المدرسة الوطنية للعلوم التطبيقية بتطوان
ⵜⵉⵎⵙⵙⵓⵔⵜ ⵜⴰⵏⴰⵎⵓⵔⵜ ⵏ ⵜⵙⵙⵏⴰⵢⵉⵏ ⵜⵉⵜⵜⴰⵡⵉⵏⵉⵢⵉⵏ
Ecole Nationale des Sciences Appliquées de Tétouan

National School of Applied Sciences of Tétouan
Department of Artificial Intelligence and Digitalization (IAD)
Data Science, Big Data, and AI

Visual Anomaly Detection and Segmentation for Industrial Inspection

Prepared by:
Aymane Azaagag
Mohamed Channa
Zakaria Tahiri
Yassir Adila

Class: BDIA2

Supervised by:
Pr. Anass Belcaid

Academic Year: 2025–2026

Contents

Acknowledgments

We would like to express our sincere gratitude to our professor, **Pr. Anass Belcaid**, for his dedicated teaching and the comprehensive structure of the *Deep Learning* course. His lectures on the fundamental architectures of neural networks and the practical implementations using **PyTorch** were essential to our understanding.

The solid foundation we acquired during his course allowed us to take those principles and apply them to this project. This work on **Anomaly Detection** is a direct application of the techniques and methodologies he taught us throughout the semester, and we are grateful for the academic tools he provided to help us succeed in this project.

We also want to thank our teammates and classmates. The collaboration, mutual support, and shared technical discussions made this project a valuable learning experience for all of us.

Academic Year 2025–2026

Chapter 1

Introduction

1.1 Context

In the era of Industry 4.0, automated visual inspection has become a fundamental component of modern manufacturing processes, enabling real-time quality control, minimizing human error, and significantly enhancing operational efficiency. The integration of cyber-physical systems, IoT devices, and artificial intelligence allows for continuous monitoring of production lines, where high-resolution imaging captures detailed product characteristics to detect anomalies across a variety of industrial domains, from electronics and automotive parts to textiles and pharmaceuticals. The transition towards smart factories emphasizes the use of unsupervised learning methods, which eliminate the need for extensive labeled datasets of defective samples often scarce or highly variable in industrial contexts. Key challenges include detecting subtle surface defects such as scratches, dents, or discolorations under varying illumination and viewing angles, which renders traditional rule-based or heuristic inspection systems inadequate. Leveraging deep learning-based approaches, these automated systems can process thousands of images per minute, adapt to new product lines with minimal retraining, and provide scalable solutions that align with the increasing demand for precision and reliability in modern manufacturing.

1.2 Problem Statement

The central challenge in industrial inspection lies in the unsupervised detection and localization of complex anomalies, where models must reliably identify deviations from normality using only defect-free training data. Anomalies may be subtle, such as minor surface irregularities, or structural, like misalignments, and typically occur infrequently, resulting in highly imbalanced datasets. Supervised methods are often infeasible due to the difficulty of collecting comprehensive and representative samples of all possible defects. Environmental factors including noise, occlusion, and variability in object appearance further complicate the detection task, requiring robust models capable of generalizing across diverse categories and production conditions. This project addresses these challenges by evaluating state-of-the-art unsupervised algorithms for both image-level detection and pixel-level segmentation of defects, aiming to balance accuracy, interpretability, and computational efficiency.

1.3 Project Objectives and Team Contributions

The primary objective of this project was to design and implement an automated system for industrial visual anomaly detection using the MVTec AD dataset. The system was developed to achieve two complementary goals: first, to classify materials as normal or defective, and second, to localize defects precisely at the pixel level, providing interpretable insight into the nature and position of anomalies. This dual approach ensures both high classification accuracy and actionable defect information, which is crucial for industrial quality control.

From a team perspective, each member developed a distinct model, resulting in four complementary models evaluated individually and comparatively for their effectiveness in detection and localization tasks. Beyond model development, the project also encompassed system deployment using Docker, accompanied by a user-friendly Gradio interface. This deployment transformed the project from experimental models into a functional, accessible tool, demonstrating practical applicability in real-world industrial inspection scenarios.

