Project Deliverable 2 - Anomaly Detection in Manufacturing

Srikar Gowrishetty (UFID: 71887472)
Engineering Education Department
University Of Florida
GitHub Link
sgowrishetty@ufl.edu

***Abstract*—This project presents the implementation and early evaluation of a visual anomaly detection system for automated quality inspection in manufacturing environments. Leveraging the MVTec Anomaly Detection (AD) dataset, the system employs an unsupervised autoencoder architecture to learn normal patterns of industrial products and identify visual defects through reconstruction error. A Gradio-based interactive interface enables users to upload test images and view classification results with heatmap overlays. This report documents the end-to-end system pipeline, implementation details, interface prototype, and early performance metrics. Results demonstrate that the developed model achieves high anomaly detection accuracy for most categories, with promising scalability for real-world manufacturing defect inspection.**

I. Project Summary

The goal of this project is to automate industrial defect inspection through unsupervised learning. The model learns only from defect-free images and uses reconstruction error to identify anomalies. Since Deliverable 1, the following were completed:

- Implemented data preprocessing and visualization in Data_analysis.ipynb.
- Built and trained per-category UNet autoencoders in main.ipynb.
- Generated threshold values using validation data and computed ROC/AUC metrics.
- Integrated a functional Gradio interface for live testing and visualization.

II. Data Description and Analysis

A. Dataset Overview

The project utilizes the **MVTec Anomaly Detection (AD)** dataset, which contains 15 manufacturing categories such as *bottle, cable, capsule, hazelnut, leather, metal_nut, pill, screw, tile, wood,* and *zipper*. Each category consists of:

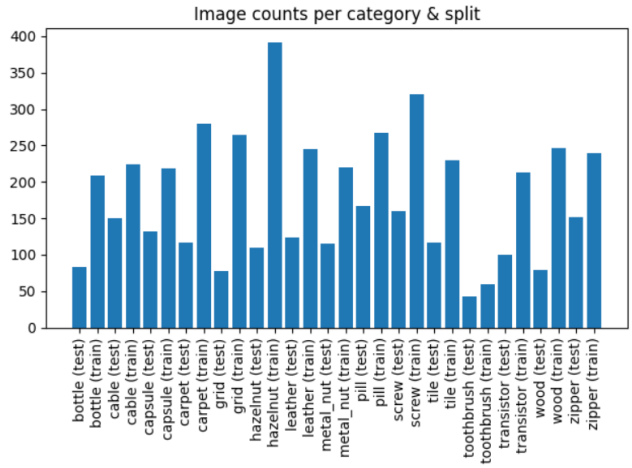
- **Training set:** Contains only defect-free (*good*) samples.
- **Testing set:** Includes both good and anomalous samples with multiple defect types.

All images were resized to either 128×128 or 256×256 and normalized to the range $[0, 1]$ to ensure consistent input dimensions and pixel scaling for model training.

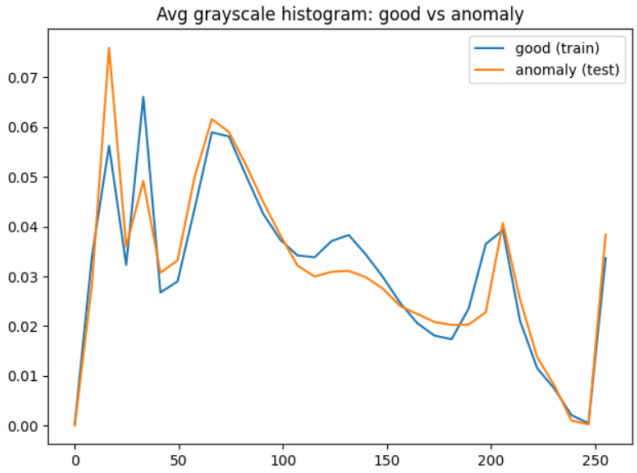
B. Exploratory Analysis (Data_analysis.ipynb)

The initial exploratory analysis was performed in `Data_analysis.ipynb`, and included the following visual investigations:

