Отчет по выполнению тестового задания.

Целью задания было распознавание различных блюд в ресторане на видео с помощью нейронной сети YOLOv11, организация датасета и аналитика результатов.

1. Извлечение и аннотация данных

Этапы обработки данных:

- Выборка кадров из видео:
Использован редактор Roboflow для извлечения 2 кадров/сек из исходных видео. Исходный объем: 470 кадров.

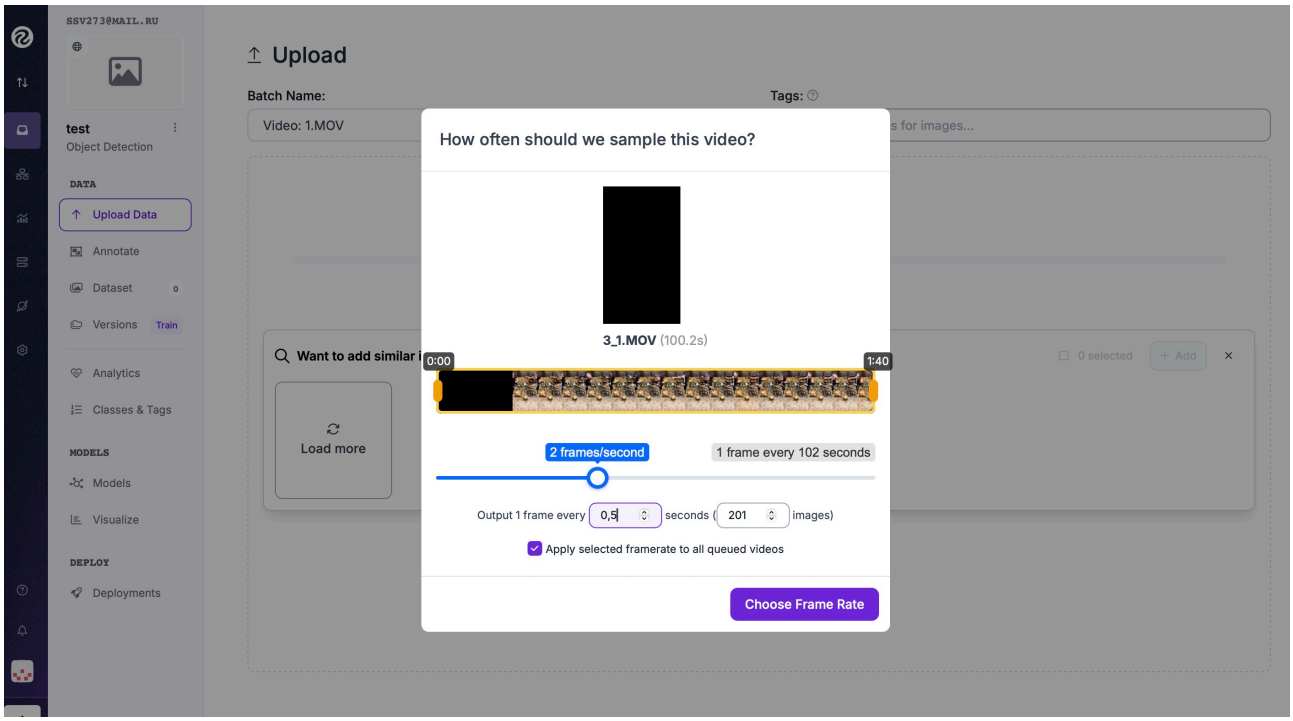

Рисунок 1: Пример исходного видео с выделенными объектами

- Аннотация и очистка:
Вручную размечены 7 классов объектов:
`чайник`, `стеклянная чашка с блюдцем`, `борщ`, `суп`, `салат`, `салат_2`, `мясо`.
Удалены дубликаты и неинформативные кадры.
Итоговый набор: 90 уникальных размеченных изображений.
При формировании датасета применена одна трансформация: ресайз до 640x640 пикселей для более быстрого обучения.

Рисунок 2: Пример аннотированного изображения в Roboflow

- Аугментация:
  Применен пакет Albumentations с трансформациями:

```
[A.HorizontalFlip(p=0.5),
   A.VerticalFlip(p=0.5),
  A.RandomBrightnessContrast(p=0.5),
  A.SquareSymmetry(p=0.5),
  A.OneOf([
      A.ToGray(p=0.1),
      A.ChannelDropout(p=0.1)
  ], p=0.2),
  A.ColorJitter(p=0.1)]
```

  Результат: 270 изображений (увеличение в 3x).

Рисунок 3: Сравнение оригинального и аугментированного изображения

- Сплит данных:
  Разделение на train/val/test в пропорции 70%/20%/10% (189/54/27 изображений).

