Technical Design Document: Industrial Offline VLM for PCB Inspection

Name:GUDEME NIKHIL CHAND

Date: January 2026

Subject: Custom VLM Architecture for Low-Latency (<2s) Edge Inspection

---

1. Executive Summary

This document outlines the architecture for a Domain-Specific Vision Language Model (VLM) designed for offline PCB defect inspection. The system addresses the constraints of **latency (<2s)**, **localization precision**, and **hallucination mitigation** by leveraging a Small Language Model (SLM) architecture with constrained decoding and hardware-aware optimization.

(A) Model Selection: Qwen2-VL-2B (Instruct)

Choice: Qwen2-VL-2B I selected Qwen2-VL-2B over LLaVA-7B, BLIP-2, or proprietary models for three critical reasons:

1. **Edge-Optimized Size:** At 2 billion parameters, it is significantly smaller than the standard 7B models (LLaVA), making it feasible to run on edge hardware (e.g., NVIDIA Jetson Orin) within the **<2s inference** constraint without aggressive pruning that destroys accuracy.
2. **Native Resolution Support:** Unlike models that resize everything to 224x224 or 336x336, Qwen2-VL uses a **Naive Dynamic Resolution** mechanism. PCB defects (e.g., micro-cracks, missing resistors) are often tiny features in high-resolution images. Resizing leads to information loss; Qwen2-VL processes crops of the native resolution.

3. **Localization Capabilities:** The model is pre-trained with bounding box coordinates as tokens (<|box_start|>(y1,x1),(y2,x2)<|box_end|>), making it inherently better at the "where is the defect?" task compared to BLIP-2.

(B) Design Strategy & Architecture

To handle PCB-specific requirements, I propose a modified architecture focusing on high-frequency detail retention.

1. Vision Encoder Replacement

- **Standard:** OpenAI CLIP (ViT-L/14).
- **Proposed: SigLIP-SO400M (Shape-Optimized)**.
- **Reasoning:** CLIP is trained on internet captions and focuses on semantic centers (e.g., "a dog"). SigLIP (Sigmoid Loss for Language Image Pre-training) provides better dense feature extraction, which is critical for distinguishing between a "soldering pad" and a "cold solder joint."

2. Localization Head (Architecture Modification)

Standard VLMs output text. To enforce precise coordinates, I propose adding a lightweight **Detection Head (YOLO-style)** branched off the vision encoder.

- **Hybrid Approach:** The VLM provides the reasoning ("This is a missing resistor because the pads are empty"), while the parallel detection head provides the precise bounding box [x, y, w, h].
- **Fusion:** The LLM receives the image tokens *and* the proposed bounding boxes from the detection head as prompt context.

(C) Optimization for <2s Inference

Achieving <2s on edge hardware requires moving beyond standard PyTorch inference.

1. Quantization (4-bit GPTQ/AWQ)

- We will quantize the Linear layers of the LLM to **INT4** using **AWQ (Activation-aware Weight Quantization)**.

- **Impact:** Reduces memory bandwidth requirement by ~70%. On a Jetson Orin, this improves token generation speed from ~15 tokens/s to ~50 tokens/s.

2. TensorRT-LLM Compilation

- The entire pipeline (Vision Encoder + Projector + LLM) will be compiled using **NVIDIA TensorRT-LLM**.
- **Operator Fusion:** This fuses adjacent layers (e.g., Conv+ReLU) into single kernels to reduce GPU memory access overhead.

3. Key-Value (KV) Cache Quantization

- Since PCB inspection is often a single-turn QA ("Find defects"), we can use **Int8 KV Cache** to further reduce the memory footprint during the pre-fill phase.

(D) Hallucination Mitigation

Generic VLMs hallucinate because they default to "completing the sentence" based on probability. We solve this via **Constrained Decoding**.

1. Grammar-Constrained Decoding (JSON Mode)

- Instead of letting the model output free text, we enforce a **Context-Free Grammar (CFG)** on the output logits.
- The model is *forced* to output only valid JSON.
    - *Constraint:* Output structure: {"defect": string, "confidence": float, "bbox": [int, int, int, int]}
- If the model attempts to generate a token that violates this JSON schema, its probability is set to 0. This physically prevents the model from rambling.

2. "I Don't Know" Token Tuning

- During training, we explicitly include negative samples (clean PCBs).
- **Prompt:** "Find defects." -> **Target:** "No defects detected."
- This trains the model to recognize the *absence* of features, reducing false positives.

(E) Multi-Stage Training Plan

Since we have 50,000 images with boxes but *no* QA pairs, we must synthesize the data.

Stage 1: Synthetic Data Generation (The "Cold Start")

- We convert the 50,000 bounding box annotations into textual QA pairs using templates.
  - *Input:* Image + Box [100, 100, 200, 200] + Label scratch.
  - *Generated Q:* "Identify the defect at [100, 100, 200, 200]."
  - *Generated A:* "This is a scratch."
  - *Generated Q:* "Where is the scratch?"
  - *Generated A:* "The scratch is located at [100, 100, 200, 200]."
- **Volume:** This expands our dataset to ~200,000 synthetic QA pairs.

Stage 2: Vision-Language Alignment (Pre-training)

- Freeze the Vision Encoder and LLM. Train *only* the **Projector (MLP)**.
- **Goal:** Teach the LLM to "see" the PCB features (aligning visual tokens to text tokens).

Stage 3: Instruction Fine-Tuning (LoRA)

- Apply **LoRA (Low-Rank Adaptation)** to the LLM's attention layers (Target modules: q_proj, v_proj).
- Train on the synthetic QA pairs.
- **Hyperparameters:** Rank=64, Alpha=128, Learning Rate=2e-4.

(F) Validation Strategy

We cannot rely on BLEU/ROUGE scores (text metrics) for an inspection task.

1. **IoU (Intersection over Union):**
   - Measure the overlap between the VLM's predicted coordinate tokens and the ground truth box.
   - *Pass Criteria:* IoU > 0.5.
2. **POQE (Parsing-based Outcome Quality Evaluation):**
   - Since we enforce JSON output, we validate if the output is parsable. If the JSON breaks, it counts as a failure.

3. **Hallucination Rate (Negative Test):**
    - Run inference on 1,000 defect-free images.
    - Metric: Percentage of images where the model predicts a defect box. Target $< 1\%$.