1. Изучение датасета (видео)

На видео подается несколько блюд, после чего пара человек их съедает. Чуть позже официант уносит блюда со стола.

Задача: распознать разнообразные блюда в ресторане.

Вопросы:

а) Как следует аннотировать составные блюда? Например, бутерброд с колбасой - следует ли его аннотировать как отдельное блюдо, или отдельно аннотировать хлеб и колбасу?

б) Вид блюда меняется по мере его потребления. Полусъеденное блюдо - его следует аннотировать также как исходное? А следут ли вообще аннотировать полусъеденные блюда?

в) Одни и те же блюда могут подаваться в разной посуде, этого нет в данном датасете, но в будущем это будет крайне важным условием для составления датасета, поскольку модель начать ориентироваться на вид посуды, а не его содержимое.

г) Чай, суп и прочие однородные блюда может быть сложнее идентифицировать чем всю остальную еду, поскольку они различаются только цветом, а на цвет может влияет освещение и настройки камеры.

д) Создание универсальной модели вряд ли возможно, поскольку авторская подача блюда (форма, декор) могут полностью изменить его внешний вид, более вероятным решением будет создание модели под конкретное меню

е) Как в будущем будут обрабатываться ситуации, когда в заведении закончился какой-то ингредиент, из-за чего его внешний вид блюда изменился?

ж) Блюда, отличающиеся только начинкой (пирожки, шаурма, гамбургеры, паста с соусами), будет очень сложно отличить друг от друга.

Выводы:

Я попробую найти баланс между аннотацией крупных блюд и разбиением их на составные части. Например, набор мяса я разобью на отдельные элементы, поскольку важно, чтобы при подаче присутствовали все элементы: лаваш, овощи, мясо, соус. Кроме того, расположение элементов может варьироваться, а его потребление происходит постепенно: кусочек за кусочком. Поэтому модели проще будет идентифицировать именно отдельные части этого блюда. В то же время каждый бутерброд с салом я аннотирую как цельное блюдо – хлеба из-за сала почти не видно, кроме того, бутерброд откусывают, захватывая сразу сало и хлеб.

Я буду аннотировать блюда в их первоначальном виде при подаче, поскольку это гораздо проще, и не ясно, будет ли вообще необходимость как-то классифицировать съеденные блюда.

На самом видео блюда хорошо видны, они явно различаются. По своему опыту я могу сказать, что итоговая точность будет высокой.

Проще всего будет создавать модели для: заведений с максимально простой подачей блюд, потому что там элементы декора минимальны, присутствуют только базовые элементы, и для высококлассных заведения, потому что там строго контролируется внешний вид подачи.

2. Подготовка дасета

Извлечем по кадру в секунду - всего 320 кадров. Для этого я использую decord (https://github.com/dmlc/decord) и OpenCV. Я уже писал скрипты для подобных задач, поэтому использую один из них.

Аннотировать данные я буду с помощью сервиса https://supervisely.com.

3. Аугментация данных

Я не вижу необходимости как-то предварительно аугментировать данные – зачастую стандартный набор аугментации в YOLO от Ultralytics позволяет достичь отличных результатов.

4. Аннотация данных

В каком-то роде я решил поставить для себя эксперимент – мне было интересно узнать, сможет ли модель отличить исходное блюдо от съеденного состояния. При этом я решил считать фоном только, когда блюдо полностью съедено (пустая посуда).

Я аннотировал 67 снимков.

В датасете я выделил 15 классов:

- чайник с чаем (tea_pot);

- красный соус, не уверен – кетчуп это или аджика (red_sauce);

- красный лук (red_onion);

- маринованный чили (pickled_chili), он однозначно присутствует на левой доске, а вот на счет правой я не уверен – это он или нет, поэтому аннотировал только чили на доске слева;

- лаваш (pita);

- куриное филе (chicken_meat);

- ребрышки (ribs), объеденные ребрышки – это фон

- какое-то мясо, я предположил, что это какая-то свиная вырезка (pork_meat);

- салат с жареным сулугуни (salad);

- греческий салат (greek_salad);

- рюмка водки (vodka), почти допитую рюмки я не аннотировал, она будет относится к фону;

- хлеб с салом (snack);

- красный суп, не уверен – борщ это или солянка (red_soup);

- суп-пюре (yellow_soup);

- чашка чая (tea_cup).

