Финальный отчет

Описание задачи

Распознавание разнообразных блюд в ресторане с использованием YOLOv11, организация датасета и аналитика результатов.

Результаты

- mAP@0.5: 0.887
- mAP@0.5: 0.95: 0.81
- Precision: 0.938
- Recall: 0.895

№	Класс объекта	Изображения	Экземпляры	Точность (P)	Полнота (R)	mAP50	mAP50-95
1	Все классы	40	600	0.938	0.895	0.887	0.81
2	Столовый нож	35	54	0.984	0.296	0.379	0.342
3	Пустая чашка	40	80	1	0.915	0.994	0.92
4	Стеклянный чайник	40	40	0.99	1	0.995	0.953
5	Посетитель	21	25	0.57	0.6	0.504	0.248
6	Миска с салатом	5	10	0.973	1	0.995	0.995
7	Лаваш	21	21	0.986	1	0.995	0.995
8	Соусница	21	42	0.994	1	0.995	0.99
9	Жареные рёбрышки	21	21	0.98	1	0.995	0.985
10	Миска с супом	5	10	0.974	1	0.995	0.795
11	Столовая ложка	21	47	0.951	0.872	0.915	0.723
12	Бумажная салфетка	40	65	0.717	0.742	0.563	0.467
13	Столовая вилка	21	42	0.995	1	0.995	0.969
14	Рюмка	40	40	0.994	1	0.995	0.893
15	Смартфон	20	20	0.964	1	0.995	0.911
16	Пустая тарелка	35	83	1	0.995	0.995	0.964
17	Официант	8	15	0.92	0.896	0.888	0.8

Время выполнения

Около 23 часов

Предыдущий опыт с YOLO

Есть - обучал детектор для нахождения различных сущностей на слайде (заголовки, подзаголовки, блоки контента, схемы и тп.). Модель успешно внедрена в продакшн, за все время использования обработала более 8 млн. клиентских слайдов.

Подготовка датасета

Характеристики видео

Все видео имели частоту 30 fps. Среди изначальных видео было одно видео-дубликат - 4_1.MOV, которое представляло собой часть видео 4.MOV, поэтому кадры из него не использовались. Кадры из остальных видео были распределены между train/val/test.

Извлечение и предобработка кадров

Из каждого видео через ffmpeg были извлечены кадры с частотой 1-2 кадра в секунду в зависимости от сложности движений в видео - на выходе получились такие 347 кадров:

- 12 кадров из видео 1.mov
- 25 кадров из видео 2_1.mov
- 50 кадров из видео 3_1.mov
- 116 кадров из видео 3_2.mov
- 144 кадра из видео 4.mov

При нарезке на кадры слишком похожие не брались за счет фильтра mpdecimate. Слишком похожие кадры нам не нужны, так как это только приведет к бесполезному раздутию датасета, замедлению и переобучению.

Процесс аннотации

Первоначально рассматривался подход с zero-shot детекцией через Grounding DINO, однако при тестировании эта модель показала недостаточно качественные результаты разметки, поэтому для разметки датасета использовался CVAT, как один из самых удобных инструментов, позволяющих удобно размечать кадры видео через межкадровый трекинг (достаточно разметить первый кадр видео, и в дальнейшем нужно будет только править ббоксы движущихся объектов).
На выходе был получен датасет в формате Ultralitics YOLO Detection 1.0, готовом для старта трейна.

Классы

По соображениям логики были взяты следующие классы:

1. *table knife* - столовый нож у посетителя слева
2. *empty cup*- пустые чашки с чаем, присутствующие на всех видео

3. *glass teapot* - чайник, есть на всех видео
4. *visitor* - класс для обоих гостей
5. *bowl of salad* - оба различных салата объединены в один класс
6. *lavash flatbread* - кусок лаваша у гостя слева
7. *sauce boat* - соусники у обоих гостей
8. *roasted ribs* - жареные ребрышки у гостя справа
9. *bowl of soup* - оба различных супа объединены в один класс
10. *tablespoon* - столовые ложки в руках гостей и корзинке для приборов
11. *paper napkin* - салфетки
12. *table fork* - вилки
13. *shot glass* - стопка с водкой у гостя слева
14. *smartphone* - смартфон гостя справа
15. *empty plate* - пустые тарелки + пустое блюдце после окончания трапезы
16. *waiter* - официант, выделен в отдельный класс из-за совершенно другой роли

Мелкие объекты, такие как пакетики с сахаром, чайные ложки, чайные блюдца, кольца лука и тп. были признаны несущественными для бейзлайн-характера задачи и не выделялись отдельно.

