Title : Edge AI-Driven Semiconductor Defect Classification

Team Details

Team Name: Die-Trying

SR. NO	ROLE	NAME	ACADEMIC YEAR
1	**Team Leader**	Aditya Malpani	3rd year B.Tech
2	Member 1	Yash Dobariya	3rd year B.Tech
3	Member 2	Krins Italiya	3rd year B.Tech
4	Member 3	Pratyush Maheshwari	3rd year B.Tech

COLLEGE NAME

Pandit Deendayal Energy University Gandhinagar Gujarat

TEAM LEADER CONTACT NUMBER

+ 91 6352589940

TEAM LEADER EMAIL ADDRESS

adityamalpani006@gmail.com

Problem Statement Addressed

Edge AI–Based Defect Classification System for Semiconductor Wafer and Die Images

High-Volume Inspection Challenge
Modern semiconductor fabs generate massive volumes of die inspection images, where even microscopic defects can significantly impact **yield, performance, and long-term reliability**.

Limitations of Conventional Inspection
Centralized and manual inspection pipelines suffer from **high latency, bandwidth bottlenecks, rising infrastructure costs, and limited scalability**, making them unsuitable for real-time, high-throughput production.

Need for Edge AI–Driven Inspection
There is a critical need for **real-time, on-device defect detection and classification**, enabling **low-latency, scalable, and Industry-4.0-ready semiconductor manufacturing**.

Idea Description

➢ IDEA SUMMARY :

- Develop a **custom, semiconductor-specific wafer and die defect dataset** to address the lack of reliable public inspection data.
- Focus on **realistic defect classes and imaging conditions**, ensuring the data reflects actual inspection scenarios.
- Train a **lightweight CNN baseline (MobileNetV2)** to learn domain-relevant defect patterns effectively.
- Validate **Phase-1 feasibility** through model accuracy, class-wise performance, and compact model size suitable for edge execution.

➢ KEY CONCEPT & APPROACH :

- **Edge-first AI architecture** that brings defect intelligence closer to wafer and die inspection, minimizing dependence on centralized analysis.
- **Constraint-driven model design** where accuracy, latency, and memory are co-optimized for edge feasibility from the outset.
- **Scalable inspection strategy** designed for consistent use across multiple inspection points in high-throughput fabs.

Proposed Solution

SOLUTION DETAILS :

Methodology

• Design, train, and validate a defect classification model using wafer and die inspection images, with emphasis on robustness across defect categories and imaging variations.

Technology

• Adopt MobileNetV2 as the core architecture for its strong accuracy–efficiency balance, supported by task-specific preprocessing and lightweight model optimization.

Implementation Strategy

• Establish a **model-development workflow** where training and validation are performed on standard compute systems, with architectural decisions explicitly aligned to edge execution constraints to ensure smooth transition to on-device inference in later phases.

IMPLEMENTATION STRATEGY :

SEM / Die Images → Dataset Prep → Preprocessing → MobileNetV2 → Edge Inference

DATASET PLAN & CLASS DESIGN

- Dataset Overview
- Total images current : 22,672
- No. of classes: 8 (6 defect + Clean + Other)
- Class balance plan: 2500 Images Per Class
- Train/Validate/Test split: 70 % / 20% / 10%
- Image type: Grayscale
- Labeling method/source: Generated / Python

- Class Design (8 Classes) :
  - Defect_1 – Open
  - Defect_2 – Via
  - Defect_3 – Crack
  - Defect_4 – LER
  - Defect_5 – CMP
  - Defect_6 – Bridge
  - Clean
  - Other

Baseline Model & Results

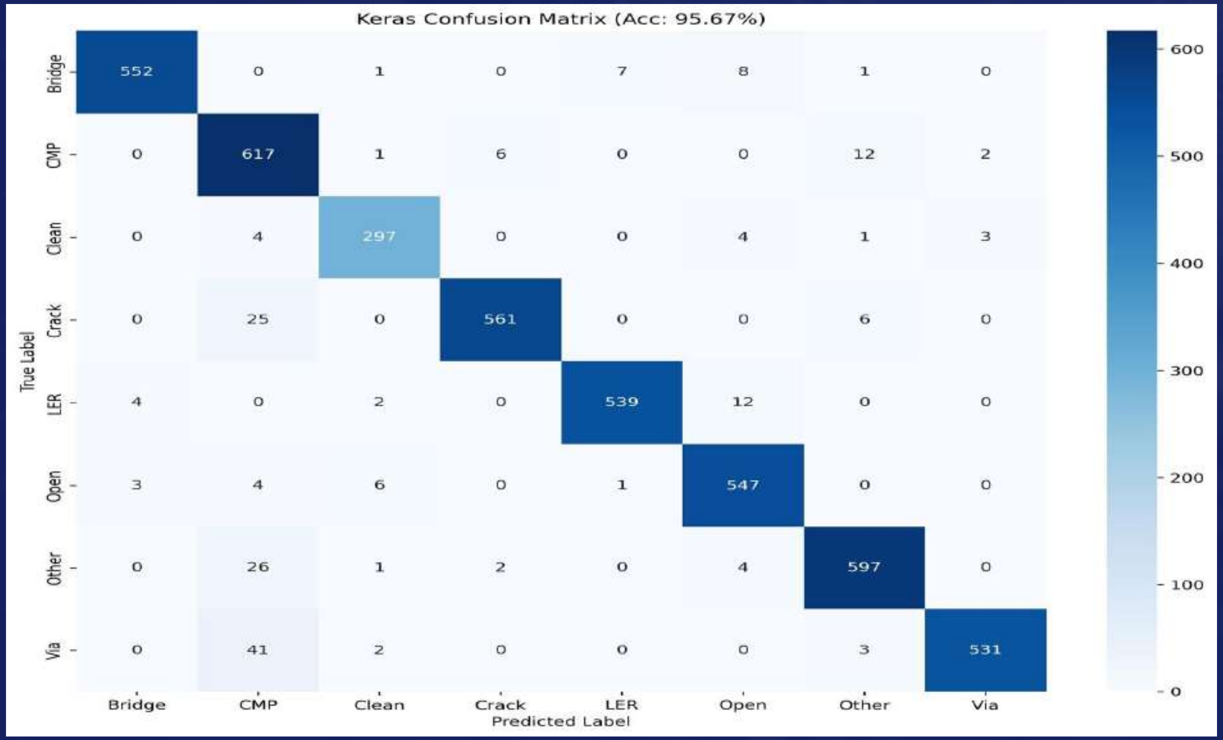
➢ Model details :

- Architecture : MobileNetV2
- Training approach : Transfer Learning
- Input size : 224×224
- Model size : 5 MB
- Framework : TensorFlow 2.x [Keras API]

➢ Metrics on your test split :

- Accuracy : 96%
- Precision/Recall : 96% / 96%

➢ Confusion Matrix:



GitHub & Video Link

GitHub Repository	https://github.com/yash4959/Die-Trying
Dataset ZIP link: (Drive)	https://drive.google.com/file/d/1aESAhJMB3ur3kpgqoUetGvQf6l2wuAiW/view
ONNX model link:	https://drive.google.com/file/d/1qF1G4cor4Z5GbD9xmd6pGoq5miGSplLI/view
Results report link :	https://drive.google.com/drive/folders/17m2kinjPerkSyWzLjhlzqza4CHVz_CmX
Simulation Video :	https://drive.google.com/file/d/1xw8oAEsZ2deSmF13DgoUed1c7Vpq72vX/view