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 кадры:

- 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.

Процесс аннотации

Первоначально рассматривался подход с zero-shot детекцией через Grounding DINO, однако при тестировании эта модель показала недостаточно качественные результаты для 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)

Классы

По соображениям логики были взяты следующие классы: *table knife, empty cup, glass teapot, visitor, bowl of salad, lavash flatbread, sauce boat, roasted ribs, bowl of soup, tablespoon, paper napkin, table fork, shot glass, smartphone, empty plate, waiter*.
Мелкие объекты, такие как пакетики с сахаром или кольца лука были признаны несущественными для бейзлайн-характера задачи.

Статистика датасета

Исходя из логики задачи было решено взять такое разделение на сплиты:
62.5% / 26.0% / 11.5% (Train/Val/Test)
При этом в разных сплитах не было смежных кадров во избежание data leak-ов

Используемые аугментации

Применялся комплекс аугментаций, учитывающий специфику видео данных:
- Геометрические трансформации (поворот, сдвиг, масштаб)
- Цветовые преобразования (HSV)
- Техники мозаики и copy-paste
- Label smoothing для "мягких" границ при движении

Обучение

Были протестированы все доступные модели - от yolo11n до yolo11x, лучше всего себя показала самая маленькая yolo11n.

Параметры обучения

Через тюнинг гиперпараметров были получены следующие основные оптимальные параметры для обучения модели (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
- Аугментация: Специально настроена для видео данных