Статистика датасета

Кросс валидация для данной задачи не подойдет, так как она случайно перемешивает кадры, что может привести к data leakage в случае смежности кадров, а так же в этом случае в каждом фолде было бы слишком мало данных для обучения.

Поэтому исходя из логики задачи было решено взять такое разделение:

- Train: 62% данных, 217 кадров
- Val: 26% данных, 90 кадров
- Test: 11% данных, 40 кадров

Каждое видео было темпорально разделено между train/val/test по следующей схеме:

Временная линия: [Train period] -> [Val period] -> [Test period],

тем самым модель тестируется на будущих данных, в трейне нет информации “из будущего”, что имитирует реальное использование модели в продакшне.

Используемые аугментации

Аугментации были подобраны для задачи детекции блюд с учётом особенностей съёмки, включая различные условия освещения и движение объектов:

Геометрические трансформации

- Поворот изображений (degrees: 8.0°) — случайный поворот изображений до ±8 градусов
- Смещение (translate: 0.15) — случайное смещение изображения до 15% от размера

- Масштабирование (scale: 0.25) — изменение масштаба изображения на ±25% для имитации различных расстояний до объекта

Геометрические искажения
- Сдвиг (shear: 2.0°) — применение сдвига до ±2 градусов
- Перспективные искажения (perspective: 0.0002) — лёгкие перспективные искажения для моделирования различных углов обзора

Отражения
- Горизонтальное отражение (fliplr: 0.5) — отражение изображения по горизонтали с вероятностью 50%
- Вертикальное отражение (flipud: 0.0) — отключено для сохранения естественной ориентации блюд

Цветовые трансформации
- Изменение цветового тона (hsv_h: 0.015) — модификация оттенка на ±1.5%
- Коррекция насыщенности (hsv_s: 0.3) — изменение насыщенности на ±30%
- Регулировка яркости (hsv_v: 0.3) — изменение яркости на ±30%

Продвинутые техники
- Мозаика (mosaic: 0.5) — объединение 4 изображений в одно с вероятностью 50% для улучшения детекции мелких объектов
- Копирование-вставка (copy_paste: 0.2) — копирование объектов между изображениями с вероятностью 20% для увеличения разнообразия сцен
- Смешивание (mixup: 0.0) — отключено для сохранения чёткости границ объектов
- Сглаживание меток (label_smoothing: 0.1) — применение сглаживания меток для улучшения обобщающей способности модели при детекции движущихся объектов

Также можно было использовать больше разнообразных аугментаций (например добавить blur, noise, dropout и тп.) из пакета albumentations, но в условиях ограниченности временных ресурсов и высоких метрик детекции уже на бейзлайне было принято решение не внедрять их на данном этапе.

Обучение

Архитектура модели

Были протестированы все доступные модели - от yolo11n до yolo11x, лучше всего себя показала самая маленькая yolo11n. Такие результаты связаны с тем, что на небольших датасетах большие модели чаще переобучаются, а сама задача с детекцией блюд не требует избыточной сложности больших моделей, так как объекты всех классов имеют относительно простые формы и четкие границы.