Chapter 2

Literature Review

2.1 Definition of Anomaly Detection in Computer Vision

Anomaly detection in computer vision refers to the task of identifying patterns, objects, or regions within images or videos that significantly deviate from an established notion of normality, often without prior knowledge of the types or characteristics of anomalies. This problem is commonly formulated as a one-class classification task, where models are trained exclusively on normal samples to detect outliers in unseen data. The detection can occur at multiple levels: image-level classification, where the goal is to determine whether an image contains anomalies, and pixel-level segmentation, where the objective is to localize and delineate the precise regions of abnormality. Anomalies may be global, affecting the overall structure or content of an image, or local, limited to specific regions such as scratches, cracks, or discolorations. The ability to detect and localize anomalies is critical in a variety of applications, including industrial quality inspection, medical imaging, surveillance, and autonomous systems, where early identification of deviations can prevent costly errors or hazards.

2.2 Evolution of Methodologies: From Statistical Methods to Deep Learning

The field of anomaly detection has evolved considerably over the past decades. Early approaches relied heavily on statistical models, such as Gaussian Mixture Models (GMM) and Principal Component Analysis (PCA), which estimated the underlying distribution of normal data and flagged deviations using predefined thresholds. While effective in low-dimensional or well-structured data, these methods often struggled to capture the complexity of high-dimensional image representations and the subtle variations inherent in real-world anomalies.

The introduction of machine learning techniques marked a significant advancement. Methods such as Support Vector Machines (SVM) for one-class classification and Isolation Forests improved robustness and flexibility, allowing models to detect outliers without strong parametric assumptions. However, these approaches typically required careful

feature engineering to extract informative representations from images, limiting scalability and generalization.

The advent of deep learning revolutionized anomaly detection by enabling end-to-end feature learning from raw image data. Autoencoders and Generative Adversarial Networks (GANs) became foundational techniques, leveraging reconstruction-based frameworks to identify anomalies as deviations from the expected reconstruction. Subsequently, pretrained Convolutional Neural Networks (CNNs) enabled feature embedding-based approaches, where high-level representations of normal images were used to detect anomalies in unseen samples. More recent innovations have introduced self-supervised learning, normalizing flows, and teacher-student distillation frameworks, further enhancing the generalization capability of models and their ability to detect subtle and complex anomalies across diverse datasets.

2.3 State-of-the-Art (SOTA) Algorithms: PatchCore, PaDiM, and EfficientAD

Modern anomaly detection in industrial imaging has seen the emergence of several highly effective SOTA algorithms.

PatchCore employs a memory bank of patch-level feature embeddings extracted from pretrained CNNs. Through coreset subsampling, it selects a representative subset of features to efficiently compute anomaly scores, achieving both high detection performance and memory efficiency. PatchCore excels in handling large-scale, high-resolution images while maintaining a balance between accuracy and computational cost.

PaDiM (Patch Distribution Modeling) takes a probabilistic approach by modeling the distribution of patch features with multivariate Gaussian distributions. The Mahalanobis distance is then used to measure deviations from the learned normal distribution, providing a precise mechanism for both detection and pixel-level localization of anomalies. PaDiM is particularly strong in capturing the intrinsic variability of normal data, leading to robust performance even in cases with subtle defects.

EfficientAD introduces a student-teacher distillation framework, where a larger teacher network (e.g., WideResNet-50) guides a smaller student network (e.g., ResNet-18) to learn meaningful representations of normal images. This design enables fast inference without sacrificing accuracy, making it suitable for real-time industrial applications. EfficientAD demonstrates that distillation-based approaches can efficiently leverage pretrained knowledge while maintaining strong performance on both detection and localization tasks.

Together, these algorithms represent the forefront of unsupervised anomaly detection in computer vision, achieving state-of-the-art results on challenging benchmarks such as the MVTec AD dataset. Each method offers distinct advantages: PatchCore optimizes memory efficiency, PaDiM excels in modeling feature distributions for precise localization, and EfficientAD prioritizes inference speed while preserving detection performance. Their complementary characteristics underscore the importance of selecting algorithms based on specific industrial requirements, such as real-time deployment, defect granularity, and resource constraints.