5. Структурирование датасета

Сначала я выделил самые сложные на мой взгляд примеры в тестовый набор. Потом из оставшихся снимков я выделил набор максимально разнообразных снимков – это валидационные данные. Я постарался сделать валидационный набор побольше, поскольку всего снимков не очень много. Остальные снимки стали тренировочным набором. Данных не очень много, поэтому я могу распределить их вручную.

6. Обучение модели

Вообще, сначала я разметил около 25 снимков, но обученная на таком наборе модель не справлялась с некоторыми классами. Поэтому я доразметил еще снимки – до вышеуказанного количества в 67 снимков.

В ходе обучения я сначала попробовал небольшой learning rate, но модель обучалась очень медленно. Поэтому я вернулся к стандартному значению.

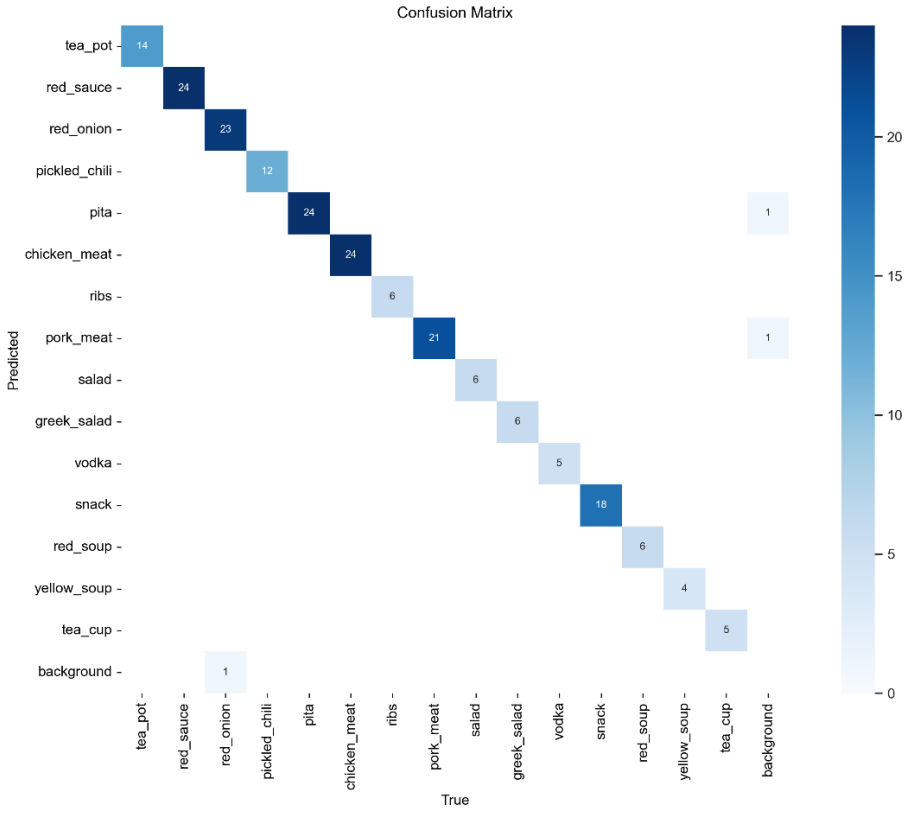
Для обучения я взял самую маленькую версию модели - yolo11n. Я попробовал yolo11s, но для ее обучения данных оказалось недостаточно.

Исходное видео имеет очень большое разрешение, поэтому я уменьшил наибольшее измерение (высоту) в 4 раза до 960 и использовал его как размер input’а у модели.