Параметры обучения

Через стандартный тюнинг гиперпараметров были получены следующие оптимальные их значения (полный список в конфиге репозитории)

- Эпохи: 200 (с early stopping при patience=30)
- Размер батча: 12
- Размер изображений: Мультимасштабный [608, 640, 672]
- Оптимизатор: AdamW с learning rate 0.003
- Аугментации: Специально настроены для видео данных

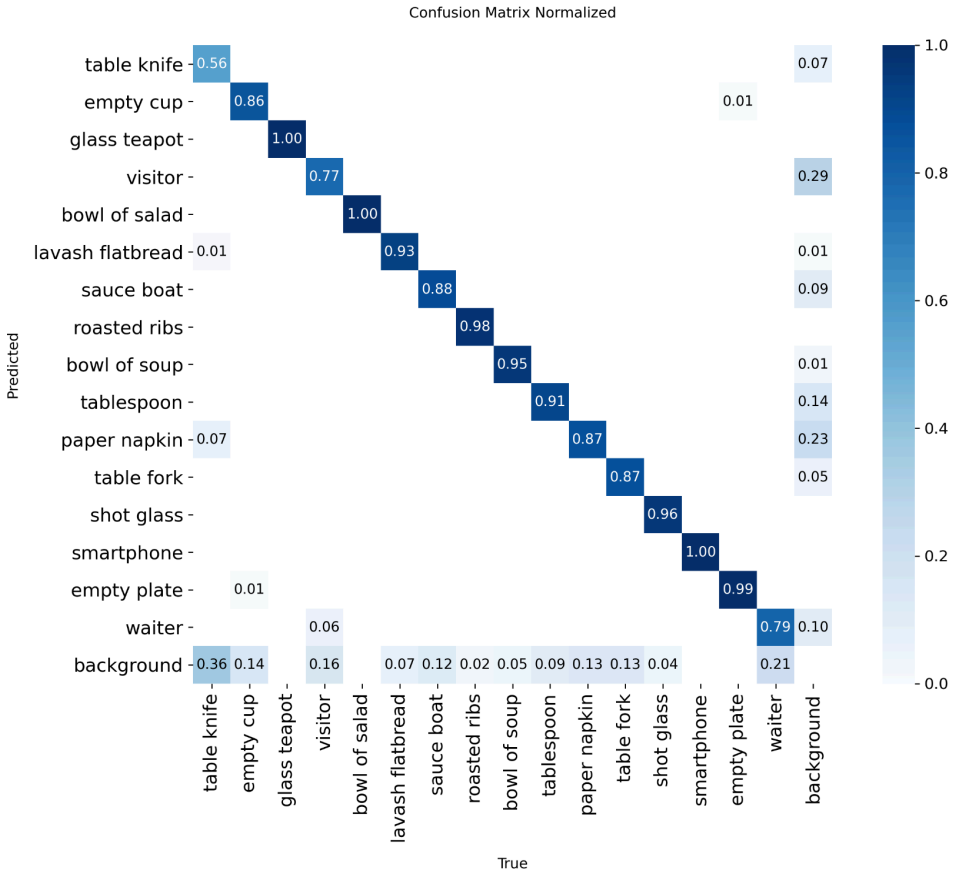
Анализ результатов

Лучше всего себя показали следующие улучшения относительно бейзлайна:

- Увеличение размера изображений (640→800)
- Расширение количества эпох (50→200)
- Введение Distribution Focal Loss для борьбы с дисбалансом классов
- Техники аугментации: многомасштабная тренировка, motion-aware обучение
- Оптимизаторы: тестирование AdamW vs SGD

