs=5, imgsz=640)
    wrapper.save(str(trained_dir / f"{model_path.stem}_trained.pt"))
```

В репозитории уже лежат три готовые pretrained-модели (`yolo11n.pt`, `yolov5nu.pt`, `yolov8n.pt` в `pretrained_models/`) — скрипт дообучает каждую на локальном мини-датасете (5 эпох) и сохраняет результат в `trained_models/`. Там же уже лежат готовые `*_trained.pt` — то есть скрипт был прогнан раньше, и его можно перезапускать повторно (перезапишет результат).

predict.py — ?????????? ?????? ????????????? ??????????

```
trained_models = list(trained_dir.glob("*.pt"))
for model_path in trained_models:
    model = YOLO(str(model_path))
    wrapper = YoloImageModel(model)
    results = wrapper.predict(str(image_path), save=False)
    results[0].plot(save=True, filename=str(out_img))
```

Берёт одно фиксированное изображение (`data/images/0000000000009.jpg`), прогоняет через каждую модель из `trained_models/` и сохраняет визуализацию в `predictions/` (`{model}_prediction.jpg`) — готовые примеры результатов уже есть в репозитории.

?????????????: ?????????????? ????? ? ????????????? + ?????????????????? ?????????????? ?????????????????

Раньше `examples/image_models/data/dataset.yaml` содержал абсолютные пути конкретной машины разработчика:

```
train: /home/niashin/mephi/mlumina/examples/image_models/data/images
val: /home/niashin/mephi/mlumina/examples/image_models/data/images
```

Заменял на `train: images / val: images` без ключа `path:`. Важный нюанс, из-за которого первая попытка (`path: .`) не сработала: ultralytics резолвит `path:` **относительно текущей рабочей директории процесса** (если такой путь физически существует — а `.` существует всегда), а не относительно самого yaml-файла. Если `path:` в датасете вовсе не задан, ultralytics использует явный fallback — директорию самого yaml-файла, независимо от CWD запускающего процесса. Поэтому правильное решение — **не указывать** `path:`, а не указывать его как `.`.

Отдельно от датасета в самом Docker-образе не хватало системных библиотек `libgl1/libglib2.0-0`, из-за чего `import cv2` (тянется через `ultralytics`) падал с `ImportError: libGL.so.1: cannot open shared object file`. Добавлено в `Dockerfile`.

??? ????????

```
python examples/image_models/train.py      # дообучение – работает из любой рабочей директории
python examples/image_models/predict.py     # инференс уже обученными моделями
```

Оба скрипта используют `YoloImageModel` в однопроцессном режиме (`train` / `predict` , не `train_ray` / `predict_ray`) — Ray не требуется и не инициализируется.

Revision #2

Created 2026-09-23 06:54:40 UTC by Маслов Алексей

Updated 2026-09-23 07:53:04 UTC by Маслов Алексей