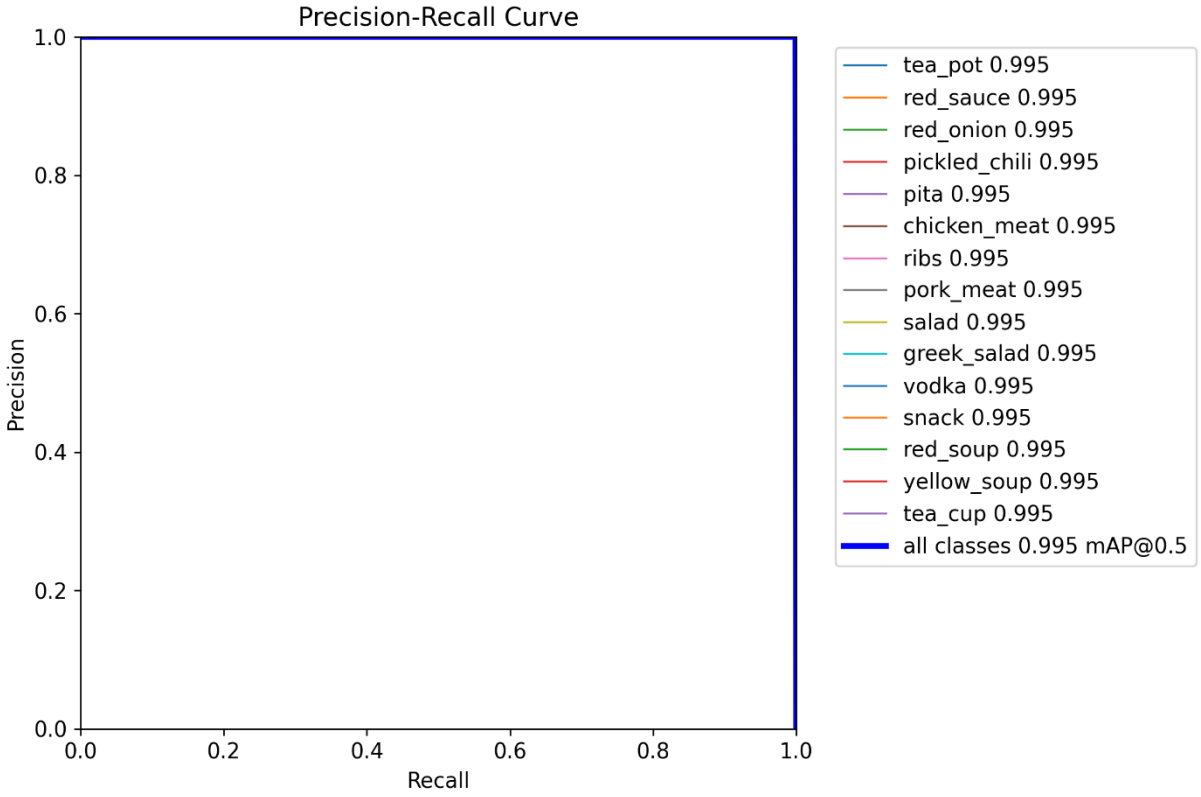
Поскольку я обучал модель на домашней видеокарте, то я сначала взял небольшой размер батча – 4. Постепенно я увеличивал его, что приводило к повышению точности модели. Наилучший результат был достигнут при размере батча 32.

В различных итерациях модель хуже справлялась с определением пустого стакана водки, путала чайник с чашками, путала пустую тарелку супа-пюре и полную. Но в последней итерации мне удалось свести эти ошибки к минимуму.

