

Пояснительная записка

к итоговому проекту на тему:

**“Детекция простых объектов c использованием модели архитектуры yolov8 из библиотеки ultralytics при обучении на датасетах разного характера.”**

Автор: Мурашка Андрей

Группа: DLL-31

**Оглавление**

Постановка задачи для глубокого обучения

Анализ данных

Методика реализации

Итоги обучения модели

Выводы и заключение

**Постановка задачи для глубокого обучения**

Своей темы с готовым датасетом у меня не оказалось. Поэтому я решил совместить итоговую работу с практически полезным тестированием и освоением одной из самых популярных на настоящий момент библиотек компьютерного зрения Ultralytics. Я выбрал типичную задачу детекции объектов с использованием архитектуры YOLOv8, как одной из “рабочих лошадок” для решения такого типа вопросов. Распознавание объектов широко используется во всех видах человеческой деятельности. Такие задачи возникают повсеместно, например при мониторинге, уходе, сборе и сортировке урожая. Этими задачами занимаются как производители соответствующего оборудования, так и ученые и разработчики сельхоз методик и технологий. Примерами интереса к этой теме могут служить следующие две работы, найденные в свободном доступе.

Immature Green Apple Detection and Sizing in Commercial Orchards using YOLOv8 and Shape Fitting Techniques

Authors:

Ranjan Sapkota, Dawood Ahmed, Martin Churuvija and Manoj Karkee

<https://arxiv.org/pdf/2401.08629>

Authors:

Santi KUMARI Behera, Namrata Mishra, Prabira kumar Sethy, Guru Ghasidas University, Amiya Rath

<https://www.researchgate.net/publication/329190353_On-Tree_Detection_and_Counting_of_Apple_Using_Color_Thresholding_and_CHT>

Я же себе поставил менее научную задачу, используя хорошие средства автоматизации процесса обучения, валидации и распознавания, предоставляемые библиотекой Ultralytics, решить задачу с распознаванием апельсинов с реального видео не самого лучшего качества, и поэкспериментировать, каким образом обучение на разных датасетах и использование предобученных моделей разного размера будут влиять на результат детекции.

Для распознавания я выбрал 2 ролика из сети Интернет. Первый ролик с движением камеры вдоль рядов деревьев с плодами в апельсиновом саду (https://drive.google.com/file/d/1F93R9C26iYgeZOaighQr4dfcxj5DHQv4/view?usp=drive\_link). Второй ролик показывает фрагмент работы многоцелевой весовой станции (https://drive.google.com/file/d/1IYBTrBANRFGPlmmjOgC5KQDxzic6sQ-c/view?usp=drive\_link). Ролики не обладают хорошей стабилизацией и качеством изображения. Они были сняты в рекламных целях. Это добавило сложности и приблизило к “реальности”.

Моя идея состояла в том, что апельсин, как объект, является очень простым для детекции за счет стабильной и простой формы и яркого, тоже практически стабильного и оригинального, цвета. Следовательно признаки, позволяющие нейросетям детектировать такие объекты, должны быть менее требовательны к обучающему материалу, условиям съемки, освещения и масштаба изображения. Сравнение результатов распознавания после обучения моделей разных размеров на разных размеров и разного характера датасетах поможет получить представление о границах применимости данного инструмента при макетировании на бесплатном google colab’е и выработать понимание, как можно достигать оптимального результата с минимальным расходованием машинных ресурсов и времени на программирование.

**Анализ данных**

Для работы я взял три разных датасета с открытой платформы roboflow.com

1 Первый датасет (<https://universe.roboflow.com/1-xjgvf/orange-8ywhf/dataset/5>) содержит 5399 изображения размера 1920х1080.

TRAIN SET 5364 / VALID SET 19 / TEST SET 16

К данным применены приемы аугментации (по 3 на каждый экземпляр):

Flip: Horizontal, Vertical

90° Rotate: Clockwise, Counter-Clockwise, Upside Down

Rotation: Between -15° and +15°

Shear: ±15° Horizontal, ±15° Vertical

Saturation: Between -5% and +5%

Blur: Up to 3.5px

Noise: Up to 1.01% of pixels

Все изображения являются изображениями апельсинов, растущих на деревьях, прямо как на кадрах нашего целевого видео про сад. Ожидаю, что этот датасет обучит нейросеть хорошо распознавать апельсины на деревьях.

