Design Document

3. Custom Visual-Language Model (VLM) for Offline PCB Quality Inspection

---

1. Introduction

In semiconductor manufacturing, **Printed Circuit Board (PCB) inspection** is a critical quality assurance task. Manual inspection is time-consuming and error-prone, while generic AI models often fail due to **hallucinations and poor localization** in industrial imagery.

This document proposes a **custom offline Visual-Language Model (VLM)** that enables inspectors to ask **natural language questions** about PCB defects and receive **structured, accurate, and grounded responses** within strict latency constraints.

---

2. Problem Statement

The system must:

- Answer natural language queries about PCB defects
- Output **defect type, location (bounding boxes), and confidence**
- Operate **offline**
- Achieve **< 2 seconds inference latency**
- Avoid hallucinations common in generic VLMs

Available Data

- **50,000 PCB images**
- Each image annotated with **defect bounding boxes**

- **No question–answer (QA) pairs**

---

3. Model Selection (A)

Chosen Approach

Custom detection-aware VLM inspired by BLIP-2 architecture

Comparison with Existing VLMs

Model	Limitation in Industrial PCB Inspection
LLaVA	Large LLM, slow inference, hallucination-prone
Qwen-VL	High performance but heavy and costly for offline use
BLIP-2	Modular design, efficient, suitable for customization

Final Choice

BLIP-2–style architecture with a custom vision encoder and lightweight language decoder

---

Key Selection Factors

1. **Inference Speed**
    - Decoupled vision and language processing
2. **Fine-Tuning Flexibility**
    - Independent tuning of vision encoder, Q-Former, and LLM
3. **Licensing & Deployment**
    - More permissive for industrial offline usage

4. **Hallucination Control**
    - Enables structured and grounded outputs

Architectural Modifications for Localization

- Replace generic ViT with:
    - **Swin Transformer / ConvNeXt + Feature Pyramid Network (FPN)**
- Add:
    - Bounding box regression head
    - Defect classification head
    - Confidence estimation head
- Enforce **structured output generation** instead of free-form text

---

4. Design Strategy (B)

High-Level Architecture

```
PCB Image
   ↓
Vision Encoder (Swin / ConvNeXt + FPN)
   ↓
Defect Region Tokens (ROIs)
   ↓
Q-Former (Region-based Cross-Attention)
   ↓
Lightweight Language Decoder (1–3B)
   ↓
Structured Output (JSON)
```

Component-Level Design

4.1 Vision Encoder

- Optimized for small, texture-based defects
- Multi-scale feature extraction
- Trained using:
    - Classification loss
    - IoU / GIoU loss

4.2 Fusion Mechanism

- **Region-based cross-attention**
- Each detected defect → one visual token
- Prevents reasoning over non-existent regions

4.3 Language Decoder

- Small instruction-tuned LLM (1–3B parameters)
- Outputs machine-readable structured responses

Example:

```
{
  "defect_type": "solder_bridge",
  "location": [x1, y1, x2, y2],
  "confidence": 0.93
}
```

---

5. Optimization for Offline Deployment (C)

Model-Level Optimization

- **INT8 / INT4 quantization** for LLM
- Mixed precision inference
- **Structured pruning** of attention heads
- **Knowledge distillation** from larger teacher models
- **LoRA fine-tuning** for efficient updates

System-Level Optimization

- ROI batching
- ONNX / TensorRT deployment
- Vision feature caching where applicable

---

6. Hallucination Mitigation (D)

Root Causes

- Over-reliance on language priors
- Domain gap between web images and PCB images

Mitigation Strategies

Architectural Constraints

- Language model can only reason over detected regions

- Mandatory reference to bounding boxes

Training Techniques

- Negative sampling (“No defect present” cases)
- Contrastive grounding loss
- Explicit “unknown” responses for ambiguous inputs

Loss Functions

- Detection loss + language modeling loss
- Confidence calibration loss (ECE, Brier Score)

Output Restrictions

- Strict structured outputs
- No speculative or descriptive responses

---

7. Training Plan (E)

Stage 1: Vision Pretraining

- Train defect detector using annotated bounding boxes

Stage 2: Synthetic QA Pair Generation

- Automatically generate QA pairs using templates:
    - “How many defects are present?”
    - “Is there a solder bridge near IC U3?”
- Include hard negatives and empty-region queries

Stage 3: VLM Alignment Training

- Train Q-Former and language decoder
- Freeze most vision layers initially

Stage 4: Hallucination Stress Training

- Ambiguous and noisy images
- Occlusions and low-quality inputs
- Enforced abstention (“No defect detected”)

Data Augmentation

- Illumination changes
- Blur and noise
- Limited PCB rotation (orientation-aware)

---

8. Validation & Evaluation (F)

Localization Accuracy

- mAP @ IoU 0.5 and 0.75
- Pixel-wise bounding box error

Counting Accuracy

- Mean Absolute Error (MAE)
- Exact match accuracy

Hallucination Metrics

- False positive answer rate
- Grounding consistency score

Confidence Calibration

- Expected Calibration Error (ECE)
- Reliability curves

Deployment Validation

- End-to-end latency benchmarking
- Worst-case inference testing

---

9. Conclusion

The proposed **custom detection-aware VLM**:

- Operates fully **offline**
- Meets **sub-2 second inference latency**
- Produces **accurate, localized, and grounded responses**
- Significantly reduces hallucinations
- Scales efficiently to new PCB designs using LoRA

This design is well-suited for **industrial-quality inspection systems** where precision, reliability, and explainability are critical.