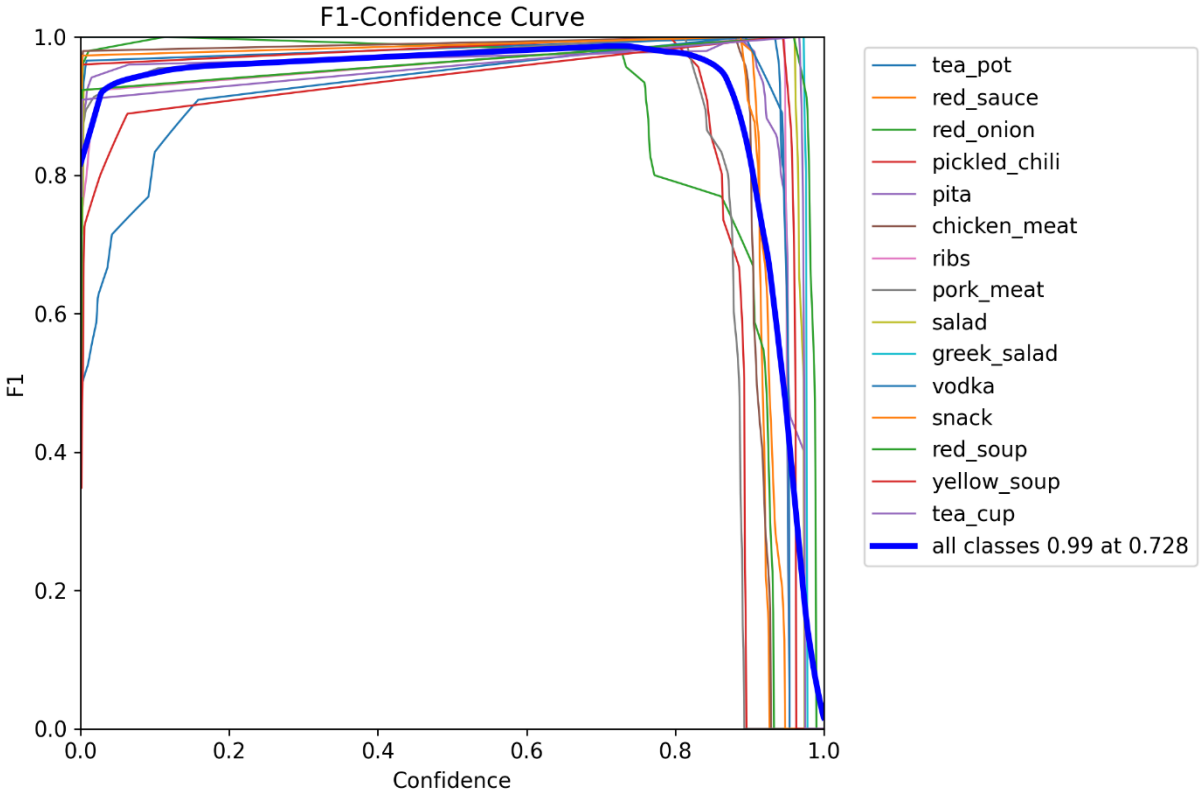
7. Результат обучения



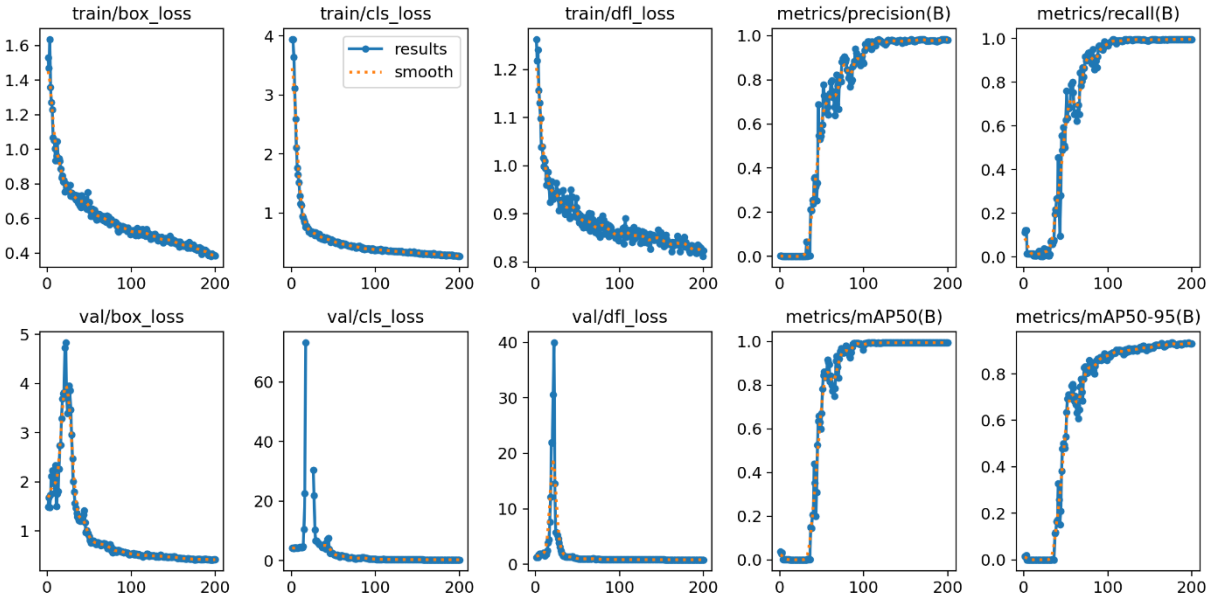
Модель отлично справляется со всеми классами, но изредка путается с красным луком (вероятно, из-за его волокнистой структуры), свиной вырезкой (ее часто перекрывают руками) и лавашем (он выглядит плоско, иногда его можно спутать с деревянной поверхностью).



Precision и Recall достигают практически 100%-го значения.



Модель имеет высокую точность (F1-score) и высокую уверенность в своих предсказаниях.



Графики, демонстрирующие изменение метрик модели в ходе обучения.

8. Что еще можно было бы сделать, если бы было больше времени

Попробовать обучить модель с меньшим размером input’а – найти оптимальный размер с минимальным снижением точности модели. Это бы снизило потребление памяти и ускорило инференс.

9. Количество потраченного времени.

Разметка: 5 часов.

Обучение: 3 часа.

Написание отчета: 2 часа.

Всего: 10 часов.