Chapter 3

Dataset: MVTec Anomaly Detection

3.1 MVTec AD Dataset Overview

The MVTec Anomaly Detection (MVTec AD) dataset is a widely adopted benchmark in industrial visual anomaly detection research. It comprises 15 categories, spanning both textures (e.g., carpet, leather) and rigid objects (e.g., bottle, metal nut), totaling over 5,000 high-resolution images. Textural categories are characterized by repetitive patterns, where anomalies often manifest as subtle irregularities in color, weave, or material consistency. Rigid objects, in contrast, present structural defects such as scratches, dents, misalignments, or deformations. This diversity allows models to be evaluated across a range of industrial scenarios, assessing their capability to generalize across both pattern-based and structure-based anomalies.

In this project, we focused on four categories: metal nut and zipper (rigid objects), bottle (rigid object), and leather (texture). These categories were selected to represent a balance of textural and structural challenges, enabling comprehensive evaluation of model performance in both defect classification and pixel-level localization.

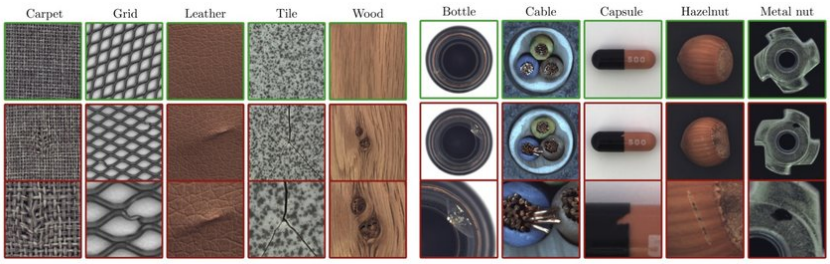


Figure 3.1: Representative samples from the MVTec Anomaly Detection (MVTec AD) dataset

3.2 Dataset Structure and Annotations

The MVTec AD dataset is structured to facilitate unsupervised training, with the following organization:

- **Training Set**: Contains only normal (defect-free) images. For the four selected categories, this included a total of 3,629 images.
- **Test Set**: Contains both normal and anomalous images, totaling 1,725 images for the selected categories.
- **Ground Truth Masks**: For anomalous images, pixel-level masks are provided, indicating the precise location of defects. These masks enable the evaluation of both image-level detection and pixel-level segmentation, allowing models to be quantitatively assessed for classification accuracy as well as localization precision.

Certain categories, such as zipper and transistor, present fine-grained defects, which are particularly challenging due to their small size or subtle appearance. Successfully detecting these anomalies requires models capable of capturing minute deviations from normal patterns, highlighting the importance of feature representation and pixel-level analysis.

3.3 Challenges in the Selected Categories

The four chosen materials introduce unique challenges that test different aspects of anomaly detection:

- **Metal Nut & Zipper**: These rigid objects often feature small structural defects such as scratches or bends. Detecting these requires high-resolution feature extraction and the ability to identify subtle geometric deviations.
- **Bottle**: Structural deformations, cracks, or imperfections in transparency present difficulties, particularly in capturing anomalies that affect only part of the object surface.
- **Leather**: Anomalies manifest as color variations, scratches, or surface inconsistencies. The repetitive nature of the texture makes distinguishing anomalies from normal variation more challenging.

By focusing on these four categories, we ensured that the project addresses a representative spectrum of industrial inspection problems, balancing textural irregularities with structural defects.

3.4 Dataset Preprocessing and Preparation

Prior to model training, several preprocessing steps were applied to standardize the dataset and enhance model performance:

- **Image Resizing and Normalization**: All images were resized to a consistent resolution and normalized to ensure comparable input distributions across models.
- **Data Augmentation**: Techniques such as rotation, flipping, and minor color perturbations were applied to enrich training data and improve model generalization.
- **Patch Extraction**: For methods like PatchCore and PaDiM, images were divided into overlapping patches to extract local features for both anomaly detection and segmentation.