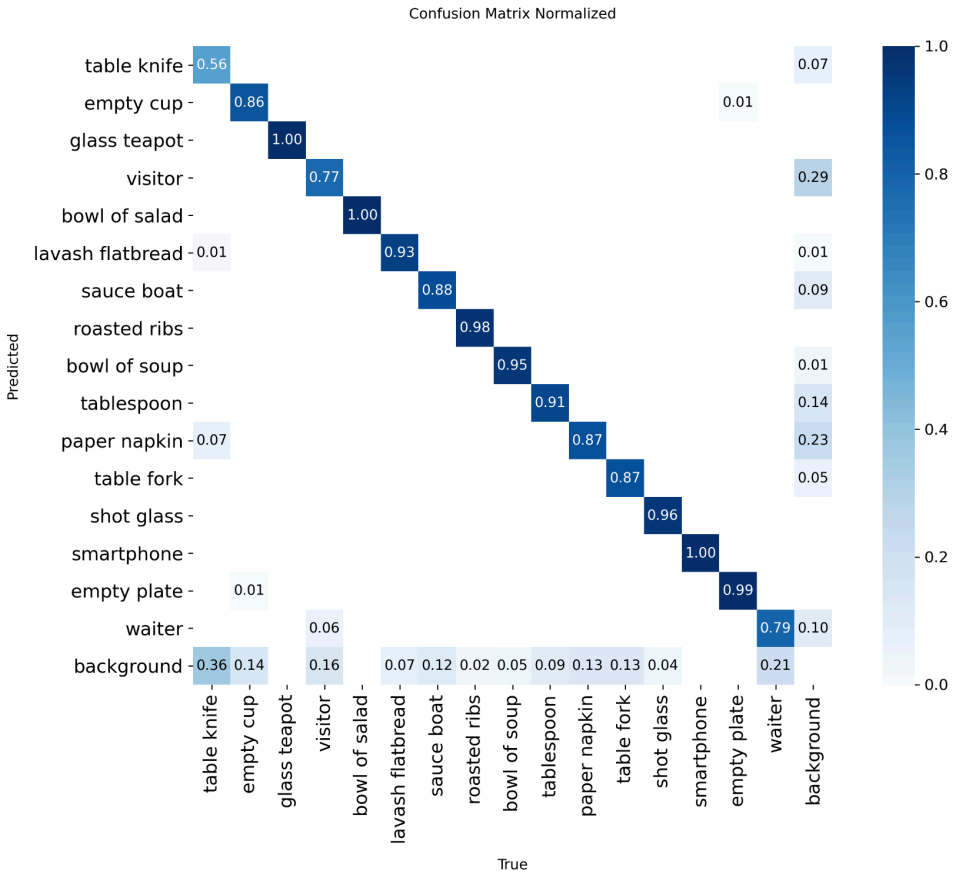
Анализ результатов

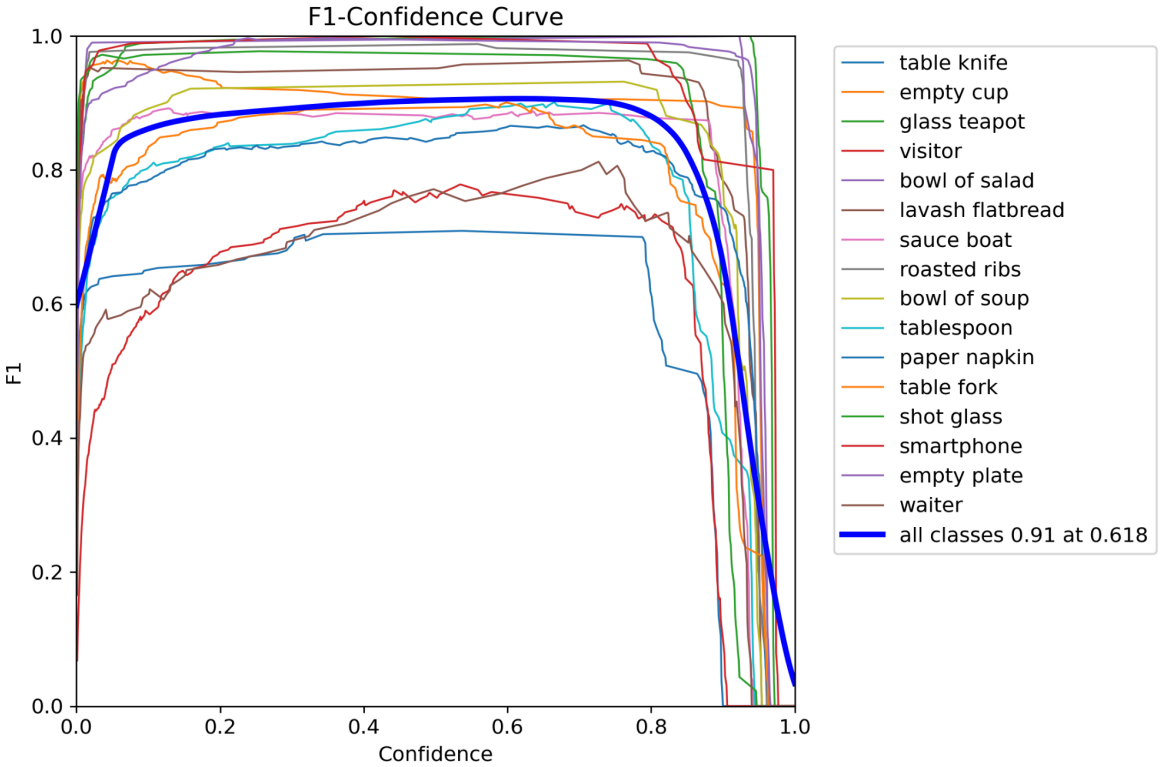
Лучше всего себя показали следующие улучшения относительно бейзлайна:

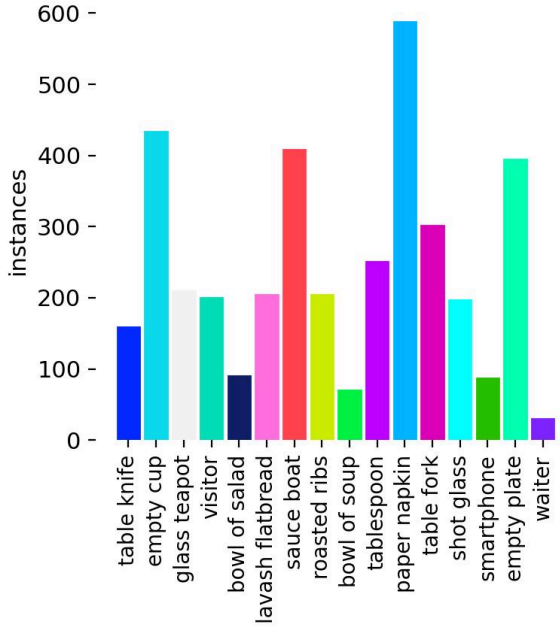
- Увеличение размера изображений (640→800)
- Расширение количества эпох (100→200)
- Введение focal loss для борьбы с дисбалансом классов
- Техники аугментации: многомасштабная тренировка, motion-aware обучение
- Оптимизаторы: тестирование AdamW vs SGD

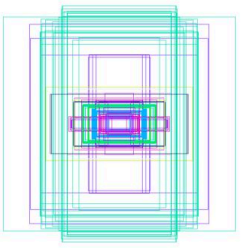
Основные графики

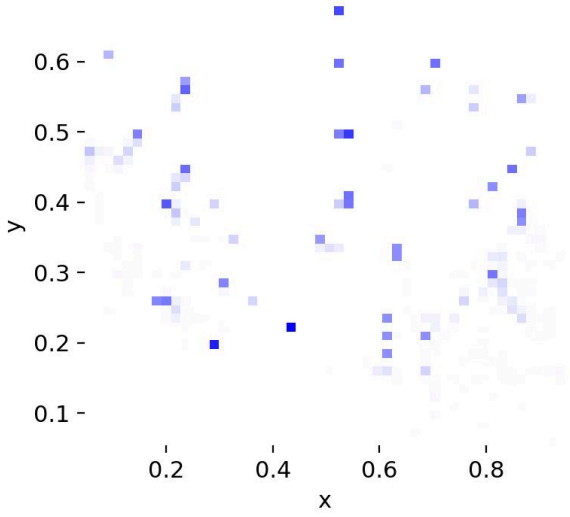
Ниже приведены графики самой качественной модели:

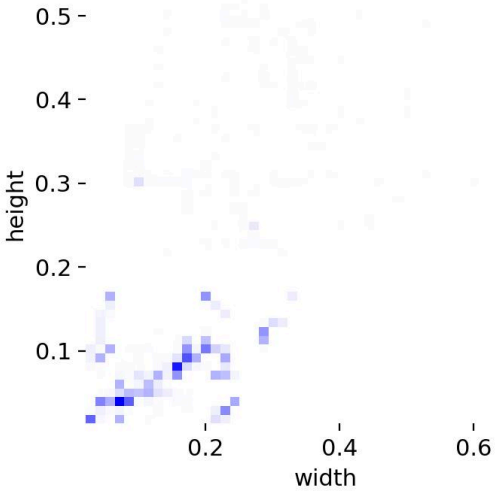

Confusion Matrix Normalized
Predicted
True
table knife
empty cup
glass teapot
visitor
bowl of salad
lavash flatbread
sauce boat
roasted ribs
bowl of soup
tablespoon
paper napkin
table fork
shot glass
smartphone
empty plate
waiter
background

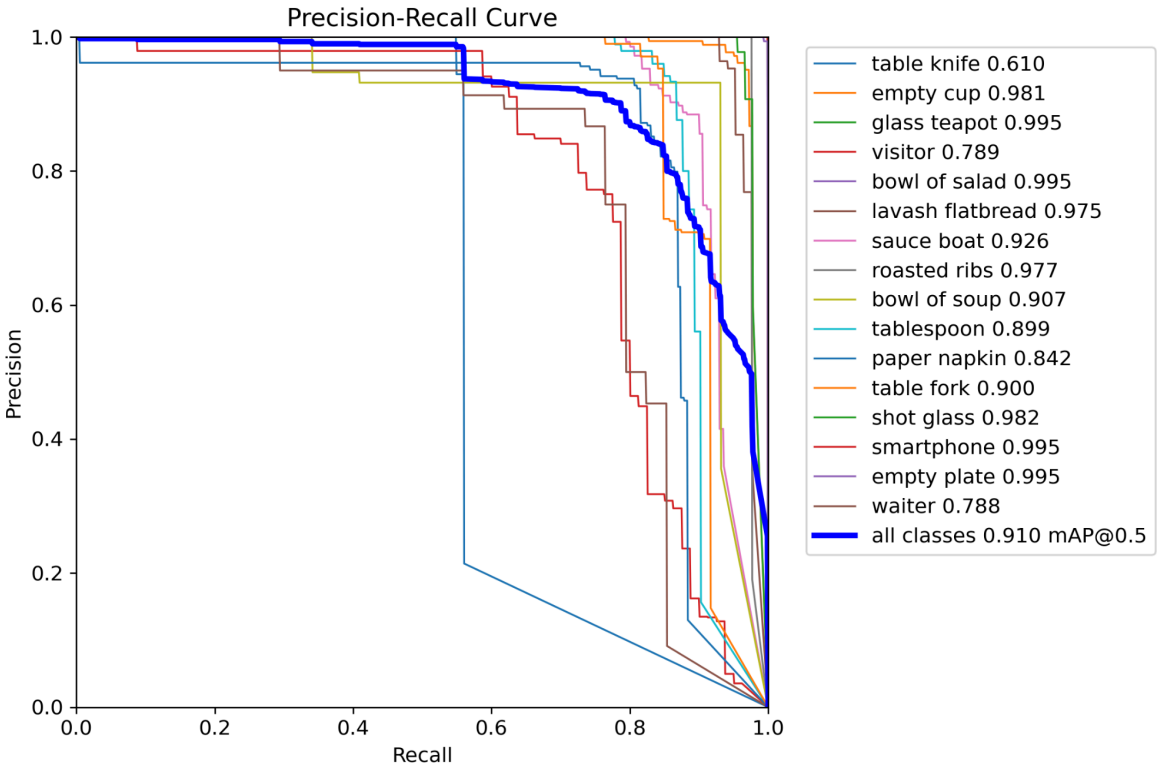

F1-Confidence Curve
F1
Confidence
table knife
empty cup
glass teapot
visitor
bowl of salad
lavash flatbread
sauce boat
roasted ribs
bowl of soup
tablespoon
paper napkin
table fork
shot glass
smartphone
empty plate
waiter
all classes 0.91 at 0.618


instances
600
500
400
300
200
100
0
table knife
empty cup
glass teapot
visitor
bowl of salad
lavash flatbread
sauce boat
roasted ribs
bowl of soup
tablespoon
paper napkin
table fork
shot glass
smartphone
empty plate
waiter




y
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.4
0.6
0.8
x


height
0.5
0.4
0.3
0.2
0.1
0.2
0.4
0.6
width



Эксперименты

Было проведено 23 эксперимента с различными гипер параметрами/агументациями - тюнинг экспериментов происходил по всем переменным, указанным в конфиге репозитория, для удобства графики самых важных экспериментов приведены в этом wandb-репорте.

Инженерные вызовы

Проблема data leakage при разделении видеоданных на train/val/test.

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. Также можно добавить дополнительные аугментации из Albumentations (blur, cutout и тп.), которые

могут поднять метрики (в текущей имплементации не использовались из-за сжатых сроков)