Integration and Comparison of Vision Models for Smart Inspection Cell

Case Study

Intelligent Systems in Production

Technische Hochschule Deggendorf

Campus Cham

Date : 06.11.2025

Group members:

Sai Prasanth Parnambedu

Yaswitha Pallela (Mt no : 12504195)

Rakshith Thatikonda

Prarthana Shenoy (Mt no : 12504810)

Chandrika Tirukkovalluri

Guided by

Prof. Hamidreza Heidari

Contents

Project Overview

"Development of a smart inspection cell for automated defect detection in automotive components."

- Utilizes YOLOv8 and MobileNetV2 deep learning models integrated with a simulated conveyor system (RoboDK + SimPy).
- Detects surface defects like scratches, dents, cracks, and pitting on metal components (such as a spur gear).
- Aims to achieve real-time, high-accuracy inspection and reduce manual intervention in manufacturing lines.

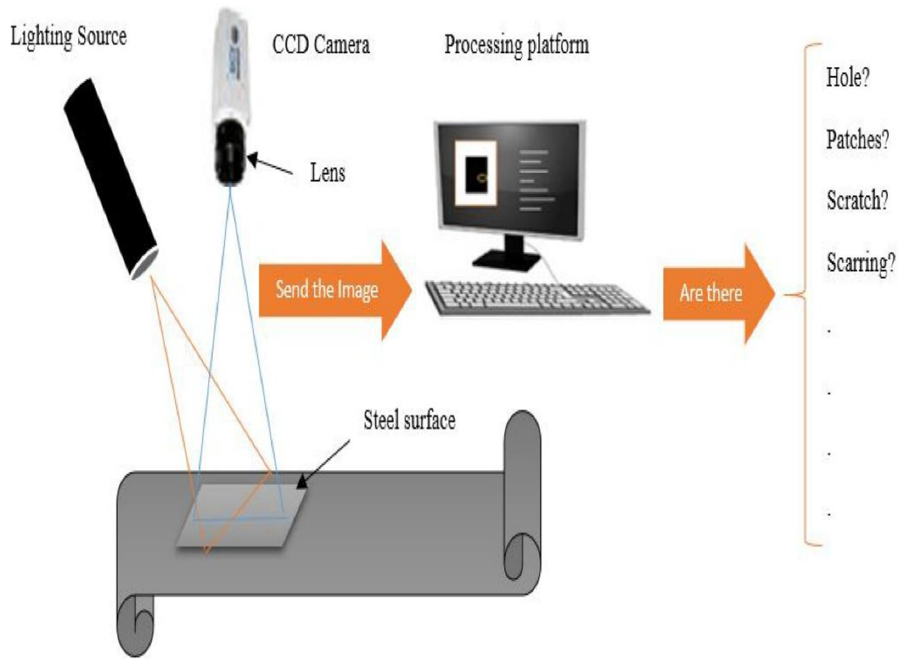


Figure 1.Architecture of Automated Defect Detection System

Industrial Problem Addressed

- Issues with manual visual inspection:
  - Slow
  - Inconsistent, and
  - Error-prone.
- Surface defects such as scratches, cracks, and dents can go undetected, causing rework and financial losses.
- Need for **automated**, **intelligent** quality inspection systems.
- The project addresses this gap by building an AI-driven inspection cell that enhances:
  - accuracy,
  - throughput, and
  - consistency.

Related Work

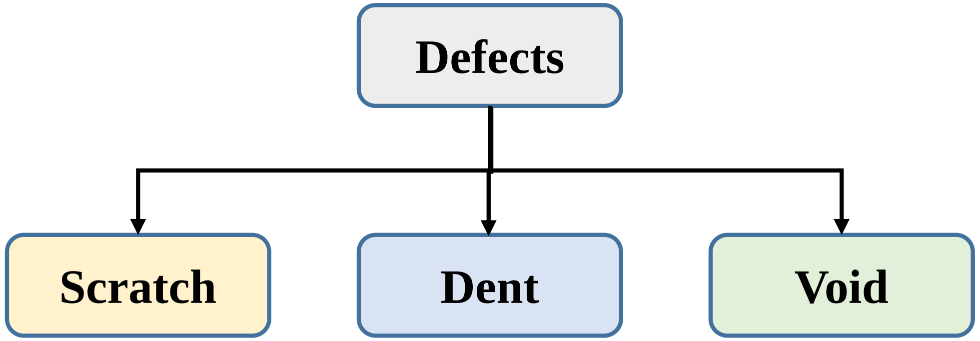
- Zhou et al., 2023 – Applied YOLOv5 for metal surface defect detection; achieved high accuracy but limited in real-time deployment.
- Liu et al., 2022 – Used CNN architectures for automotive component inspection but lacked integration with automation frameworks.

