

VLM или кастомный CV — чек-лист решения

Прогоните шесть вопросов, затем математику стоимости и задержки — до постройки фичи.

1. Дерево из шести вопросов (стоп на первом ясном ответе)

- ☐ Категории известны и фиксированы? НЕТ -> VLM. ДА -> следующий вопрос.
- ☐ Живое видео реального времени? ДА -> детектор (облачный VLM не уложится в ms). НЕТ -> далее.
- ☐ Больше ~100K кадров / мес? ДА -> детектор (экономика). НЕТ -> далее.
- ☐ Аудируемое авто-решение по выводу? ДА -> детектор (детерминизм). НЕТ -> далее.
- ☐ ML-команда + время на обучение и поддержку? НЕТ -> VLM (за дни). ДА -> далее.
- ☐ Нужен контекст сверх локализации объекта? ДА -> VLM. НЕТ -> детектор.

2. Математика стоимости — найдите точку безубыточности

- ☐ Стоимость VLM за SD-кадр ~ \$0.0006 (Gemini 2.5 Flash: 960x540 = 6 тайлов x 258 ток. @ \$0.30/M).
- ☐ На камеру при 5 FPS ~ 13M кадров/мес -> ~\$7 500/мес, если каждый кадр идёт в VLM.
- ☐ Кастомный детектор: большая разовая сборка (~\$8K-\$20K), затем почти ноль за кадр навсегда.
- ☐ Перелом ~ 100 000 кадров/мес: выше детектор выигрывает, и разрыв растёт.

3. Страховки по задержке и стабильности

- ☐ Поток реального времени? Детектор (~1,5 ms серверный GPU, ~25 ms edge). Облачный VLM = секунды.
- ☐ Не спрашивайте у VLM точные количества или позиции — галлюцинирует (GroundCount, OmniSpatial 2026).
- ☐ Аудируемое авто-решение? Нужен детерминированный вывод -> детектор, не вероятностный VLM.
- ☐ Если объединяете — подавайте рамки детектора в промпт VLM, чтобы он рассуждал над координатами.

4. Выкатывайте гибридную воронку (дешёвое фильтрует дорогое)

- ☐ Этап 1 — Детектор движения / крошечный детектор on-device отбрасывает ~99% кадров почти бесплатно.
- ☐ Этап 2 — Кастомный детектор + геометрия надёжно локализует объект и проверяет зону.
- ☐ Этап 3 — VLM со структурированным да/нет промптом, только на редких заслуживших кадрах.
- ☐ Ограничьте вывод VLM фикс. вариантами; для важных решений требуйте согласия по 2 кадрам.