Тестовое задание для CV-инженера

Контекст

Есть трамвай с маломощным вычислительным блоком с CPU, без GPU.
Необходимо реализовать **реалтайм-решение** для задачи компьютерного зрения (детекция объектов и оценка дистанции).
Основной акцент — умение работать с ограниченными ресурсами, логичность мышления, практические навыки и качество кода.

Этапы задания

1. Схема алгоритма (1 день)

Нарисовать или описать блок-схему (pipeline) обработки:

- входящий кадр → препроцессинг → детекция → постпроцессинг → вычисление дистанции → принятие решения системой.
  Оценим:
- логичность построения пайплайна,
- понимание ограничений железа,
- креативность (например, предложения по оптимизации).

---

2. Алгоритм и реализация (3 дня)

Реализовать прототип, который:

- выполняет **детекцию объектов** (можно взять YOLO, MobileNet-SSD или другой lightweight detector),
- оценивает **дистанцию до объекта** (любой метод: геометрия камеры, аппроксимация по размеру объекта, хак с motion-parallax и т.д.),
- накладывает результат (bounding box + дистанция) поверх видео и сохраняет/выводит его.

Оценим:

- насколько кандидат умеет использовать готовые фреймворки,

- умеет ли придумывать свои методы (в т.ч. простые и “изощренные”),
- внимание к оптимизации под CPU.

---

3. Тесты (1 день)

- **Unit-test**: проверить корректность работы детектора на одном тестовом изображении.
- **Performance-test**: измерить время инференса и убедиться, что оно **< 100 мс/кадр** на CPU.

Оценим:

- умение работать с Python-тестированием,
- дисциплину (есть ли автотесты, метрики).

---

4. README (1 день)

Написать понятное описание проекта:

- как запустить,
- какие ограничения,
- какие подходы использованы,
- какие метрики/тесты есть.

Итоговые материалы, которые ожидаются

- Схема алгоритма (pdf/jpg/png или md-файл).
- Код с реализацией пайплайна (Python).
- Демо-видео с выводом детекции и дистанции.
- Unit-тест и performance-тест.
- README.md.