**Step-by-step actions:**

1. Подготовил виртуальное окружение

2. Сделал файл requirements.txt - включил туда необходимые библиотеки

torch>=1.12.0

torchvision>=0.13.0

albumentations

opencv-python

tensorboard

pandas

matplotlib

3. Склонировал yolo в проект

4. Установил ffmpeg, с его помощью порезал видео на кадры -> создал датасет -> в итоге получилось 1603 изображения -> для разметки для такой простой задачи для бейзлайна слишком много - поэтому решил урезать

5. Какие классы стоит детектировать? - решил что нужно оставить только те кадры, где есть детекция блюд, посуды, чашки и т.д. Для этого можно прогнать изображения через фильтр Yolo8n предобученной на Coco. Есть более специфичные модели на HF, но для этой задачи будем использовать дефолт Yolo8n

Можно сдетектировать конечно людей, официанта и что-нибудь придумать, но задача состоит в распознавании блюд - т.ч. концентрируемся на блюдах:

Отфильтровал датасет по такой логике: брал каждый 2ой кадр, сравнивал по различию его по opencv по цвету (порог diff 8), уверенность модели Yolo8n что на кадре объекты

['bowl','cup','bottle','wine glass','fork',

'knife','spoon','pizza','cake','sandwich',

'plate', 'dining table', 'knife', 'fork', 'spoon'] = 0.25

В итоге после фильтра получилось 238 релевантных кадра

прогнал кадры через готовые модели TuyenTrungLe/ingredient-and-vietnamese-food-detection-model, ilass/OktoberfestFoodDrinkModel- получилась детекция базовых вещей, типа Spoon, Cup, Person и тд, а хочется чтобы специфичные классы были - типа шашлык, суп, салат такой-то. Можно посерчить модели на Huggingface и тд, но протестирую лишь пару моделей

поискал модели на Roboflow по тегу food, dishes - релевантных моделей нет

самый изи вариант конечно - прикрутить LLM модель на Hugging face и прогнать ее через датасет ))))

поискал модели на Hugging face по тегу food, dishes - релевантных нет

6. загрузил кадры на Roboflow - тестил авто разметку от Робофлоу, вроде нормально работает, но конфиденс низкий и путается модель, лучше самому разметить

Начинаю размечать

Можно еще сделать так - обучить Йоло на детекцию общих кропов, далее вырезать и тренировать классификатор ResNet на более детальные виды блюд. Сейчас не буду так детально делать, а просто натренирую Йоло на разные классы

Взял такие классы для разметки:

lavash

myaso\_grill

salad

salad\_grill

salad\_ovosh

soup\_2

soup\_sir

Разметил порядка 30 кадров, остальное думаю смысла нет размечать, потому что сцена одна и та же, кадры очень повторяются -> сделаю больше аугментаций, синтетики для увеличения констистентности классов

По выбору препроцессинга что выбираю:

Resize стабильный размер батча и более быстрая тренировка. Ultralytics внутри всё-равно сделает letterbox, так что лучше задать его один раз тут.

Беру аугментации следующие:

Flip (H/V)

Mosaic - p≈0.5, особенно полезен при маленьком датасете: объединяет 4 кадра, создавая «новую сервировку».

Hue / Saturation / Brightness

Exposure / CLAHE

Blur (Gaussian/Median)