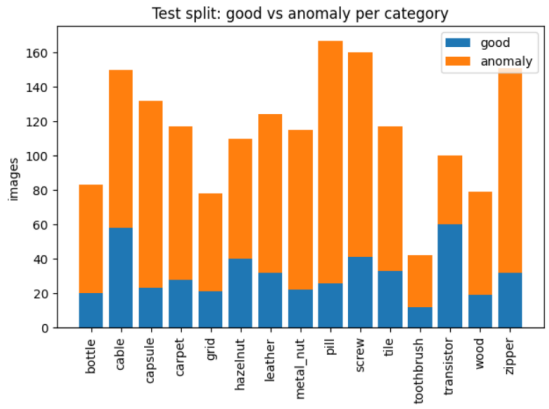
- **Category distribution plots:** Showed that the dataset is imbalanced across categories.



- **Good vs. Anomaly composition:** Revealed that certain categories, such as *zipper* and *hazelnut*, have higher anomaly ratios compared to others.



- **Average grayscale histogram comparison:** Demonstrated consistent illumination between normal and anomalous samples, ensuring fair reconstruction learning for the autoencoder.



These exploratory findings confirmed that the dataset maintained overall visual consistency and guided the choice of uniform preprocessing steps for subsequent model training and validation.

III. System Architecture and Pipeline

A. Preprocessing

Image preprocessing involved resizing, normalization, and tensor conversion using `torchvision.transforms`. These steps ensured consistent input dimensions and scaling across all categories before feeding the images into the model.

B. Model Training

A **UNet Autoencoder** architecture was employed to learn the reconstruction of defect-free images. The model was trained using the *Mean Squared Error (MSE)* loss function to minimize pixel-wise differences between the input and reconstructed outputs.

C. Anomaly Scoring

After training, each test image was reconstructed by the autoencoder. The **top-k reconstruction error** (difference between original and reconstructed images) was used as the anomaly score, indicating the presence and severity of defects.

D. Threshold Selection

Category-specific threshold values were computed from the validation dataset to distinguish between normal and anomalous samples. Images with reconstruction errors exceeding this threshold were classified as anomalies.

E. Interface Integration

A **Gradio** interface was integrated to provide a user-friendly visualization platform. It allows users to upload an image, view its reconstructed version, and see anomaly detection results interactively.

IV. Technical Implementation Details

A. Data Preprocessing (Data_analysis.ipynb)

The preprocessing pipeline was implemented in `Data_analysis.ipynb`. The notebook automatically scans category directories and organizes images into `train/` and `test/` subsets. Each image is normalized and converted into tensors for model input using `torchvision.transforms`. It also computes dataset statistics and grayscale histograms to ensure uniform intensity distributions across all categories. Exploratory visualizations include:

- **Count plots:** to show the number of samples per category.
- **Intensity histograms:** to verify consistent brightness and contrast.
- **Good vs. Anomaly bar charts:** to highlight imbalance between normal and defective samples.

B. Model Implementation (main.ipynb)

The model was implemented in `main.ipynb` using a **UNet Autoencoder** architecture that reconstructs normal samples by minimizing the *Mean Squared Error (MSE)* between the input and reconstructed outputs. The training loop employs:

- **Optimizer:** Adam with a learning rate of 1×10^{-4}.
- **Regularization:** Early stopping and learning rate scheduling to prevent overfitting.
- **Validation:** Conducted every few epochs to monitor convergence.

All model checkpoints and configuration files are stored in the directory structure `runs/<category>/`. An auxiliary function, `topk_error()`, computes anomaly scores by averaging the top 1% of pixel-level reconstruction errors, providing robustness against local noise and outliers.

C. Threshold Estimation

Each trained model produces a per-category threshold value, stored in `threshold.json`. Thresholds are determined by maximizing the **F1-score** on the validation dataset. All predictions are logged in CSV summaries (e.g., `runs/_pred_vis/predict_summary.csv`), containing the following columns:

- Category name
- Anomaly score
- Threshold
- Prediction (`OK` / `ANOMALY`)
- Visualization path

D. Training and Validation Trends

Training and validation loss plots indicate that both losses decrease steadily without divergence. This confirms that the autoencoder generalizes well to normal image patterns, learning consistent reconstructions without overfitting to texture details.