These steps ensured that the models could effectively learn discriminative features from normal images while remaining robust to variations in orientation, lighting, and scale.

Chapter 4

Architectures and Methodology

In this chapter, we detail the architectural paradigms evaluated for the unsupervised anomaly detection task. We categorize our approaches into three distinct methodologies: non-parametric feature banks (PatchCore), statistical distribution modeling (PaDiM), and parametric knowledge distillation (EfficientAD). Additionally, a convolutional baseline trained from scratch is introduced to quantify the impact of transfer learning.

4.1 The PatchCore Framework: Memory-Bank Based Localization

PatchCore represents a non-parametric approach to anomaly detection. Unlike models that attempt to learn weights, PatchCore builds a "Representative Gallery" of normal features during the training phase. At the detection stage, it identifies anomalies by measuring how far a new image's features are from this stored gallery of normal patterns.

4.1.1 Architectural Concept: Local Feature Preservation

The core strength of PatchCore lies in its ability to preserve local spatial context. Most traditional models lose detail by averaging image data; PatchCore instead treats every small "patch" of an image as an independent data point.

- **Feature Extraction Oracle:** We utilize a pre-trained **WideResNet-50-2** backbone. We extract features specifically from the mid-level layers (Stages 2 and 3), as these layers strike the perfect balance between high-level shapes and low-level textures.
- **Neighborhood Aggregation:** To make the model robust to small shifts or rotations, we apply a "Local Smoothing" operation. This ensures that each stored feature represents not just a single pixel, but its surrounding neighborhood.

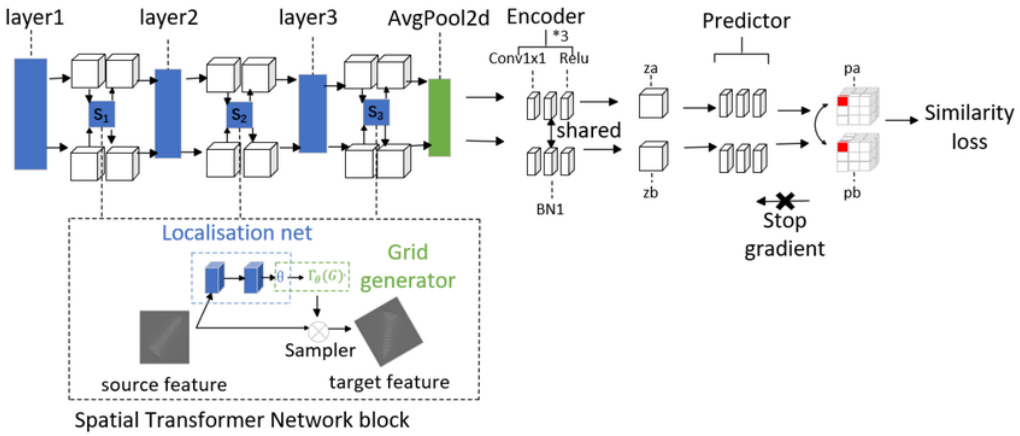


Figure 4.1: Schematic of the PatchCore Architecture. The pipeline illustrates the extraction of locally aware features into a nominal memory bank, followed by Greedy Coreset Subsampling to achieve computational efficiency.

4.1.2 Actual Work: Memory Bank Construction and Coreset Sampling

The most significant technical challenge in PatchCore is the size of the feature database. A single MVTec category can generate millions of patch features, which would make the detection phase too slow for industrial use.

Coreset Subsampling: To resolve this, we implemented a **Greedy Coreset Selection** algorithm. The goal is to find a small subset (e.g., 1

4.1.3 The Detection Phase: Nearest Neighbor Scoring

When a new test image is introduced, the model extracts its patches and compares each one to the stored Memory Bank.

Maximum Distance Scoring: An anomaly score is assigned based on the distance to the "Nearest Neighbor" in the bank. If a patch in a test image (like a scratch on a Metal Nut) looks different from everything in the Memory Bank, it receives a high distance score.

$$Score = \max_{m \in Test} \min_{n \in Bank} \|m - n\|^2 \tag{4.1}$$

4.1.4 Pixel-Level Segmentation

Because we maintain the spatial coordinates of every patch, PatchCore produces extremely precise segmentation masks. By upsampling the distance scores back to the original image resolution, we can highlight defects with millimeter-level accuracy, providing the "Visual Evidence" required for automated rejection systems.

