COMPLETE SYSTEM ARCHITECTURE

Rule-Based Defect Identification with Secondary Structural Verification

---

1. System Purpose and Design Philosophy

Purpose

The system is designed to **identify physical defects in semiconductor wafer/die images** at the point of inspection, using **deterministic rules** as the primary decision mechanism and a **lightweight verification stage** for ambiguous cases.

The system **does not perform image classification** in the conventional machine learning sense. Instead, it identifies **physical violations of known geometric and connectivity rules** that govern semiconductor layouts.

Core Design Principles

1. **Determinism First**
   Physical rules are enforced before any learning-based logic.
2. **Edge Feasibility**
   The system must run on low-power edge hardware without GPU dependency.
3. **Explainability**
   Every decision must be traceable to a rule violation or a measurable structural abnormality.
4. **Minimal AI Usage**
   Learning-based models are used only as secondary verifiers, never as primary decision-makers.
5. **Human-in-the-Loop Safety**
   All abnormal or uncertain cases are logged for offline human verification.

---

2. Input and Data Assumptions

Input Characteristics

- Image size: **128 × 128 pixels**
- Image type: **Grayscale (single channel)**
- Image source: Optical / SEM / inspection tool
- Pixel format: uint8 (converted to float during processing)

Structural Assumptions

- Semiconductor layouts are:
  - Highly symmetric
  - Periodic
  - Governed by strict geometric constraints
- Defects manifest as **violations**, not new objects

---

3. High-Level System Flow

```
Image Acquisition
        ↓
Preprocessing
        ↓
Step 1: Rule & Geometry-Based Defect Identification
        ↓
Decision:
   CLEAN ─────────────▶️ Accept
   DAMAGED ───────────▶️ Log
   UNCERTAIN
        ↓
Step 2: Structural Verification (Autoencoder)
        ↓
Decision:
   NORMAL ──────────▶️ Accept
   ABNORMAL ────────▶️ Log
```

---

4. Preprocessing Module

Purpose

To normalize image quality and prepare the image for deterministic analysis.

Operations

- Noise reduction (Gaussian filter)
- Contrast normalization
- Edge enhancement
- Optional binarization (for connectivity analysis)

Output

- Preprocessed image suitable for rule evaluation

Design Rationale

These operations are:

- Computationally inexpensive
- Stable across lighting variations
- Suitable for real-time execution

---

5. Step-1: Rule & Geometry-Based Defect Identification

Role in System

This is the **primary decision engine**.
It determines whether an image is:

- Structurally clean
- Structurally damaged
- Structurally uncertain

No machine learning is used at this stage.

5.1 Conceptual Basis

Semiconductor structures obey:

- Design rules
- Connectivity constraints
- Shape consistency

Any deviation from these constraints constitutes a defect.

5.2 Rule Categories and Identification Logic

A. Line-Based Defects (Bridges, Opens)

Analysis

- Edge extraction
- Connected-component labeling
- Gap and overlap detection

Identification Criteria

- **Bridge**: unintended connection between conductors
- **Open**: missing or broken conductor path

B. Crack Detection

Analysis

- Skeletonization
- Long, thin, irregular edge paths
- Orientation mismatch with design patterns

Identification Criteria

- Non-design-aligned discontinuities across regions

C. CMP Scratch Detection

Analysis

- Surface intensity gradients
- Directional streak patterns
- Texture variance

Identification Criteria

- Persistent directional surface damage

D. Line Edge Roughness (LER)

Analysis

- Edge waviness measurement
- Deviation variance along edges
- Frequency noise analysis

Identification Criteria

- Roughness exceeding tolerance bands

E. Via Defect Identification (Special Case)

Vias are treated as **geometric and connectivity primitives**, not visual textures.

Analysis Steps

1. Region of Interest (ROI) extraction
2. Contour detection
3. Shape measurements:
   - Area
   - Aspect ratio
   - Symmetry
   - Centroid alignment

Identification Criteria

- Aspect ratio deviation
- Shape asymmetry
- Area inconsistency

5.3 Step-1 Output Structure

Each image produces:

Field	Meaning
`status`	CLEAN / DAMAGED / UNCERTAIN
`violation_type`	Physical rule violated
`confidence_score`	Distance from tolerance boundary

5.4 Why UNCERTAIN Exists

UNCERTAIN indicates:

- Borderline violations
- Measurement noise
- Partial rule conflicts

This prevents over-rejecting valid dies.

---

6. Step-2: Structural Verification Using Autoencoder

Role in System

This stage **verifies structural normality** for UNCERTAIN cases.

It does **not**:

- Identify defect type
- Replace rule logic
- Make final decisions alone

6.1 Autoencoder Concept (Operational View)

- Learns how **correct structures** look
- Compresses image → latent representation
- Reconstructs image → output
- Compares reconstruction with original

Defective structures reconstruct poorly.

6.2 Training Strategy

- Training data: **clean images only**
- Objective: minimize reconstruction error
- No defect labels required

6.3 Inference Logic

1. Receive UNCERTAIN image
2. Perform reconstruction
3. Compute reconstruction error
4. Compare against threshold

6.4 Step-2 Output

Field	Meaning
`reconstruction_error`	Structural deviation score
`structural_status`	NORMAL / ABNORMAL
`error_map` (optional)	Localization of abnormality

---

7. Logging & Human Review Module

Purpose

To ensure:

- Traceability
- Safety
- Process improvement

What Is Logged

- Image identifier
- Rule violation (if any)
- Rule confidence
- Autoencoder error
- Error localization

Key Safety Constraint

- ❌ No online learning
- ❌ No automatic rule updates

All updates occur **offline after human review**.

---

8. Hardware Architecture

Target Platform

- ARM Cortex-A53 / A55 class processor
- 1–2 GB RAM

- No GPU
- SIMD support (NEON)

Examples:

- NXP i.MX RT series
- NXP i.MX 8M series

Compute Requirements

Component	Complexity
Rule engine	< 5 MFLOPs/image
Autoencoder	~1–2M ops/image
Memory footprint	< 10 MB

---

9. Performance Characteristics

Latency (128×128 image)

Stage	Time
Rule identification	0.5–1.5 ms
Autoencoder verification	3–6 ms
Typical total	~1 ms
Worst-case total	~7 ms

Throughput

- ~1000 images/sec (rule-only)
- ~150–250 images/sec (with verification)

---

10. Why This Architecture Works

- Aligns with semiconductor physics
- Deterministic primary decisions
- AI used only where ambiguity exists
- Edge-deployable
- Explainable and auditable
- Scales to real manufacturing throughput

---

11. Final System Interpretation

This system should be understood as:

> **A defect identification pipeline that enforces physical design rules and verifies structural integrity using lightweight learning, rather than a conventional image classification system.**