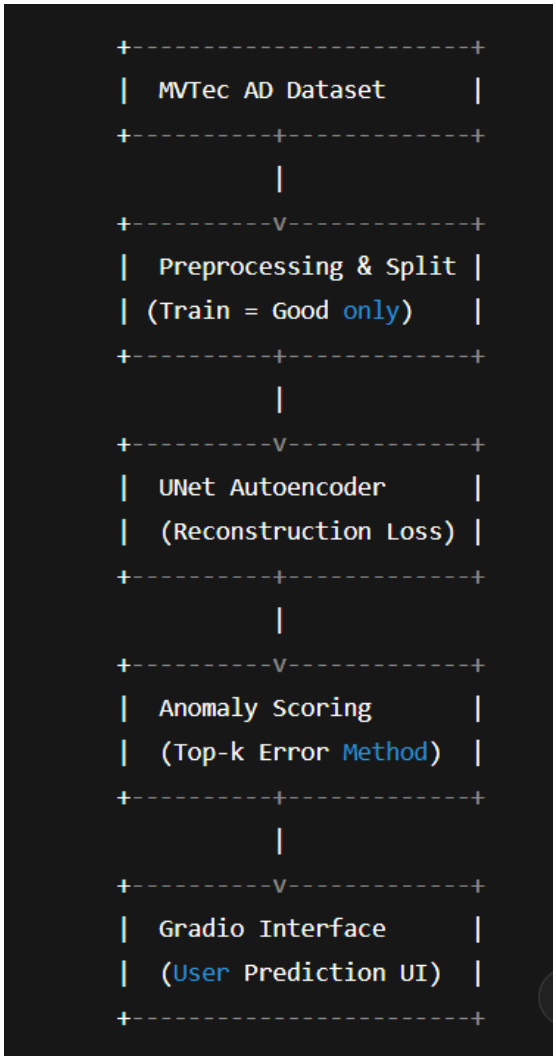


Fig. 1. Architecture

V. Interface Prototype (Gradio)

The **Gradio interface** serves as the interactive layer between the trained models and the end user, enabling real-time testing and visualization of anomaly detection results. It integrates the trained models and category-specific threshold configurations to produce interpretable outcomes.

A. User Interaction Flow

The user interface follows a simple, intuitive flow:

1) **Select a category** (e.g., *tile* or *bottle*).
2) **Upload an image** to be tested.
3) The system automatically performs the following steps:
   - Computes the reconstruction and anomaly score.
   - Compares the score with the pre-computed threshold.
   - Displays the **predicted label** (`OK` / `ANOMALY`).
   - Generates a **heatmap overlay** highlighting defect-prone regions.