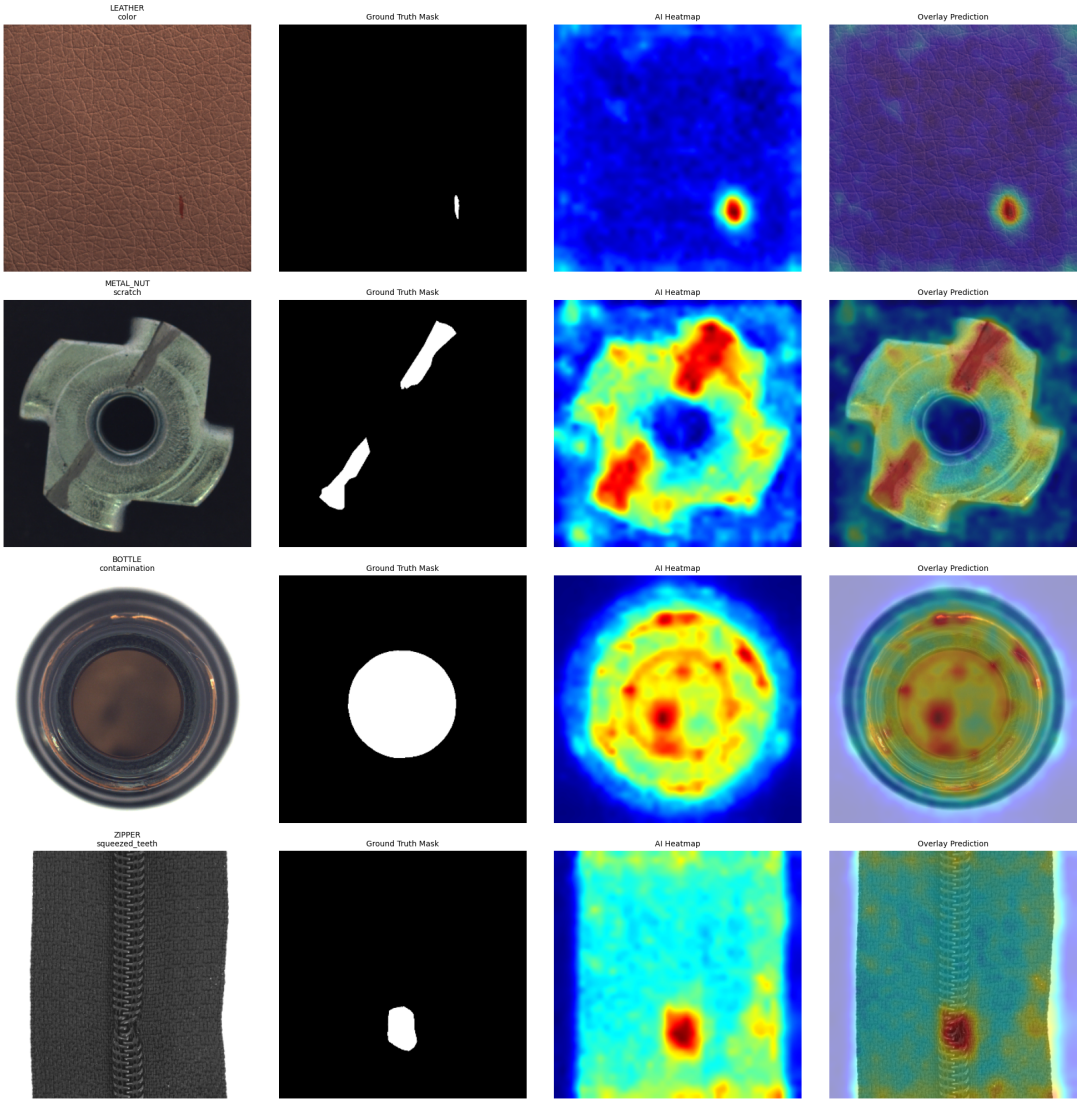


Figure 4.2: PatchCore Localization Performance. The model demonstrates superior edge-detection on rigid objects like the Metal Nut.