Будем называть его “основным”.

2 Второй датасет

(<https://universe.roboflow.com/mikeylamb/oranges-93gop/dataset/1>)

содержит 1128 изображения размера 640х640.

TRAIN SET 1017 / VALID SET 74 / TEST SET 37

К данным применены приемы аугментации (по 3 на каждый экземпляр):

Flip: Horizontal, Vertical

90° Rotate: Clockwise, Counter-Clockwise

Crop: 0% Minimum Zoom, 20% Maximum Zoom

Grayscale: Apply to 25% of images

Изображения представляют из себя отдельные апельсины и их группы практически без фона, или изображения отдельных апельсинов, искусственно внедренные в другие изображения, используемые вместо фона. В целом получается искусственный датасет, после обучения на котором, распознавание апельсинов на деревьях должно идти гораздо хуже, но все же, детекцию апельсинов увидеть ожидаем. Апельсины на сортировочной машине после такого обучения должны распознаваться гораздо лучше.

Будем называть его “искусственным”.

3 Третий датасет (<https://universe.roboflow.com/fwuit/orange-l4geq/dataset/1>) содержит всего 40 изображений размера 1024х720.

TRAIN SET 30 / VALID SET 9 / TEST SET 1

К данным приемы аугментации не применялись.

Все изображения являются изображениями апельсинов, растущих на деревьях, прямо как на кадрах нашего целевого видео про сад. Ожидаю, что обучения на этом датасете будет недостаточно, для хорошего распознавания апельсинов на деревьях, а апельсины на сортировочной машине не должны распознаваться совсем.

Будем называть его “малым”.

Методика реализации

Для реализации проверки моей идеи я обучил предобученную модель yolov8s на одинаковом количестве эпох для возможности сравнения результатов разных моделей. Это количество определялось доступным временем сессии в бесплатной версии колаба при обучении на самом ресурсозатратном первом датасете и составило 25 эпох. После обучения запускалась процедура его валидации. Перед обучением проводилась детекция на двух наборах изображений. На изображениях из текущего датасета и на отдельных кадрах из целевого ролика про апельсиновый сад. Это делалось для получения представления о качестве предобученной модели “из коробки”.

Для основного датасета было проведено обучение модели yolov8s и более легкой модели yolov8n. Для модели yolov8n было проведено дополнительное последовательное дообучение с шагом в 10 эпох.

После обучения модели применялись для детекции к изображениям из датасетов, к отдельным кадрам из целевых роликов и к самим этим роликам целиком.

По результатам сравнения метрик после обучения и результатов детекции делались выводы о границах применимости, причинах успехов и провалов детекции и практических выводах о важности различных аспектов подбора данных и процесса обучения для получения оптимального результата детекции.

Для оценки результатов обучения применялись метрики:

* **Пересечение над объединением (IoU):** IoU - это мера, которая количественно оценивает перекрытие между предсказанным ограничительным боксом и истинным ограничительным боксом. Она играет фундаментальную роль в оценке точности локализации объектов.
* **P (Precision)**: Точность обнаруженных объектов, показывающая, сколько обнаружений были правильными. Precision количественно определяет долю истинных положительных результатов среди всех положительных предсказаний, оценивая способность модели избегать ложных срабатываний.
* **R (Recall)**: Способность модели идентифицировать все экземпляры объектов на изображениях. Recall рассчитывает долю истинных положительных прогнозов среди всех реальных, оценивая способность модели обнаруживать все экземпляры класса.
* **F1 Score:** F1 Score - это среднее гармоническое между precision и recall, обеспечивающее сбалансированную оценку эффективности модели с учетом как ложноположительных, так и ложноотрицательных результатов.
* **Средняя точность (AP):** AP это **точность**, усредненная по всем значениям полноты между 0 и 1. **AP** интерпретируется как нахождение площади области ниже кривой **точность**-полнота.
* **Средняя средняя точность (mAP):** mAP расширяет концепцию AP, вычисляя средние значения AP по нескольким классам объектов. Это полезно в сценариях обнаружения многоклассовых объектов, чтобы дать комплексную оценку эффективности модели.
* **mAP50**: средняя точность, рассчитанная при пороге intersection over union (IoU), равном 0,50. Это показатель точности модели, учитывающей только "легкие" обнаружения.
* **mAP50-95**: среднее значение средней точности, рассчитанное при различных пороговых значениях IoU, варьирующихся от 0,50 до 0,95. Оно дает полное представление о производительности модели на разных уровнях сложности обнаружения.

