Project Deliverable 1

Project: Anomaly Detection in Manufacturing
Course: Applied Machine Learning 2
Author: Srikar Gowrishetty

Github: Link

a. Problem Context and Project Summary

Modern manufacturing industries rely on consistent product quality and defect-free production lines. However, manual inspection is often slow, error-prone, and inconsistent especially when defects are small, rare, or visually subtle. Missed anomalies can lead to significant financial losses, recalls, and customer dissatisfaction.
This project aims to develop an AI-based anomaly detection system that can automatically identify defects in industrial products using computer vision. By training deep learning models (autoencoders, variational autoencoders, and transformer-based architectures) on the MVTec AD dataset, the system will learn to recognize normal manufacturing patterns and detect unseen anomalies in real-time.
The ultimate goal is to demonstrate how AI can transform traditional visual inspection into a smart, scalable, and automated quality control solution aligned with the Industry 4.0 vision.

b. Dataset

Dataset Name: MVTec Anomaly Detection (AD)
Source: MVTec Software GmbH
Type: High-resolution RGB images
Size: ~5,354 images across 15 categories (objects and textures)
Categories: bottle, cable, capsule, carpet, grid, hazelnut, leather, metal nut, pill, screw, tile, toothbrush, transistor, wood, zipper

Data Format & Access:

- Images are in .png format organized by category folders.
- Each category contains a **train** folder (normal images) and **test** folder (normal + defective images).
- Pixel-level **ground truth masks** are provided for defects.
- The dataset can be downloaded directly from the official MVTec website (public license).

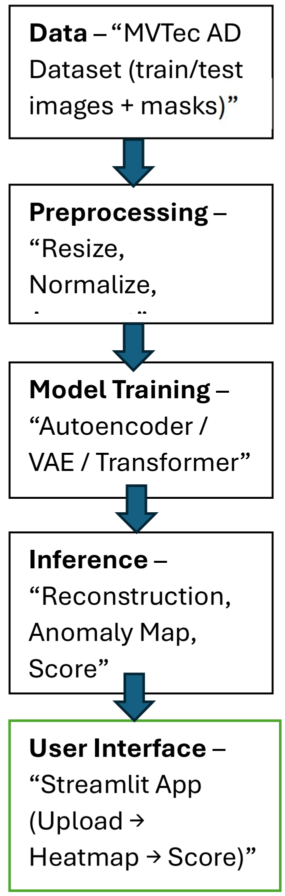
Preprocessing Challenges:

- Images vary in resolution → need resizing and normalization.
- Class imbalance (few defective samples).
- Some categories have subtle anomalies requiring precise reconstruction accuracy.

Ethical / Privacy Considerations:

- Dataset does **not contain personal or sensitive data**.
- Publicly released for research and benchmarking; ethical use aligned with license.
- No identifiable human information, so no privacy concerns.

Flowchart:



The flowchart illustrates the **end-to-end workflow** of the Anomaly Detection in Manufacturing system, showcasing how data flows from raw images to meaningful defect insights through deep learning and a user-friendly interface.

1. **Data (MVTec AD Dataset)**
   The pipeline begins with the MVTec AD dataset, which contains high-resolution images of 15 industrial product categories. Each category includes normal (defect-free) training images and defective test images with pixel-level ground-truth masks. This dataset serves as the foundation for learning the visual patterns of normal manufacturing behavior.

2. **Preprocessing**
   All images undergo preprocessing to ensure consistency and readiness for model training. This includes resizing to a fixed resolution, normalization of pixel intensity values, and optional data augmentation (such as rotations or flips) to improve model generalization. The goal is to standardize the input for efficient feature learning.

3. **Model Training**
   In this stage, a deep learning model typically a **Convolutional Autoencoder (AE)** or **Variational Autoencoder (VAE)** is trained exclusively on normal (non-defective) samples. The model learns to reconstruct normal patterns, thereby encoding the “healthy” visual characteristics of industrial products. During training, the model minimizes reconstruction error, which later serves as the anomaly indicator.

4. **Inference / Anomaly Detection**
   When a test image is passed through the trained model, it produces a reconstructed version. The system compares the input and reconstructed images to generate an **anomaly map** (or defect heatmap). Areas with high reconstruction error indicate potential defects. The system also computes an **anomaly score**, quantifying the level of deviation from normal patterns.

5. **User Interface (Streamlit Application)**
   The final stage involves deploying the trained model through an interactive **Streamlit** interface. Users can upload an image, view the original and reconstructed versions side by side, and visualize the detected defect regions through a color-coded heatmap. The anomaly score and pass/fail decision are displayed automatically, making the tool accessible to non-technical manufacturing personnel.

d. User Interface Plan

Purpose

Provide a simple, non-technical interface for factory/QA staff to upload a product image and instantly see whether it contains a defect, where it is, and how confident the model is supporting fast, explainable decisions.

Inputs (from the user)

- **Image upload**: PNG/JPEG of a product (single image).

Outputs / Feedback (from the system)

- **Original image** (left).
- **Reconstructed image** (center) from the autoencoder/VAE.
- **Defect heatmap overlay** (right) highlighting anomalous regions.
- **Anomaly score** (0–1) and **Pass/Fail** decision (based on threshold).

Interaction Flow

1. User uploads image (and selects category, if enabled).
2. App preprocesses and runs the trained model.
3. UI displays original, reconstruction, heatmap, score, and pass/fail.