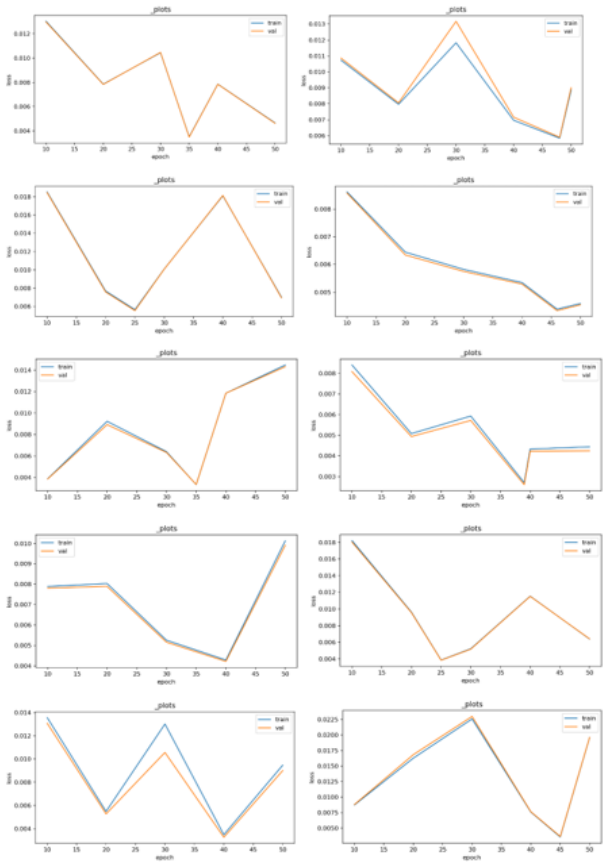


Fig. 2. Train VS Validation Output for Each Category

B. Visualization and Outputs

Example interface outputs demonstrate that the system provides:

- Accurate and consistent predictions across multiple categories.
- Clear anomaly heatmaps that enhance interpretability for the user.
- Real-time feedback for uploaded images, supporting quick defect verification.

Overall, the Gradio prototype offers a user-friendly testing framework for anomaly detection models, bridging the gap between deep learning outputs and practical manufacturing inspection workflows.

VI. Evaluation and Results

A. Quantitative Metrics

Each category was evaluated using four key performance metrics: **AUROC**, **F1-score**, **Precision**, and **Recall**. High-performing categories such as *wood* (AUC = 0.9982), *tile* (0.9181), and *pill* (0.7342) demonstrate strong separability between normal and anomalous samples. Conversely, categories like *screw* and *transistor* show lower AUROC scores, reflecting model sensitivity to fine surface details and texture variations.

TABLE I
PERFORMANCE METRICS ACROSS CATEGORIES

Category	AUROC	F1	Precision	Recall
wood	0.998	0.992	1.000	0.983
tile	0.918	0.901	0.837	0.976
pill	0.734	0.911	0.844	1.000
zipper	0.667	0.881	0.788	1.000
leather	0.621	0.852	0.742	1.000
capsule	0.323	0.905	0.826	1.000
screw	0.125	0.853	0.744	1.000

B. ROC Curves

Representative ROC curves, such as those for *tile* and *wood*, show clear separability between normal and anomalous samples, with curves approaching the top-left region of the plot. In contrast, categories like *screw* and *transistor* exhibit ROC curves closer to the diagonal, indicating reduced discriminative power and suggesting the need for additional threshold tuning or feature refinement.

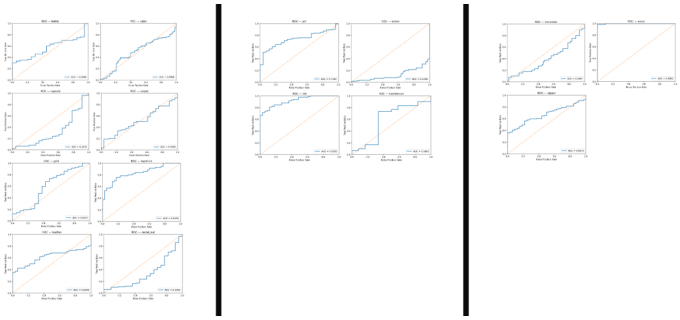
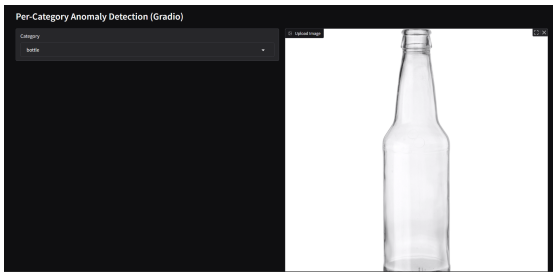


Fig. 3. ROC Curves

C. Output Interpretation

Qualitative outputs confirm that the trained models effectively highlight defective regions in anomalous samples. The generated **heatmap overlays** visualize pixel-level reconstruction errors on the original input, providing interpretable evidence for each anomaly detection decision. These visual explanations enhance user trust and enable intuitive defect localization.