Cutout / CoarseDropout

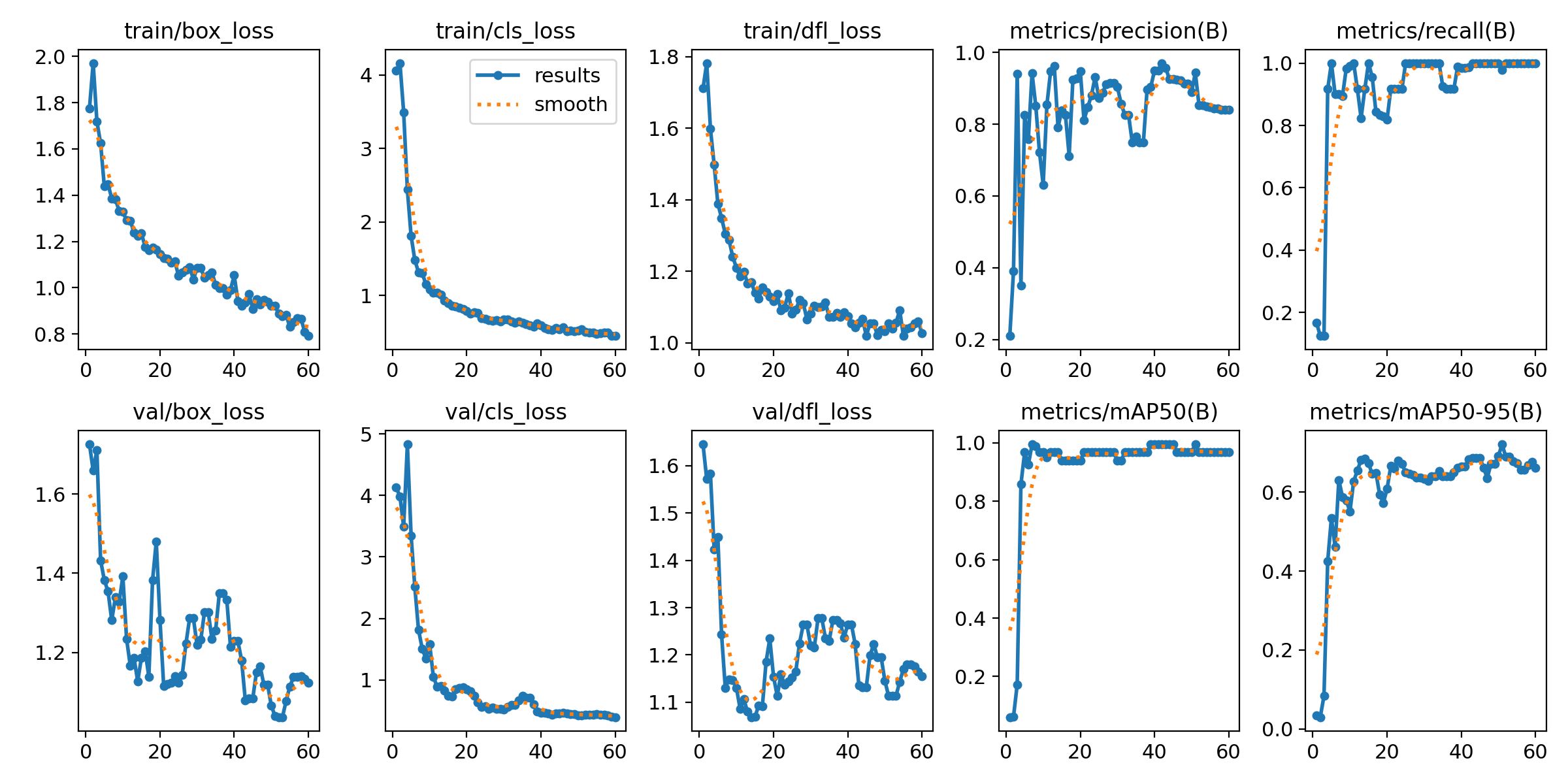
Провел обучение, настроил гиперпараметры, в 2 итерации,  
1 - lr0=0.003, mosaic=0.7, mixup=0.2