4. User can adjust threshold to see effect on decision.

How this enhances usability & interpretability

- **Visual explainability**: Heatmap shows *where* and *why* the model flags a defect.
- **Actionable metrics**: Anomaly score + pass/fail support quick line decisions.
- **Operator control**: Threshold slider adapts sensitivity to production context.
- **Consistency**: Standardizes inspection vs. variable human judgment.

An Expected UI using a Wireframe:

```
+----------------------------------------------------------------------------------------------+
| Anomaly Detection UI (Streamlit)                                                  |
+----------------------------------------------------------------------------------------------+
| Upload Image: [ Browse... ]                                          [ Predict ]     |
+----------------------------------------------------------------------------------------------+
|  Original                  Reconstruction              Defect Heatmap + Overlay   |
| [          ]              [          ]              [          ]          |
|                                                                 |
|  Anomaly Score: 0.83       Decision: FAIL                             [ CSV ] |
+----------------------------------------------------------------------------------------------+
| Notes: High reconstruction error detected on upper-right area (possible scratch).            |
+----------------------------------------------------------------------------------------------+
```

e. Innovation and Anticipated Challenges

Innovation

The proposed project introduces a **real-world, unsupervised deep learning approach** for automated defect detection in manufacturing using the **MVTec AD dataset**. Unlike traditional rule-based or fully supervised quality control systems, this approach learns only from normal (defect-free*)* images, allowing the model to detect previously unseen defects without explicit labels.

Key innovative aspects include:

- **Unsupervised Defect Detection:**
  The model (Convolutional Autoencoder or Variational Autoencoder) learns the

distribution of normal products and identifies anomalies based on reconstruction error, eliminating the need for large labeled defect datasets.

- **Visual Explainability:**
  Anomaly heatmaps highlight where the defect occurs, providing interpretable visual evidence to support decision-making important for industry adoption and trust.
- **Interactive Deployment:**
  The integration with **Streamlit** enables real-time inspection and decision support, making the system usable by manufacturing engineers and non-technical operators alike.

This design bridges the gap between cutting-edge AI research and industrial inspection needs offering a lightweight, explainable, and deployable solution for modern factories.

f. Implementation Timeline

Week	Focus	Expected Outcome
Oct 20 – Oct 29	**Dataset Exploration and Environment Setup**	Successfully load and explore the MVTec AD dataset. Generate EDA summaries, visualize image counts and categories, and verify data structure. Set up GitHub repository, environment (requirements.txt), and Jupyter workspace for initial testing.
Oct 30 – Nov 6	**Baseline Model Development (Autoencoder)**	Implement and train a baseline convolutional autoencoder on a single category (e.g., *bottle*). Validate model performance using reconstruction loss and visualize anomaly heatmaps. Save training checkpoints and graphs.
Nov 7 – Nov 14	**Model Refinement and Evaluation**	Extend the model to multiple categories, fine-tune hyperparameters, and test additional architectures (VAE or hybrid). Introduce SSIM/L1 loss combination and evaluate ROC-based thresholding for better anomaly scoring.
Nov 15 – Nov 22	**Visualization and Interpretability Enhancements**	Generate reconstruction error heatmaps and overlay defect masks for interpretability. Analyze subtle defect cases and produce clear visual outputs for report documentation.
Nov 23 – Nov 30	**Streamlit UI Development and Integration**	Develop a Streamlit-based interface with upload functionality, real-time inference, and visual results (original, reconstructed, and heatmap images). Integrate anomaly score and pass/fail display.

Dec 1 – Dec 7	**UI Refinement, Testing, and Final Evaluation**	Optimize UI performance, add threshold slider, and validate across multiple categories. Conduct final experiments, ensure reproducibility, and compile evaluation metrics.
Dec 8 – Dec 11	**Final Report and Presentation Preparation**	Finalize report and GitHub documentation. Prepare demo video or live Streamlit walkthrough. Submit final technical report and present project outcomes.

g. Responsible AI Reflection

This project emphasizes the responsible application of artificial intelligence in industrial manufacturing by addressing key aspects of **fairness**, **transparency**, and **sustainability**.

Fairness

While the MVTec AD dataset does not involve human subjects or demographic data, fairness is maintained by ensuring the model performs consistently across **all product categories** and **defect types**. To prevent bias toward specific textures or lighting conditions, the training process includes data normalization, augmentation, and cross-category validation. The project's evaluation metrics (e.g., ROC-AUC, F1-score) are applied uniformly to verify model reliability under varying conditions.

Transparency and Explainability

The project promotes transparency by using **visual explainability techniques**, such as reconstruction error maps and defect heatmaps, to clearly show *where* and *why* the system flags an image as defective. These visualizations allow non-technical users—such as manufacturing operators—to trust and interpret the model's decisions. All model parameters, code, and results are documented in the public GitHub repository for reproducibility and auditability.

Environmental Considerations

The model is designed with **computational efficiency** in mind. Using lightweight convolutional autoencoders instead of large transformer architectures minimizes GPU usage, reducing energy consumption and carbon footprint. Additionally, the system supports **real-time inference on local hardware**, limiting dependency on cloud computation and data transfer, further reducing environmental impact.

Ethical Deployment

In deployment, the model will serve as a **decision-support tool**, not a replacement for human oversight. Operators retain the final judgment on product quality, ensuring accountability and responsible integration of AI into manufacturing workflows.