Final Presentation: Image Recognition App

Szymon Chirowski (242621), 11th April 2025

Introduction

Persona: You're a small manufacturing company owner scaling operations.

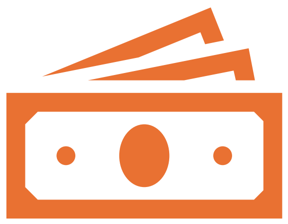
Problem: Manual classification & quality control can't keep up with growth.

Pain point: Defects are increasing and costing money.

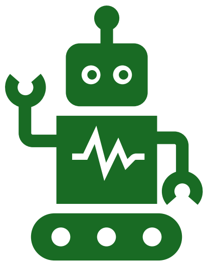
Realization: Need for an automated, real-time QC system.

Market problem: Existing industrial systems cost ~$15,000+ (without hardware).

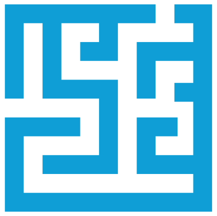
Project Presentation



Cost Advantage: My custom solution is ~10% of the cost of mainstream systems.

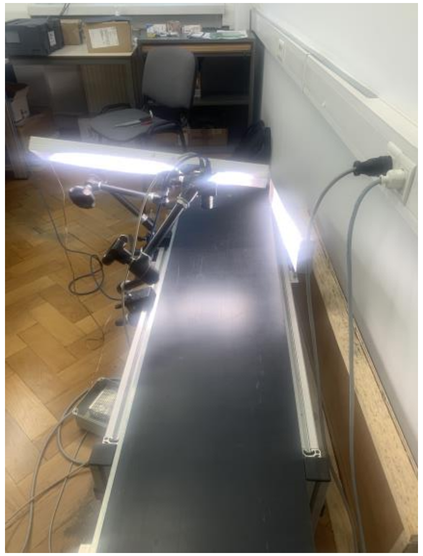


Offer: An affordable AI-powered image classification & defect detection system.



Benefits: Easy-to-use, real-time results, no unnecessary complexity.

Practical, Scalable, and Affordable



•Real-time object classification.

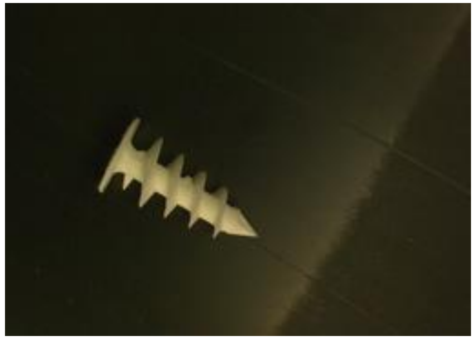
•Runs on low-cost hardware (even Raspberry Pi).

•USB3-compatible camera support.

•Proof of Concept: Simple and cost-effective setup (2 lamps, $100 sensor, $700 camera).

Let’s explore how I built the system from the ground up.

Problem Overview



- Dataset: ~500 images of 4 wall plug classes.
- Classes: Frame fixing plug, insulation anchor, ribbed wall plug, toggle anchor.
- Preprocessing:
    - Resizing 4K images for speed.
    - Rescaling to improve model performance.

Baselines

Goal: Understand human vs machine performance before training.

Human Level Performance (HLP): 95% accuracy.

Basic Neural Network (MLP):

- Looked at the full image without focus on features.
- Result: 84% accuracy.

Insight: A stronger model is needed to match/exceed human accuracy.

Model Overview: Iterative Development

- Iteration 1:
  - Started with a simple CNN architecture.
  - Architecture:
    - Input Layer
    - Conv2D Layer
    - MaxPooling2D
    - Conv2D Layer
    - MaxPooling2D
    - Batch Normalization
    - Dense Layer
    - Output Layer
  - Result: 90% accuracy.

- Iteration 2:
  - Problem: AI needs more data
  - Solution: Data augmentation – 5 new images per original.
  - Result: Accuracy improved to 92%

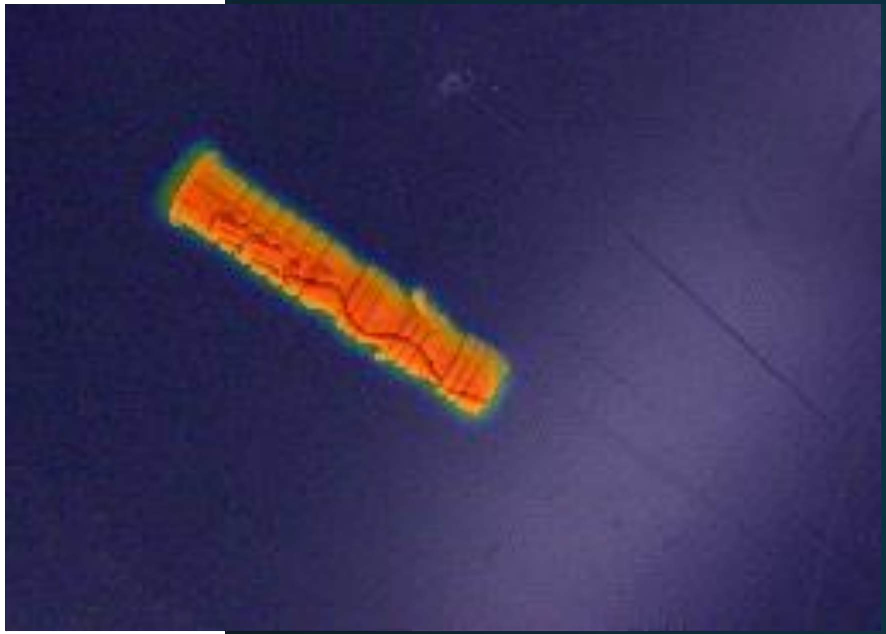
Model Overview: Iterative Development

- Iteration 3:
  - Applied Transfer Learning with MobileNetV2 (pre-trained on ImageNet)
  - Benefit: Leverage real-world knowledge for better learning.
  - Result: 98.3% accuracy.

- Iteration 4:
  - Combined augmentation with EfficientNetB0.
  - Result: 100% accuracy.
  - Limitation: Not suitable for Raspberry Pi – too computationally heavy.

Explainable AI (XAI)

- Question: Can we trust the model's decisions?
- Solution: Grad-CAM visualisations via XAI dashboard.
- Explanation:
  - Red/yellow – important areas used by the model.
  - Blue – ignored areas.
- Purpose: Verify model decisions align with human expectations.



Think-Aloud Study

- Purpose: Test the usability and intuitiveness of the app.
- Tasks:
  - Single image classification.
  - Batch classification.
  - Run XAI module.
- Results:
  - Positive: Easy to use, one-click workflows, batch upload feature praised.
  - Negative: Lack of XAI explanation, unclear batch results layout.
- Improvements made: Updated UI, added separators, and guidance.

Thank you for your attention!

Questions?