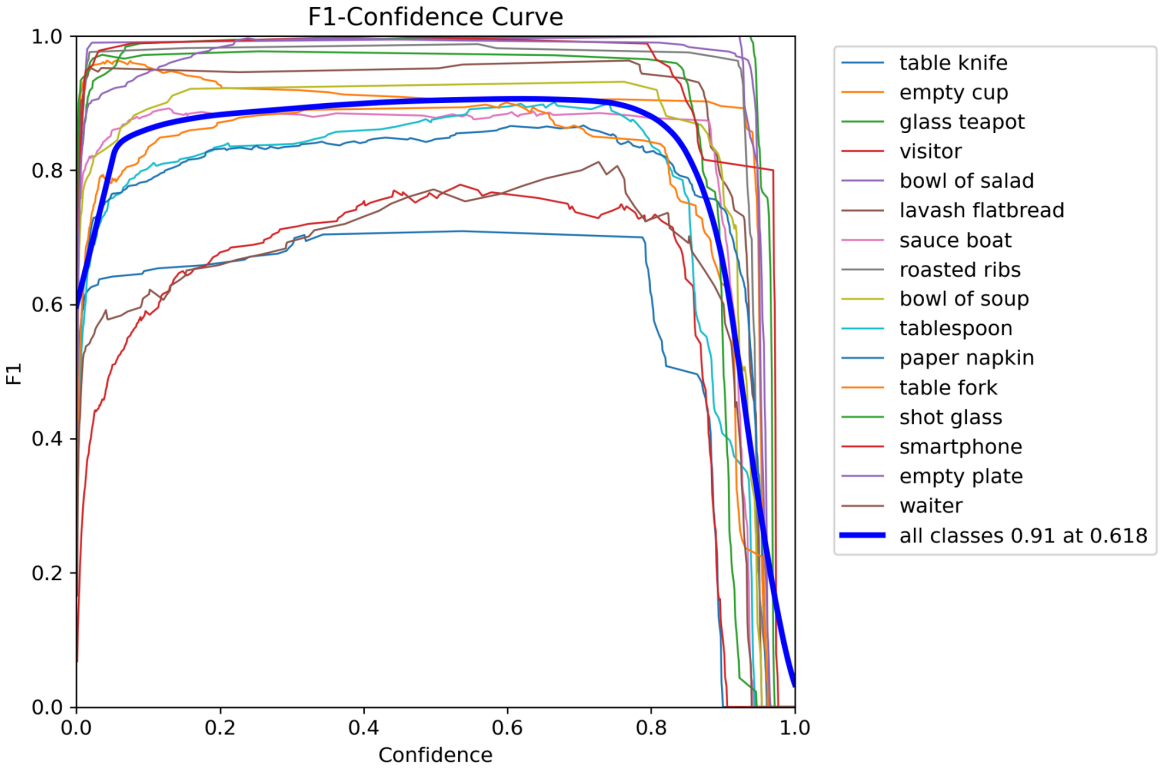
Основные графики

Ниже приведены графики самой качественной модели:
confusion_matrix_normalized



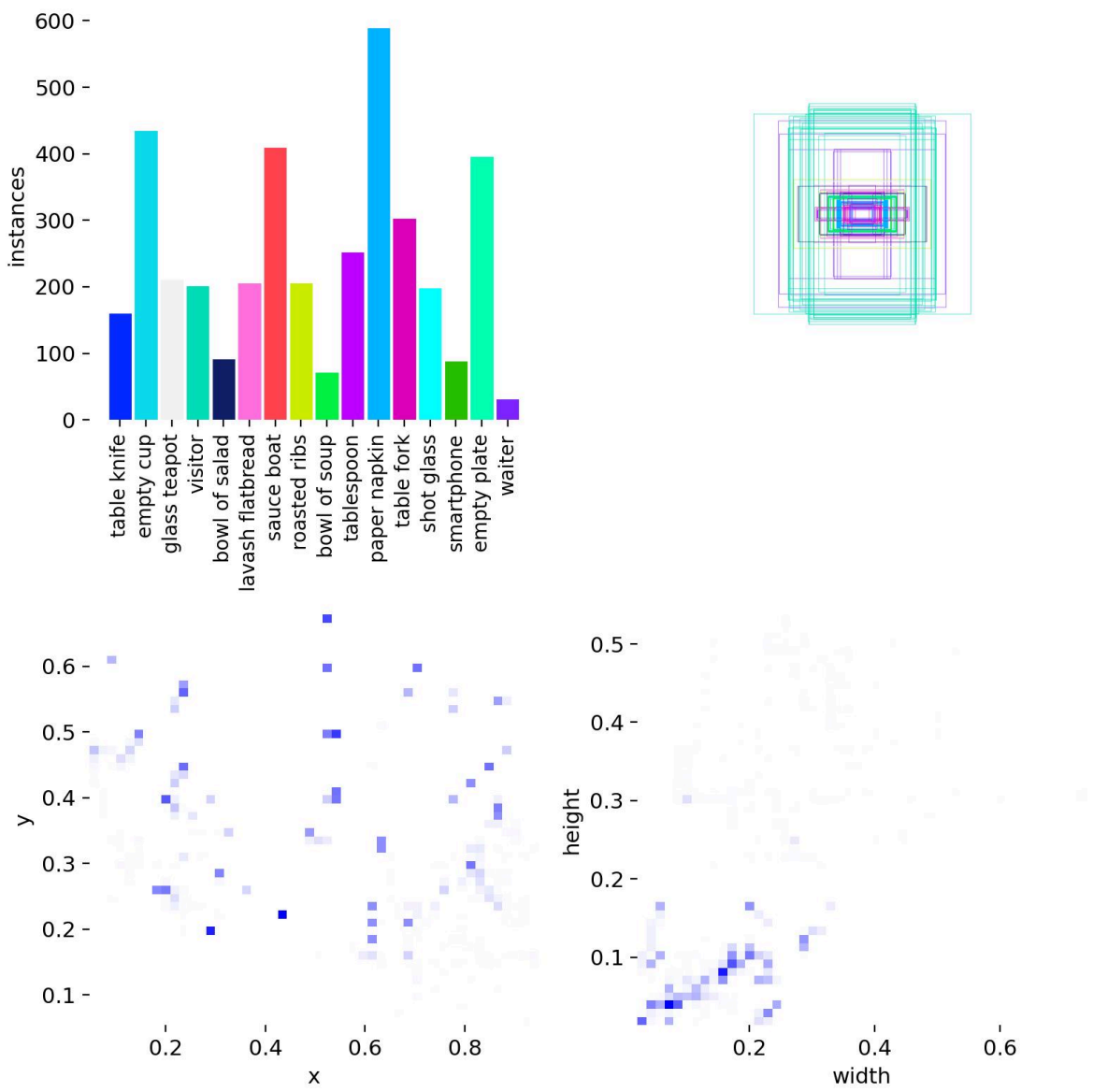
Выводы: стабильная работа с основными классами посуды и еды, при этом есть небольшая путаница между классами официант и гость, что понятно, так как люди похожи. Также есть небольшие ошибки с ложкаим, ножом, салфетками, связанные