Код и результаты можно посмотреть в файлах:

обучение модели yolov8s.pt на первом “основном” датасете - v1\_train\_itog.ipynb

<https://drive.google.com/file/d/1DdtCMBZbX-_8vxtEQqbQzNlUeW-vUY3c/view?usp=drive_link>

обучение модели yolov8s.pt на втором “искусственном” датасете - v1\_train2\_itog.ipynb

<https://drive.google.com/file/d/1cw5bOo_yji65E3-GHedf4ojy3zwPiIwl/view?usp=sharing>

обучение модели yolov8n.pt на третьем “малом” датасете - v1\_train3\_itog.ipynb

<https://drive.google.com/file/d/1xRp-CzYXYa-Sl-1vkATojJHe1EmTh_6T/view?usp=drive_link>

обучение модели yolov8n.pt на первом “основном” датасете - v1\_train4\_itog.ipynb

<https://drive.google.com/file/d/1DRYdfTkIotdZuS_xtHftk6-lQl8r2zbP/view?usp=drive_link>

поэтапное дообучение модели yolov8n.pt на первом “основном” датасете на следующие 10 и так далее эпох и предсказание на кадрах из ролика про сад - v2\_train5\_itog.ipynb

<https://drive.google.com/file/d/185HzJLw4fQpgLtuQ6-jvopC0T-a_SUhb/view?usp=drive_link>

детекция отдельных кадров разного разрешения и видео целиком из роликов про апельсины в саду и про сортировочную машину на моделях, обученных на 25 эпохах на разных датасетах, и ее результаты:

v2\_play\_itog .ipynb

<https://drive.google.com/file/d/10YciAyhhADOBJ86yay4E9Tv8wffCgllK/view?usp=drive_link>

v4\_play\_itog .ipynb

<https://drive.google.com/file/d/1J28FPr41tDi8NyD9z2bCyu1qyqSIPixM/view?usp=drive_link>

v5\_play\_itog .ipynb

<https://drive.google.com/file/d/1KxmrpIRfGg0fjJYAW3njPBblV7IVzW0N/view?usp=sharing>

**Итоги обучения модели**

Таблица сравнительного анализа метрик обучения:

| dataset | model | epoch | precision | recall | mAP50 | MAP50-95 |
| --- | --- | --- | --- | --- | --- | --- |
| 3 малый | yolov8n | 25 | 0.955 | 0.644 | 0.752 | 0.401 |
| 2 искусств | yolov8s | 25 | 0.917 | 0.922 | 0.97 | 0.817 |
| 1 основной | yolov8s | 25 | 0.96 | 0.618 | 0.789 | 0.429 |
| 1 основной | yolov8n | 25 | 0.877 | 0.592 | 0.762 | 0.39 |
| 1 основной | yolov8n | 35 | 0.875 | 0.682 | 0.81 | 0.337 |
| 1 основной | yolov8n | 45 | 0.912 | 0.592 | 0.774 | 0.393 |