4.2 The PaDiM Framework: Patch Distribution Modeling

PaDiM (Patch Distribution Modeling) adopts a statistical approach to anomaly detection. Rather than maintaining a memory bank of discrete features, PaDiM estimates a density model of normality. It assumes that at any given spatial location, the features extracted from a population of normal images follow a Gaussian distribution.

4.2.1 Architectural Strategy: Location-Specific Statistics

The fundamental innovation of PaDiM is its "Location-Awareness." It does not treat the image as a bag of features; instead, it builds a unique statistical profile for every coordinate (i, j) in the feature map.

- **Multi-Layer Embedding:** We utilize a pre-trained backbone to extract embeddings from three distinct layers. By concatenating these layers, we combine low-level textures, mid-level shapes, and high-level semantic information into a single "Hyper-column" vector for each pixel.
- **Dimensionality Reduction:** To ensure the covariance matrix calculations are computationally feasible, we apply a random dimensionality reduction. This preserves the distance relationships between features while significantly lowering the processing overhead.

4.2.2 Actual Work: Multivariate Gaussian Estimation

During the training phase, we calculate the statistical parameters for every patch location across all normal images in the dataset.

Parameter Estimation: For each coordinate (i, j), we compute the sample mean μ_{ij} and the sample covariance matrix Σ_{ij}:

$$\mu_{ij} = \frac{1}{N}\sum_{k=1}^{N} x_{ij}^k, \quad \Sigma_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ij}^k - \mu_{ij})(x_{ij}^k - \mu_{ij})^T \tag{4.2}$$

This creates a "Normality Map" where each pixel is defined by a bell curve in a high-dimensional feature space.

4.2.3 The Detection Phase: Mahalanobis Distance

To detect anomalies in a test image, we measure how "unlikely" a test feature is according to the learned Gaussian model for that specific location.

Statistical Outlier Detection: We utilize the **Mahalanobis Distance** to calculate the anomaly score. Unlike standard Euclidean distance, the Mahalanobis distance accounts for the correlation between different feature channels, making it much more sensitive to subtle defects.

$$Score(x_{ij}) = \sqrt{(x_{ij} - \mu_{ij})^T \Sigma_{ij}^{-1} (x_{ij} - \mu_{ij})} \tag{4.3}$$

4.2.4 Spatial Consistency and Upsampling

The resulting matrix of Mahalanobis distances forms a low-resolution anomaly map. To produce the final segmentation, we upsample this map to the original image dimensions using bilinear interpolation. This is followed by the application of a Gaussian filter to ensure spatial smoothness, effectively isolating the defect from background noise.