2 imgsz=768, AdamW

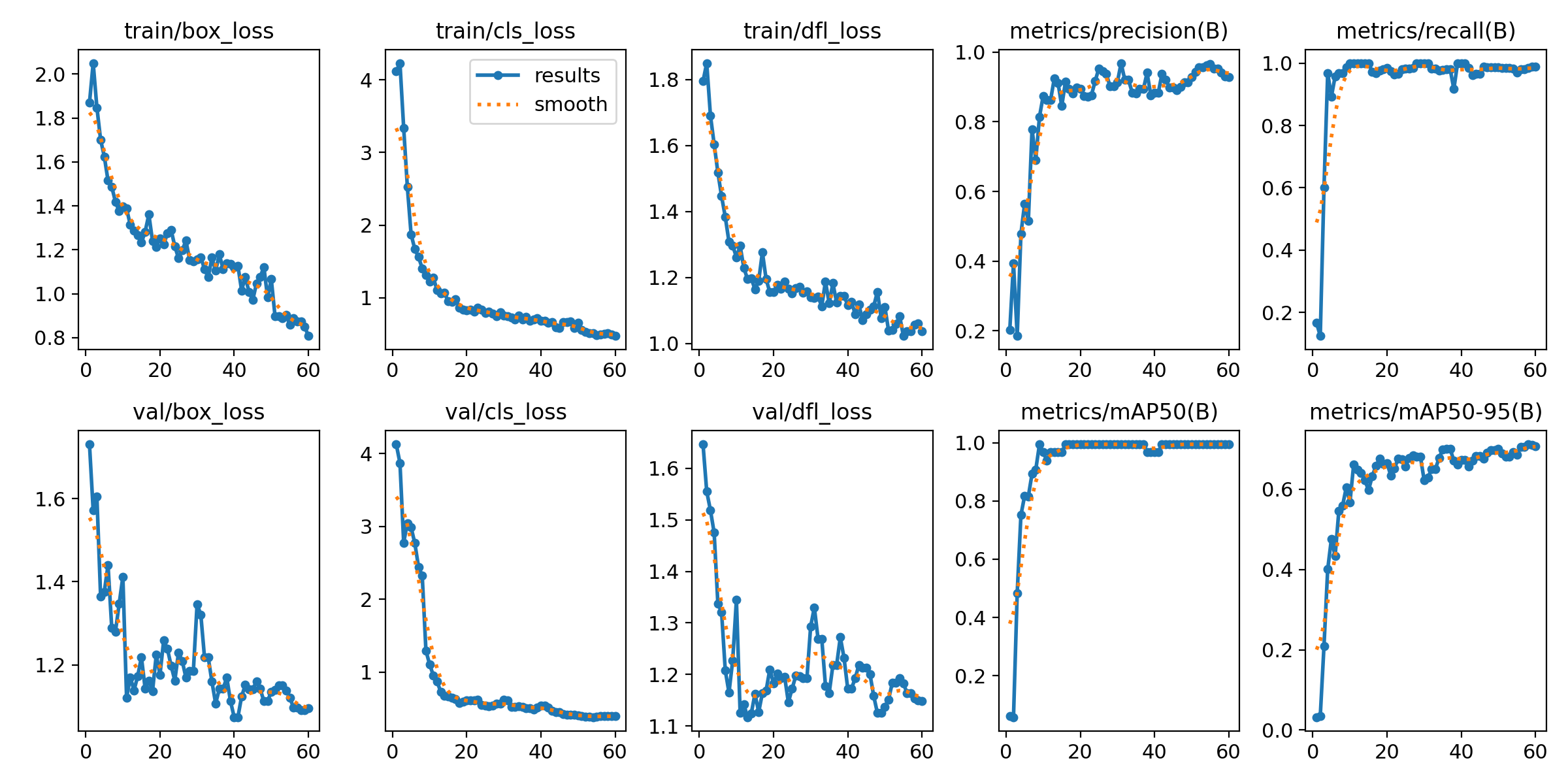
Результаты такие:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exp** | **Изменения** | **mAP@0.5-95** | **P** | **R** |
| exp00 | baseline | 00.72139 | 0.97056 | 1 |
| exp01 | lr0=0.003, mosaic=0.7, mixup=0.2 | 0.71162 | 0,92757 | 0,98827 |
| exp02 | imgsz=768, AdamW | 0.65513 | 0,9439 | 0.97709 |