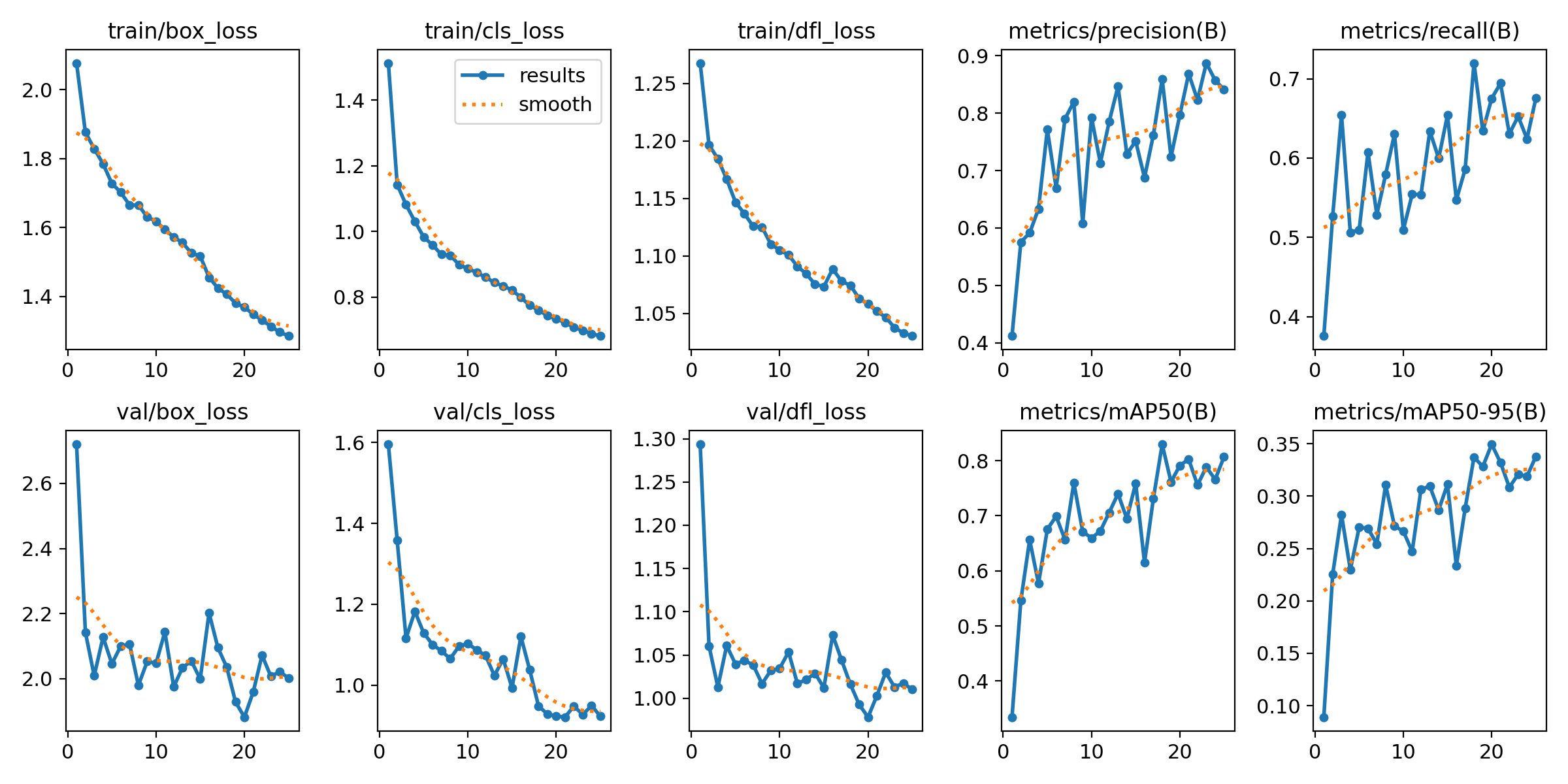
Единственная трудность при обучении - зависание сессий в колабе. При обучении модели yolov8s с 3 датасетом, вдруг не хватило ресурсов, хотя исходя из его размера он должен был требовать меньших ресурсов. Пришлось перейти на модель yolov8n.

При обучении все модели достигли за 25 эпох вполне приличного результата по метрикам, расчитанным на своих датасетах, и сохранили потенциал для дообучения. Детекция на изображениях из собственных датасетов тоже проходила хорошо.