2. Параметры обучения и обоснование

Базовая модель (baseline):
- Параметры:
  Оптимизатор AdamW(lr=0.001667, momentum=0.9)
- Обоснование:
  Стандартные параметры YOLO для начальной оценки производительности.

Model-1 (тюнинг гиперпараметров):
- Изменения:
  optimizer=Adam, lr=0.001, momentum=0.85.
- Причина:
  Adam часто улучшает сходимость на небольших датасетах, а снижение lr уменьшает риск переобучения.

Model-2 (изменение архитектуры):
- Изменения:
  Использована "тяжелая" версия YOLOv11 с глубиной сети 1.5x от базовой.
- Причина:
  Проверка гипотезы, что увеличение емкости сети улучшит распознавание объектов.

3. Графики метрик

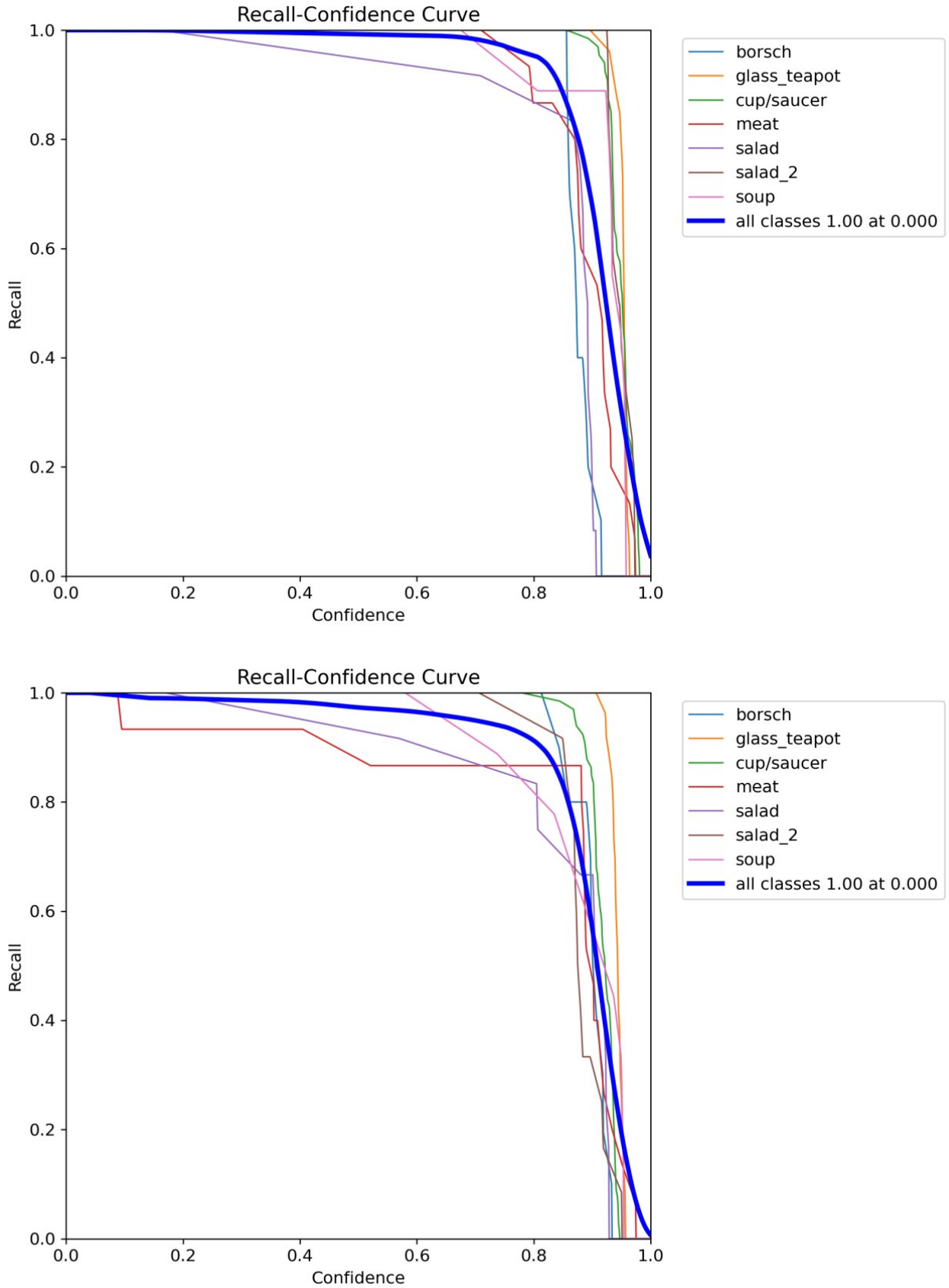

Рисунок 4: Кривые обучения для моделей (Recall).