Knowledge gap Identified

- *No comparative evaluation of YOLOv8 and MobileNetV2 in a simulated smart inspection environment combining vision AI with industrial automation tools.*

Phase 2 Progress

- Generated a synthetic dataset of spur gear images with defects.



- Modelled defects using **Onshape** and rendered multiple geometric variations for dataset diversity.
- Generated baseline simulation setup with robot setup in **RoboDK**.
- Maintained the GitHub repository up-to-date with generated dataset and documentation.

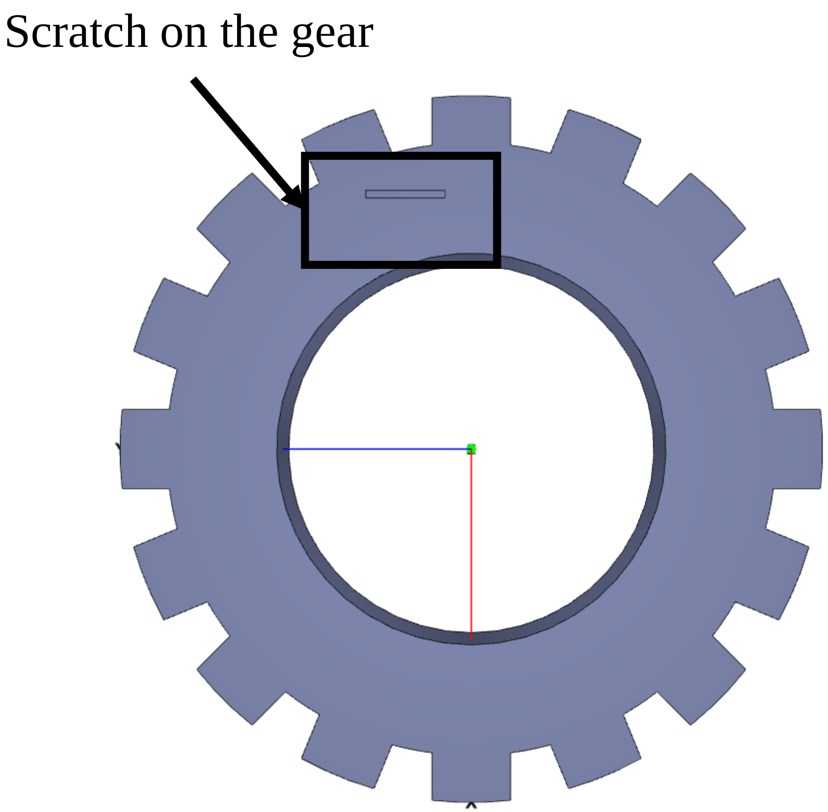


Figure 2. Example of a data sample with a scratch.

Work in Progress

- Data annotation and labelling.
- Setting up the model training.
- Scaling of the gear components to make it fit to the conveyor sim.
- Configure camera and inspection path for the virtual cell using **SimPy** and **RoboDK** API**.**
- Python Scripting for automatic loading of components from dataset into sim setup at fixed intervals

```
Dataset/
│
├─ gear_good_master.step    → generates → gear_good/ (good samples)
├─ gear_dent_master.step    → generates → gear_dent/ (dent defect images)
├─ gear_scratch_master.step → generates → gear_scratch/ (scratch defect images)
├─ gear_voids_master.step   → generates → gear_voids/ (pitting/void defect images)
│
└─ (Each folder contains rendered .png images for YOLOv8 training)
```

Figure 3. Dataset Structure Used for Training YOLOv8 Model on Gear Defects

Tools and Methods Used

- **Data Generation**: Onshape **(**defect modeling) + synthetic image rendering.
- **Simulation**: RoboDK for virtual conveyor cell simulation; integrated via SimPy for process logic.
- **Model Training**: YOLOv8 and MobileNetV2 (image classification).
- **Annotation**: CVAT for bounding boxes and defect labeling.
- **Evaluation Metrics**: Accuracy, Precision, Recall, F1-Score, and Inference Latency.

