

# ML & AI Internship Assignment: Industrial Equipment Defect Detection

---

Author: Archita Dhande

Program: ML & AI Internship

Date: 6th June 2025

## 1. Objective

The goal of this project was to develop a machine learning model capable of classifying images of industrial equipment into two categories:

- **Defective**
- **Non-Defective**

Bonus objectives included:

- Identifying specific defect types
- Optimizing the model for hardware-accelerated inference (optional, not covered here)

## 2. Dataset Description

Dataset: <https://www.mvtec.com/company/research/datasets/mvtec-ad/downloads>

The provided **hazelnut dataset** consisted of two main folders:

- train with one class: good
- test with five sub-classes: good, crack, cut, hole, and print

For binary classification:

- All images except good were labeled **defective**
- good images labeled **non\_defective**

### Initial Data Distribution

Set	good (non-defective)	defective (combined from test defects)
Train	391	0

Set	good (non-defective)	defective (combined from test defects)
Test	40	70

### 3. Problem Faced

Training set lacked defective images, causing model overfitting with 100% training accuracy and ~50% validation accuracy.

### 4. Solution Applied

Moved 10 defective images from the test set into the training set to ensure both classes were represented.

#### Resulting Distribution:

Set	non-defective	defective
Train	391	10
Test	40	8

Recomputed class weights to address imbalance.

#### Lesson:

Always verify class balance in both train and test sets before model training.

### 5. Data Preparation

- Resized Image resizing to (224, 224)
- Rescaling pixel values to [0, 1]
- Data augmentation:
  - Horizontal flip
  - Random zoom
  - Random rotation

## 6. Model Development

Used **MobileNetV2** pretrained on ImageNet for transfer learning.

### Model Pipeline:

- MobileNetV2 base (frozen)
- Global Average Pooling
- Dropout (0.3)
- Dense layer with sigmoid activation for binary classification

**Loss Function:** Binary Crossentropy

**Optimizer:** Adam

**Metrics:** Accuracy

## 7. Handling Class Imbalance

Computed class weights based on new class counts to penalize minority class errors more.

## 8. Training & Validation

Epochs: 20, EarlyStopping (patience=5)

Epochs	Final Train Accuracy	Final Validation Accuracy	Early Stopping
6	96.9%	83.3%	Yes

## 9. Results & Evaluation

Metric	Value
Accuracy	83.33%
Precision	69.44%
Recall	83.33%
F1-Score	75.76%

## 10. Insights & Challenges

1. **Initial overfitting** caused by the absence of defective images in the training set.
2. **Severe class imbalance** (391 vs 10) still affected performance despite using class weights.
3. **High recall** is useful in defect detection scenarios but at the cost of false positives.
4. MobileNetV2 with transfer learning and data augmentation significantly improved model stability and accuracy.

## 11. Lessons Learned

Check class distributions early, class imbalance severely affects reliability, Transfer Learning boosts small dataset performance.

## 12. Conclusion & Future Scope

This project successfully built a defect detection model using transfer learning with MobileNetV2, achieving an **83.3% test accuracy**. Class imbalance remains a critical issue in industrial defect detection workflows and should ideally be addressed by collecting more balanced data or applying advanced synthetic data generation techniques. Implement multi-class classification, synthetic data generation, model quantization, and hyperparameter tuning.