с тем, что эти классы часто меняют геометрическую конфигурацию в кадре.

F1_curve



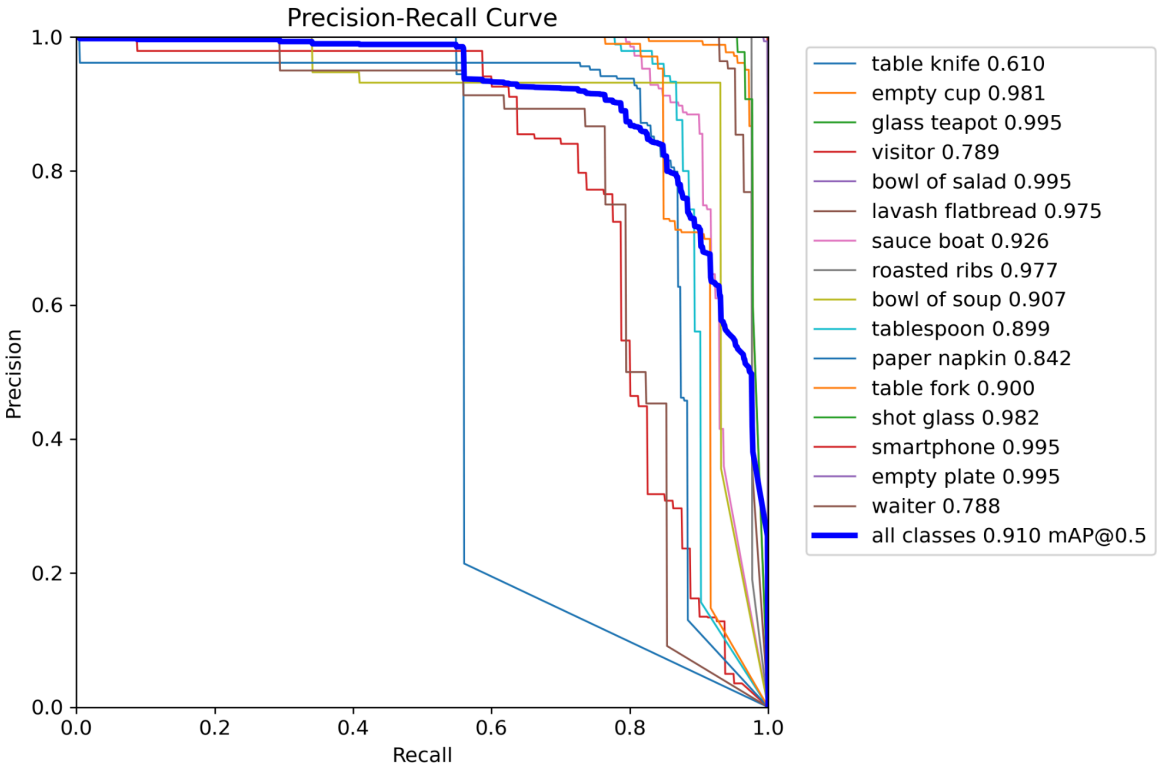
Выводы: модель в целом показывает отличные результаты, но visitor и waiter показывают значительно худшие результаты, что подтверждает выводы из confusion matrix о сложности различения людей. Для продового использования оптимальный порог confidence=0.618.