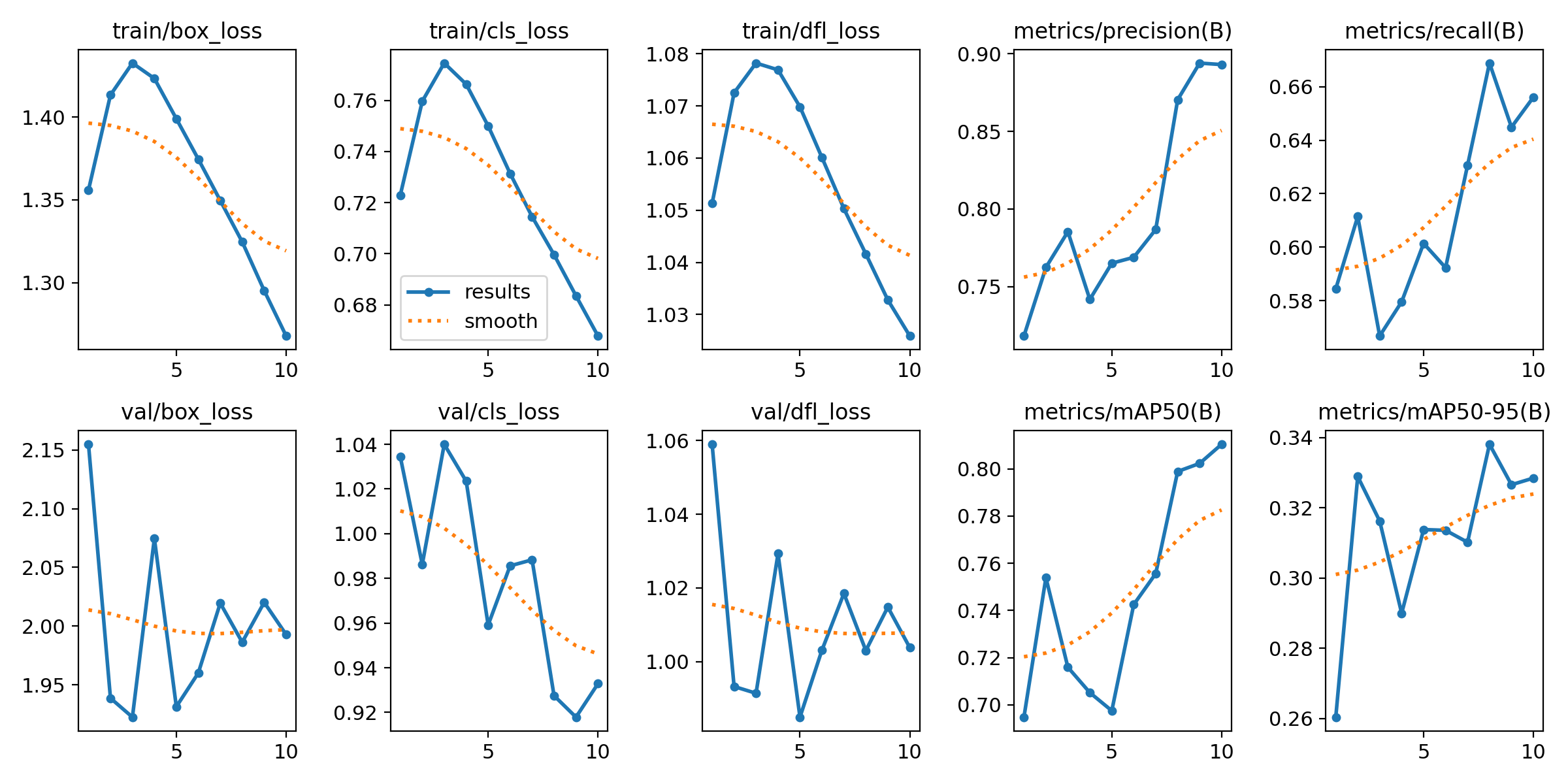
Эта модель библиотеки Ultralytics (при запуске процедуры обучения по умолчанию) не рассчитана на поэтапное обучение с сохранением промежуточных результатов, что сужает ее применимость с бесплатным колабом.

Увидеть это можно, посмотрев на графики обучения первого этапа и последующих.

графики первого этапа обучения:



графики последующих этапов обучения:



На графиках видно, что при дообучении модели с нуля значения функций потерь гладко постоянно убывают, а значения метрик гладко постоянно растут.