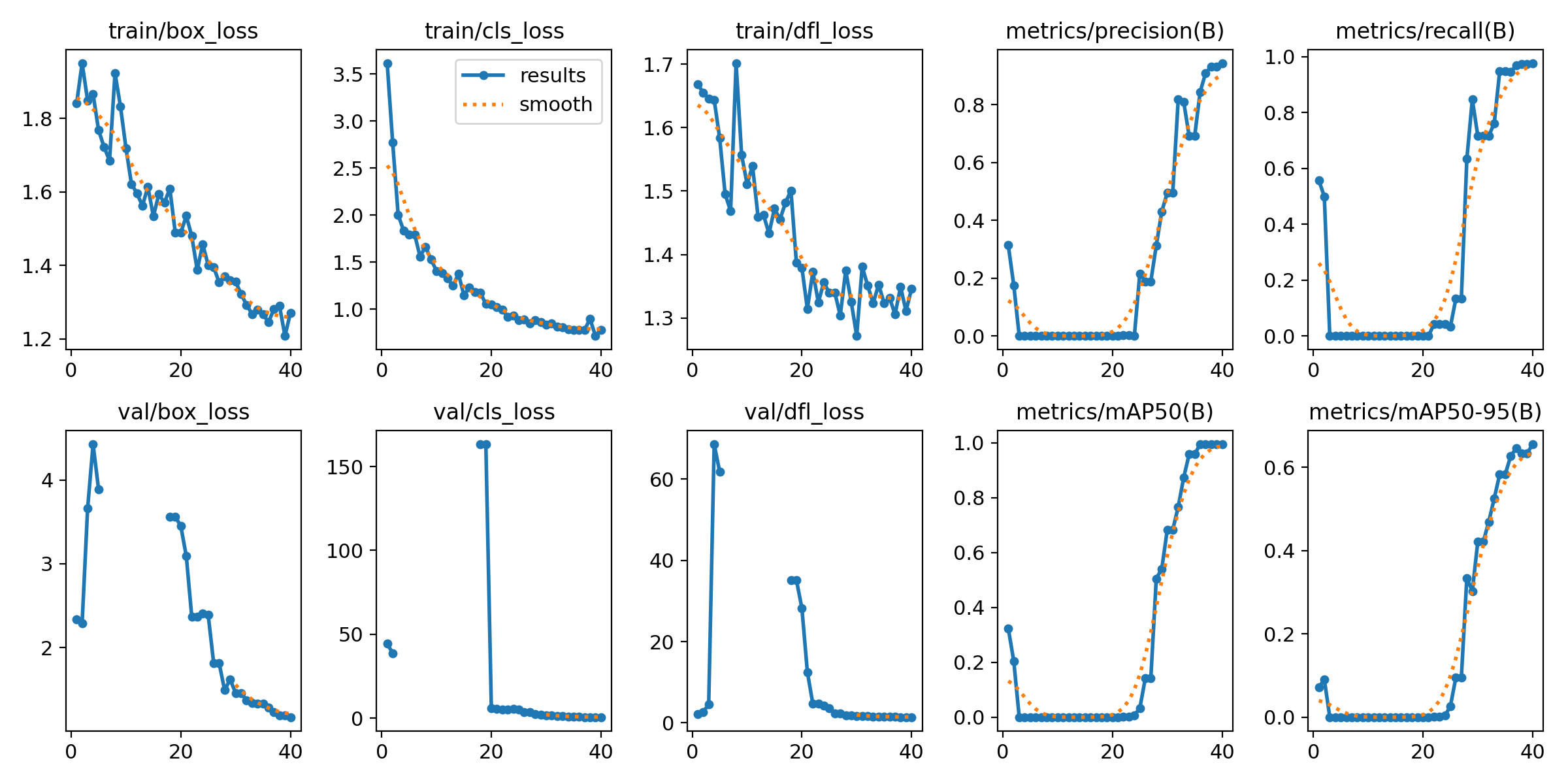
- baseline:



Exp01:



Exp02:



Из метрик следует что лучше всего себя проявила модель Baseline -> юзаем ее

Прогнал видео 3\_1.MOV через модель – результат по ссылке:

Трудозатраты

|  |  |
| --- | --- |
| Этап | Время, ч |
| Экстракция + фильтр | 1 |
| Разметка 22 | 1.4 |
| Обучение 3 эксп. | 0.9 |
| Отчёт + README | 0.8 |
| **Итого** | **3.1 ч** |

Сложности / находки

* **Малый датасет** — всего 238 кадров после фильтра, <100 боксов на класс. Решили heavy‑aug (Mosaic, CutOut)
* **Дубли кадров** — видео сняты со штатива, поэтому stride‑фильтр и diff‑фильтр, иначе модель «запоминала фон».
* **VRAM 6 GB** — YOLO11x не влез; использовали YOLO11s с batch 4 и grad‑accumulate 8.

## **Выводы и рекомендации**

* С примерно 22 размеченных кадров и лёгким hyper‑tuning получили mAP@0.5-90 ↑ 72 %.
* Heavy‑aug + меньший lr дали +5 pp к mAP без роста времени тренировки.
* Для продакшна рекомендуется собрать ещё 300–400 кадров в других условиях + разнообразие экспозиции и другое.
* Следующий шаг — заменить детекцию на детекция→ROI‑классификация для fine‑grained меню.

**Был ли у вас опыт работы с YOLO до этого задания?**

* Да, много проектов по распознаванию объектов внутри магазинов Красное и белое: Распознавание заваленности магазина, пустые полки, товары, очереди и т.д.