Figure 4. YOLOv8 framework Logo.

Simulation logic Summary

Sequence	Action
1	Conveyor moves spur gear into camera view.
2	Robot-mounted camera captures image.
3	Python AI model classifies gear (good/defect).
4	If “good” → Robot activates gripper, picks gear, drops in blue bin.
5	If “defect” → Robot activates gripper, picks gear, drops in orange bin.
6	Robot returns to home; process repeats for next gear.

Table 1. Sequence of actions in Simulation Environment.

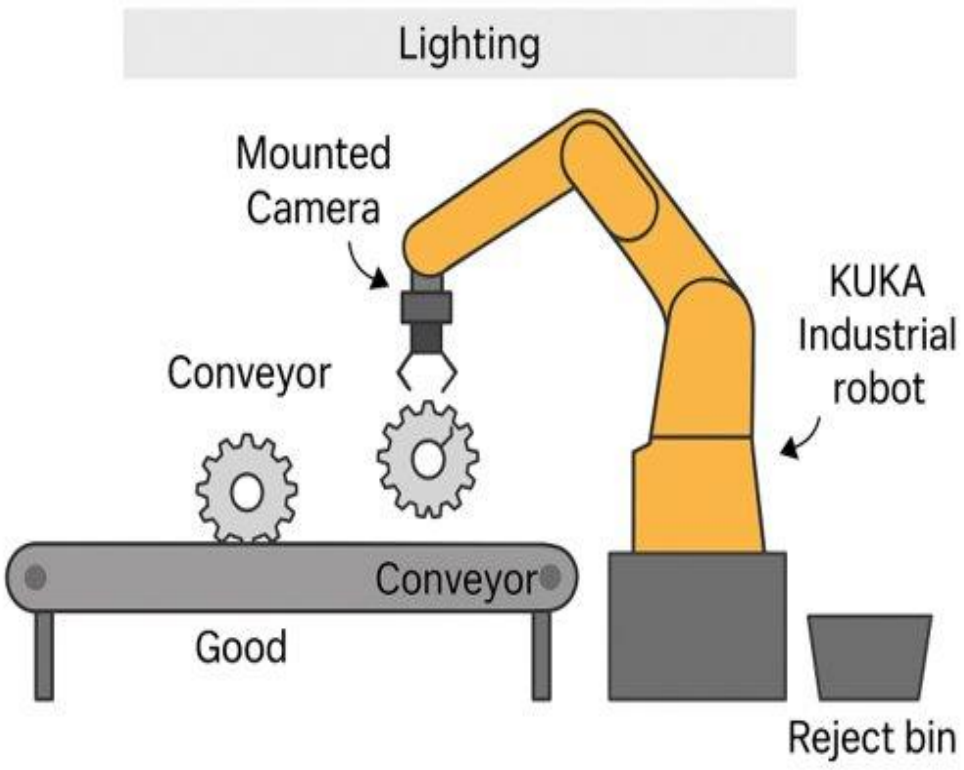


Figure 5. Simulation Logic Flow of smart inspection cell.

Simulation Setup

Frame	Parent	Function
Base_Frame	Global	Main cell reference
Robot_Base_Frame	Base_Frame	Robot coordinate origin
Conveyor_In_Frame	Base_Frame	Entry conveyor position
Conveyor_out_Frame	Base_Frame	Exit conveyor position
Bin_Blue_Frame	Robot_TCP	Container for parts
Bin_Orange_Frame	Robot_TCP	Container for defect parts
Camera_Frame	Robot_TCP	Vision alignment

Table 2. Defined Coordinate Frames and their Functions in the Simulation setup.