При повторной попытке дообучить эту же, уже дообученную модель, значения функций потерь на первых эпохах возрастают а значения метрик убывают, что показывает, что модель как бы “забывает” предыдущее дообучение, ломается и начинает обучаться “заново”.

По факту нескольких экспериментов, в результате обучения по 10 эпох, после первичного дообучения на 25 эпохах, значения функций потерь все же убывают а значения метрик растут, но очень незначительными темпами. Даже если это не случайно и справедливо, лучше дообучаться за один проход, заранее резервируя ресурсы. Возможно это поведение можно переопределить, но как , я пока не нашел.

Если смотреть на результаты детекции предобученной модели yolov8n, на глаз разница практически незаметна, а возможно дообученная на дополнительные 10 эпох модель работает и похуже обученной на 25 эпохах на некоторых фреймах.

Детекция модели yolov8n на изображении из датасета:



Детекция модели yolov8n 25 эпох на фрейме из ролика:



Детекция модели yolov8n 35 эпох на фрейме из ролика:



Примеры детектирования целевых видеороликов:

[сад\_предсказан\_без\_дообучения.avi](https://drive.google.com/file/d/1YIUjQSx0PbboG9QkPOSfs7LbMkfJmy74/view?usp=sharing)

[сад\_предсказан\_после\_дообучения.avi](https://drive.google.com/file/d/1XMlx2VvGdkG1uvvjbMTncqd7hwixg0VZ/view?usp=drive_link)

**Выводы и заключение**

Из “коробки” (без дообучения) модели способны к детекции, но процент удачной детекции на нашем материале был невысок и было большое количество ложных срабатываний, а главное, ошибок классификации. Если применять такое решение для осмысленной деятельности, следует по крайней мере фильтровать детектированные классы на предмет заведомо несуществующих на рабочем материале, например рыб на деревьях.

Увидели, что способность к детекции сильно зависит от качества видеоматериала. На кадрах из рекламных роликов, где разрешение кадров маленькое, сами объекты мелкие, быстро перемещаются и из-за этого размытые - детекция даже на хороших моделях затруднена. Увеличение разрешения детектируемых изображений практически никак или очень слабо влияет на способность моделей детектировать дополнительные объекты. Если сказать точнее - если модель нормально находит объекты, то повышение разрешения увеличивает детализацию и за счет этого модель иногда способна обнаружить дополнительные более мелкие объекты. Если модель плохо или никак не обнаруживает объекты на этом типе фреймов, то и увеличение разрешения практически не поможет.

Больше всего на конечный результат влияет качество датасета. Важно, насколько он похож на фреймы, которые модель будет опознавать в продакшене. Абстрактные признаки типа: круглый, оранжевый не позволяют модели обученной на изображениях отдельных апельсинов крупным планом или несоразмерно крупных апельсинов на абстрактном фоне мотоциклов,самолетов и т.д. , не позволяют находить соразмерные апельсины на деревьях.

Вторым по важности фактором является размер датасета и время, затраченное на дообучение модели.

С этим же фактором связано применение качественной аугментации при необходимости.

Размер модели для моего достаточно простого случая оказался не так важен.

Как можно улучшить мою работу:

1 Вместо кучи файлов написать несколько функций и потом переносом ячеек скомпоновать в одном документе для повышения наглядности и читаемости.

2 Из фреймом фильма сделать датасет наиболее близкий к теме детекции именно этого ролика, и показать, что результат детекции улучшится.

3 Найти ролик с транспортером более хорошего качества, чтобы можно было проводить более четкую детекцию объектов, и дополнить детекцию трекингом и подсчетом объектов.

4 Найти возможность обучать модель более длительное время.

Выводы на будущее:

1 Для задач этого типа важно проработать четкую установку камер с надежной фиксацией, достаточным разрешением и освещением.

2 Необходимо изготавливать датасет именно по месту применения, чтобы облегчить модели детекцию объектов за счет стабильных условий освещения, геометрии и фона и т.п.

3 В применении nano модели нет ничего страшного, наоборот, в реальных условиях производства большие вычислительные мощности не всегда доступны, и у легких моделей выше быстродействие.

4 Нельзя экономить на времени обучения.