	model	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	fitness
0	baseline	0.932741	0.987997	0.965084	0.913215	0.918402
1	model_1	0.922020	0.979027	0.972236	0.901101	0.908214
2	model_2	0.919973	0.987607	0.968981	0.899757	0.906679

- КЛЮЧЕВЫЕ НАБЛЮДЕНИЯ:
- Precision/Recall всех моделей >0.9, что указывает на высокое качество.
- Baseline показала наивысшую точность (Precision=0.93).
- Model-1 достигла лучшего mAP50 (0.972), но проигрывает в mAP50-95.
- Model-2 имеет значительно худшие показатели mAP50-95

Но, надо помнить, что я обучал всего на 10 эпохах каждую модель, и, возможно, ввиду большей глубины Model-2 не успела качественно обучиться.

Для более удобного отслеживания графиков я встроил dvclive, с помощью которого можно посмотреть все графики в удобном виде, в коде есть этот момент

4. Анализ ошибок

Типичные проблемы:

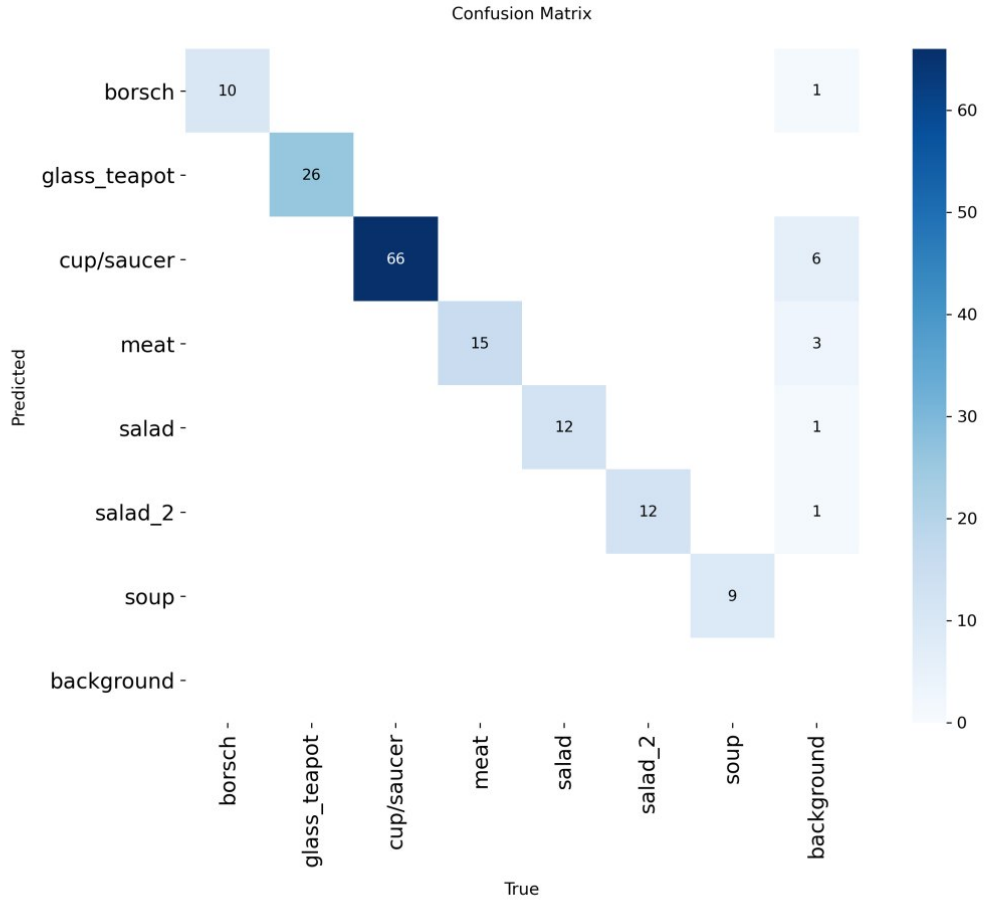

Рисунок 5: Матрица ошибок

1. Ложные срабатывания:
   - Фоновые объекты (например рюмка) ошибочно детектируются как «чашка с блюдцем".

5. Выводы и рекомендации

Результаты:
- Baseline показала лучшую сбалансированность (Fitness=0.918).
- Model-1 достигла пика по mAP50 (0.972), но уступает в обобщении (mAP50-95=0.901).
- Model-2 подтвердила: усложнение архитектуры без роста данных ведет к переобучению.

