## Шаги разработки:

### 1. Тренируем классификатор на имеющихся данных

- Можно использовать любую SOTA архитектуру для image classification (например, ResNet, EfficientNet, Swin, ConvNeXt), если данных мало — начать с небольших моделей, добавить аугментации.

- Получаем базовые метрики (accuracy, ROC-AUC, confusion matrix).

- Это будет baseline.

### 2. Берем открытую VLM модель

VLM (Vision-Language Model):

- Самые простые: OpenAI CLIP, SigLip, BLIP-2, Kosmos-2.

- Если хочется zero-shot без заморочек - бери CLIP ViT-B/16 (или ViT-L/14, если есть видеокарта).

- Вариант сложнее и новее: BLIP-2, SigLip, а также более продвинутые модели например, Qwen-VL или Qwen-VL-Plus, поддерживающие многоязычные промты и во-можность задавать человеческие вопросы.

**Что выбрать — Qwen-VL или CLIP/BLIP?**

- **Если хочешь быструю фильтрацию тысяч изображений:**
  CLIP лучше (fast, simple, батчами), но результат - только вероятность по классам. Возможно придется дообучать (такой цели пока нет, пока есть цель собрать данных)

- **Если хочется “понимающей” модели, которая сможет еще и пояснять:**
  Qwen-VL (или BLIP-2) отлично подойдет, особенно если есть разнообразные кейсы грязи и хочется автоматизировать поиск edge-case.

  Можно даже объединить подходы:

1. CLIP — быстро фильтрует очевидные clean/dirty

2. Qwen — дообрабатывает спорные или edge-case кадры

# 3. Пишем промпты и валидируем VLM на своих данных

- Важно **правильно подобрать текстовые промпты** - иногда перефразировка сильно влияет на результат.

- Попробуй несколько формулировок:

    - “camera is dirty”, “dirty camera”, “lens covered with dirt”, “obstructed lens”, “clean camera”, “clear view”, “no obstruction”

- Прогоняешь свои размеченные изображения через модель, смотришь confusion matrix, accuracy, ROC-AUC. Сравниваешь ее с классификатором который ты обучил до этого (пункт 1)

- Если плохо — подбираешь/объединяешь промпты, делаешь ensembled prompt (например, среднее по нескольким формулировкам).

# 4. Zero-shot inference на неразмеченных данных

- Когда VLM с промтом дает нормальный результат прогоняешь большой массив неразмеченных кадров через VLM с найденными промптами.

- Сохраняешь кадры с высокой вероятностью загрязнения (“dirty camera”) для дальнейшей ручной разметки (или наименее уверенные - для активного обучения).

# 5. Разметка и дальнейшее обучение

- Отдаёшь выбранные кадры на ручную разметку (сегментация — где именно грязь).

- Продолжаешь улучшать классификатор на расширенном датасете (тут размечать не надо, у тебя после выхода из vlm уже есть информация о классе грязно/чисто)

- Можно делать self supervised (но пока не будем)

# **6. Сегментация загрязнения**

- Как только накопится сегментированная разметка — обучаешь UNet/DeepLabV3/Mask2Former/PIDnet или что-то подобное исходя из

требований к инференсу.

- Разбивать изображение на тайлы и обучать сегментацию “грязь/не грязь”, как в статье.

- Можно применять augmentations, mixup/cutmix для еще лучшей генерализации.
- Тут в целом уже стандартный процесс обучения модели