labels



Выводы: наблюдается дисбаланс классов, особенно в части кадров с официантом - это очевидно, так как он появляется в видео всего несколько раз.

PR_curve



Выводы: модель показывает стабильную работу в целом, хуже всего справляется с детекцией класса “нож”, так как он есть только в последнем видео, и его внешний вид сильно меняется относительно начала и конца видео.

Эксперименты

Бейзлайн

Для бейзлайна была взята yolo11n с размеров входа в 640 пт, 50 эпох трейна, optimizer Adam, базовые для характера задачи аугментации. У такой модели целевые метрики были такими:

- mAP@0.5: 0.84
- mAP@0.5: 0.95: 0.68
- Precision: 0.91
- Recall: 0.85

Последующие эксперименты

Далее были проведены 23 эксперимента с различными гиперпараметрами/агументациями - тюнинг экспериментов происходил по всем переменным, указанным в конфиге репозитория, для удобства графики только восьми самых важных экспериментов приведены в этом wandb-репорте. Характеристики лучшей модели приведены в начале отчета.

Инженерные вызовы

- Баланс между детализацией и практичностью классов

Объединение похожих объектов и исключение мелких объектов
- Риск переобучения на похожих кадрах из видео
  Решение: Использование фильтра mpdecimate при извлечении кадров
- Риск data leakage при случайном разделении кадров из видео
  Решение: Внедрение темпорального разделения Train→Val→Test по временной линии
- Grounding DINO показал недостаточное качество для zero-shot детекции
  Решение: Переход на ручную аннотацию через CVAT с межкадровым трекингом
- Дисбаланс классов
  Решение: Внедрение Distribution Focal Loss
- Столовые приборы меняют геометрическую конфигурацию в кадре
  Решение: Специальные аугментации для motion-aware обучения
- Столовые приборы меняют геометрическую конфигурацию в кадре
  Решение: Использование мультимасштабных изображений [608, 640, 672]

Выводы

Обученная модель эффективно детектирует заданные пул объектов, но есть очевидные возможности для улучшения качества решения задачи.

Аспекты для улучшения

- Для решения проблем с низкими метриками на редких классах для претрейна можно взять множество видео со стоков, где люди обедают в ресторанах, разметить их и тем самым сильно увеличить объем датасета и потенциальные метрики.
- Применить файнтюн сеток, обучавшихся на данных из нужного нам домена.
- Также можно добавить дополнительные аугментации из Albumentations (blur, cutout и тп.), которые могут поднять метрики
- Для кейсов с путаницей с классами можно применить двухэтапную детекцию - например, для кейса с официантом сначала искать класс "человек", затем применять классификацию