Рекомендации:
1. Улучшение датасета:
   - Добавить кадры с перекрытиями объектов (руки, посуда).
   - Дополнительно разметить рюмку.
2. Оптимизация модели:
   - Добавить аугментации, имитирующие условия ресторана (засветы, тени).
3. Эксперименты:
   - Тестирование YOLO-NAS или EfficientDet для улучшения детекции мелких объектов.
   - Кластеризация якорей под специфику посуды.

Заключение:

Текущая Model-1 пригодна для MVP (mAP50=0.972). Основное направление улучшений — расширение датасета и оптимизация аугментаций под сцены ресторана.

6. Оптимизация гиперпараметров.

Была проведена оптимизация гиперпараметров с помощью встроенного метода tune.

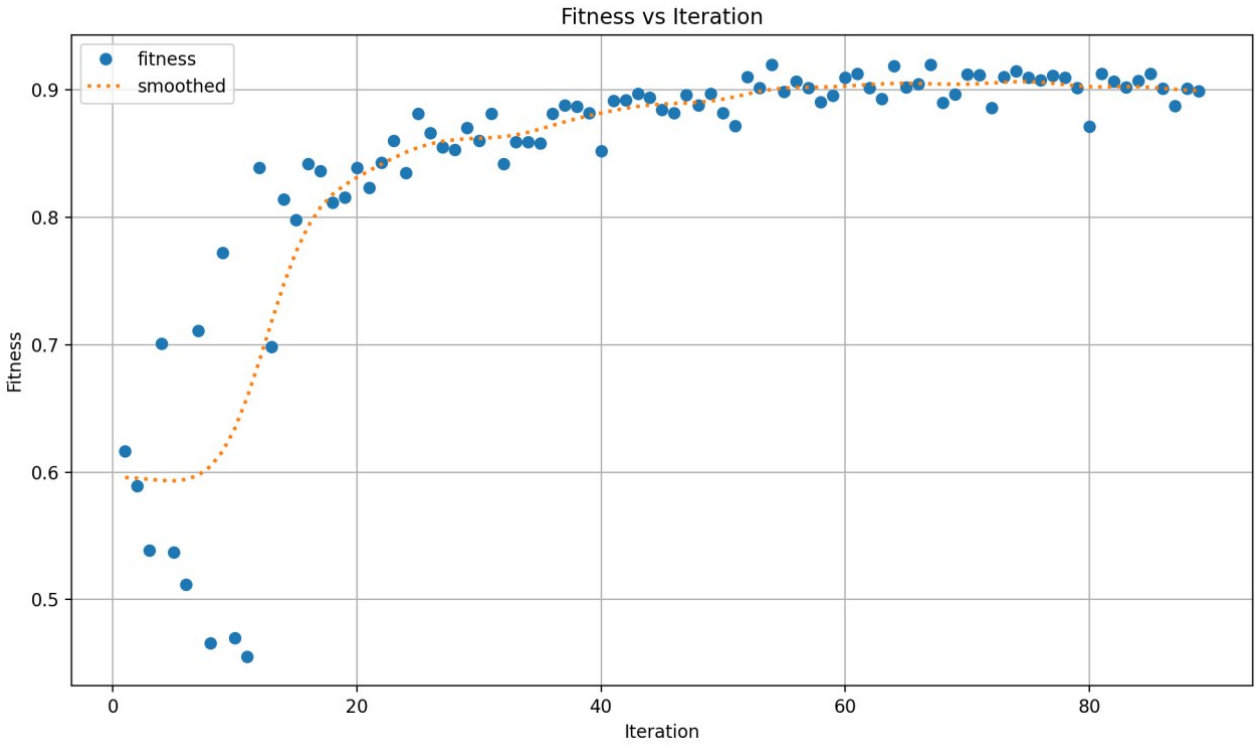

Рисунок 6: Кривая интеграционной метрики fitness при подборе гиперпараметров

Результат:

	model	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	fitness
0	baseline	0.932741	0.987997	0.965084	0.913215	0.918402
1	model_1	0.922020	0.979027	0.972236	0.901101	0.908214
2	model_2	0.919973	0.987607	0.968981	0.899757	0.906679
3	tuned_metrics_1	0.949235	0.961105	0.978213	0.930258	0.935054

После подбора гиперпараметров tuned_model_1 значительно повысила точность по сравнению с baseline (на 0.016494 в precision) и довольно близка к baseline по recall. Это говорит о том, что определенные настройки и улучшения в модели могли привести к существенному улучшению общей производительности, особенно в условиях, когда важна точность детекции, например, в приложениях, где критично не пропустить объекты.