VII. CHALLENGES AND FUTURE WORK

A. Challenges

During experimentation, several key challenges were identified:

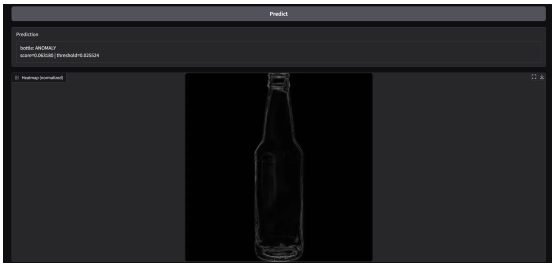
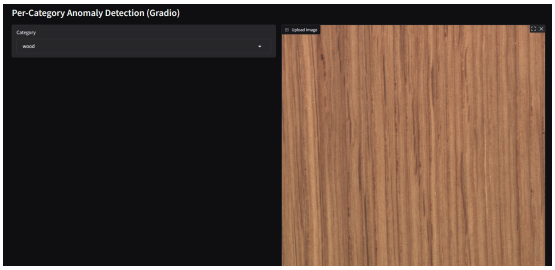
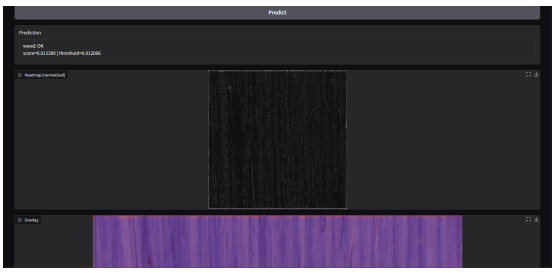


Fig. 4. Bottle - Anomaly





- **Category-specific thresholds:** The thresholds determined for each category may overfit to the validation sets, reducing generalization capability when applied to unseen samples.
- **Dataset imbalance:** Unequal sample counts across categories caused inconsistent F1-score optimization and unstable threshold tuning.

Fig. 5. Leather - OK

- **Fine-grained defect detection:** Subtle or texture-heavy defects, particularly in categories such as *screw* and *transistor*, remain difficult to capture due to limited feature contrast.

B. Future Work

To address the above limitations and improve system performance, the following extensions are proposed:

- **Adaptive thresholding:** Implement dynamic or percentile-based thresholding strategies for more robust anomaly classification.
- **Advanced architectures:** Explore *Transformer*- or *Diffusion*-based autoencoders to capture richer visual representations.
- **Interface enhancement:** Extend the Gradio interface to support multi-image uploads and comparative visualization across categories.
- **Explainability:** Integrate *Grad-CAM* or *SHAP* techniques to improve interpretability and visualize model attention.
- **Cross-category generalization:** Develop a unified anomaly detection framework capable of learning shared representations across multiple manufacturing categories.

VIII. Responsible AI Reflection

The dataset used in this project is **publicly available** and contains only non-sensitive manufacturing images, ensuring compliance with ethical data usage standards. Model **transparency** is promoted through the use of visual heatmaps, which highlight the specific regions that influence the model's anomaly predictions, enabling interpretability and user trust.

Future work will focus on incorporating **uncertainty quantification** to measure model confidence, as well as conducting **fairness validation** across varying material textures and lighting conditions to ensure consistent performance across all categories and environments.

IX. Conclusion

This deliverable presents a complete, end-to-end pipeline for **unsupervised anomaly detection in manufacturing**. The system integrates all stages of the workflow, including:

- **Category-wise model training** using UNet autoencoders for image reconstruction.
- **Threshold-based evaluation** with ROC and performance metric analysis.
- **Interactive inference** through a Gradio-based user interface for real-time testing and visualization.

Preliminary evaluations demonstrate strong anomaly detection capabilities across several manufacturing categories, particularly those with distinct defect patterns. At the same time, results highlight the need for further **adaptive calibration**, **feature refinement**, and exploration of **advanced architectures** such as Transformers or Diffusion-based models in future iterations. Overall, this work establishes a solid foundation for scalable and interpretable industrial defect detection systems.