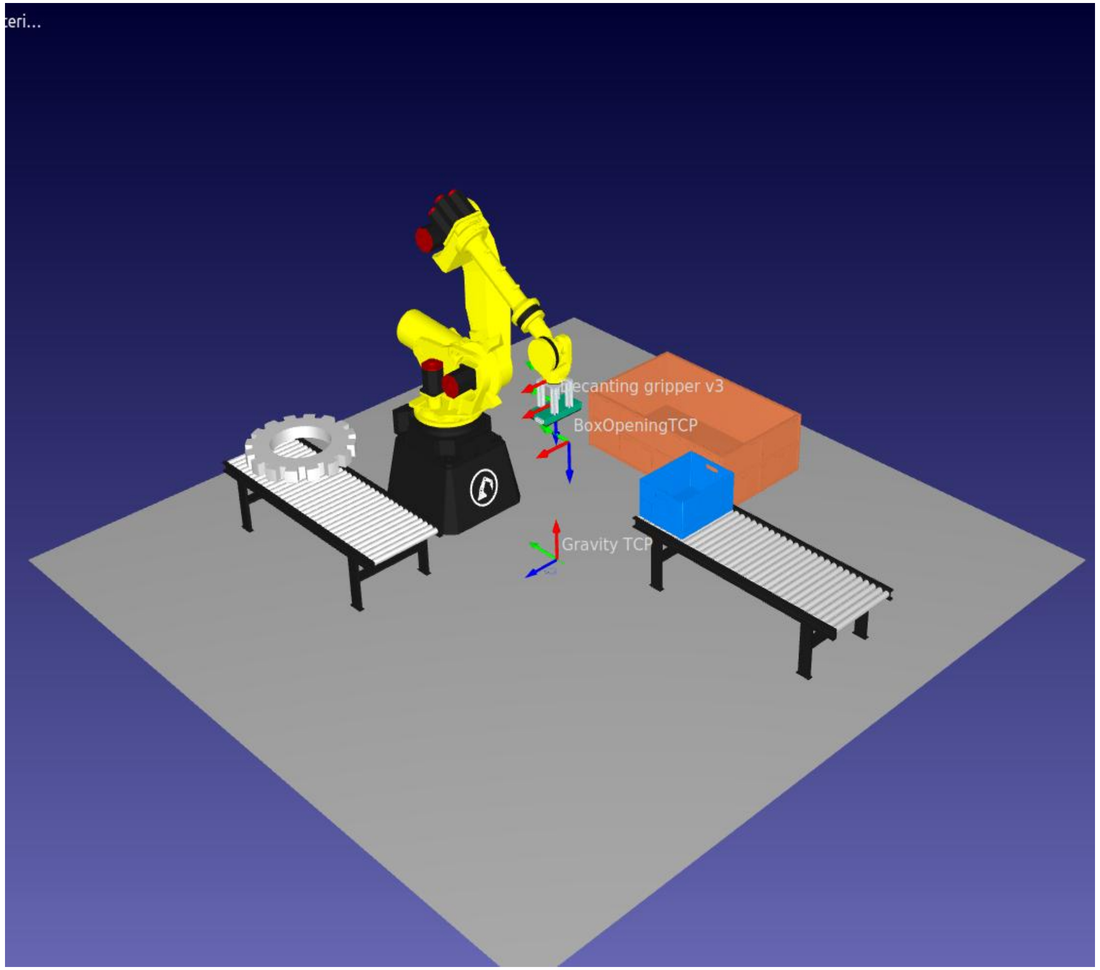


Figure 6. Operational Logic Sequence in Robodk environment.

Expected Outcomes

- Fully functional simulated smart inspection cell with automated defect detection.
- Comparative results of YOLOv8 vs MobileNetV2 on accuracy, latency, and throughput.
- Curated synthetic dataset of spur gear defects for training and future research.
- System framework ready for real-world deployment in industrial quality inspection cells.

Milestone

Work	(Oct)Week42	Week43	Week44	(Nov)Week45	Week46	Week47	Week48	(Dec)Week49	Week50	Week51	Week 52	(Jan)Week 01	Week 02
Dataset preparation & annotation													
Sim Setup & Model training													
Model deployment & evaluation													
Final presentation & report													

Next Steps and GitHub Upload

- Training Yolov8, MobileNetV2 models and comparing performance metrics.
- Scripting API between Simpy and RoboDK for component loading and robotic sorting integration.
- Upload the sim demonstrations, latest report, refined datasets, and code to the team GitHub repository.

Key References

- Zhang et al., “Deep Learning for Visual Surface Defect Detection,” IEEE Access, 2023.
- Liu et al., “CNN-Based Automated Defect Detection in Automotive Parts,” Journal of Manufacturing Systems, 2022.
- Ultralytics YOLOv8 Documentation, 2024.
- SimPy Documentation, 2024.
- OpenCV and RoboDK API References, 2024.

Thank You

https://github.com/Pallelayaswitha1/Integration-and-Comparison-of-vision-models-for-smart-inspection-cell/tree/main

Any Questions?