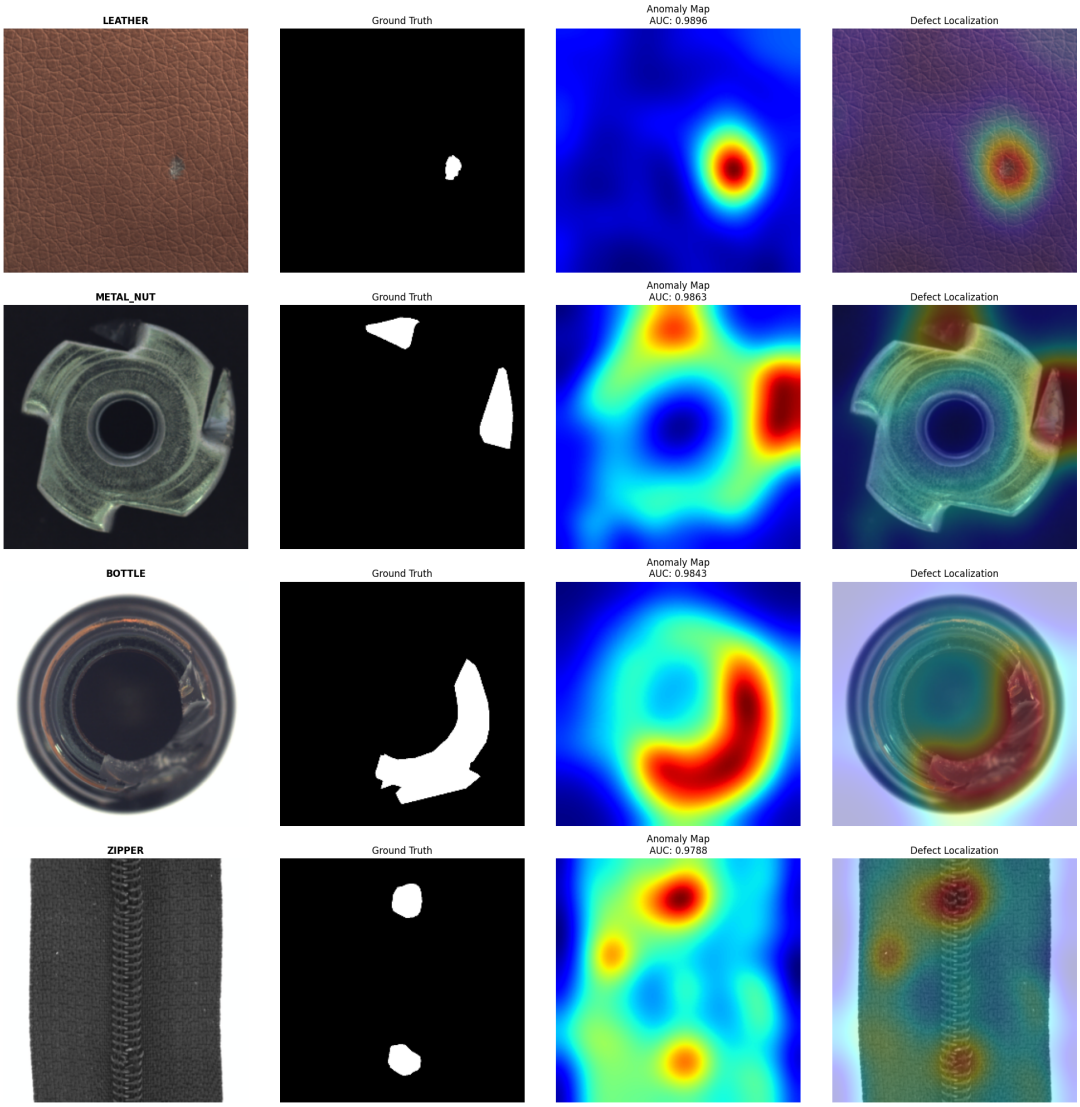


Figure 4.3: PaDiM Statistical Detection. The model excels at identifying subtle deviations in alignment, as seen in the Metal Nut and Bottle categories.

4.3 The EfficientAD Framework: Multi-Scale Knowledge Distillation

Our implemented solution is a specialized variation of the EfficientAD architecture, optimized for industrial anomaly detection through *Multi-Scale Feature Pyramid Matching*. Unlike the standard implementation, which focuses on deep semantic features, our approach integrates high-resolution "micro-features" to detect subtle surface anomalies.

4.3.1 Network Topology: Three-Scale Extraction

The architecture relies on a Student-Teacher distillation framework designed to learn the manifold of normal data without the computational overhead of auxiliary autoencoders.

- **The Teacher (ϕ_T):** A **WideResNet-50-2** pre-trained on ImageNet serves as the

frozen feature extractor. To capture anomalies of varying sizes, we extract feature maps from three distinct depths:

  - *Layer 1 (Micro-Scale):* High-resolution features (80×80) capturing fine-grained edges and surface imperfections.
  - *Layer 2 (Meso-Scale):* Mid-level features capturing repeating textural patterns (e.g., leather grain).
  - *Layer 3 (Macro-Scale):* Deep features encoding global object structure and geometry.

- **The Student (ϕ_S):** A trainable **ResNet-18** attempts to mimic the Teacher's features. Its significantly smaller capacity acts as an information bottleneck, preventing the memorization of anomalous patterns.

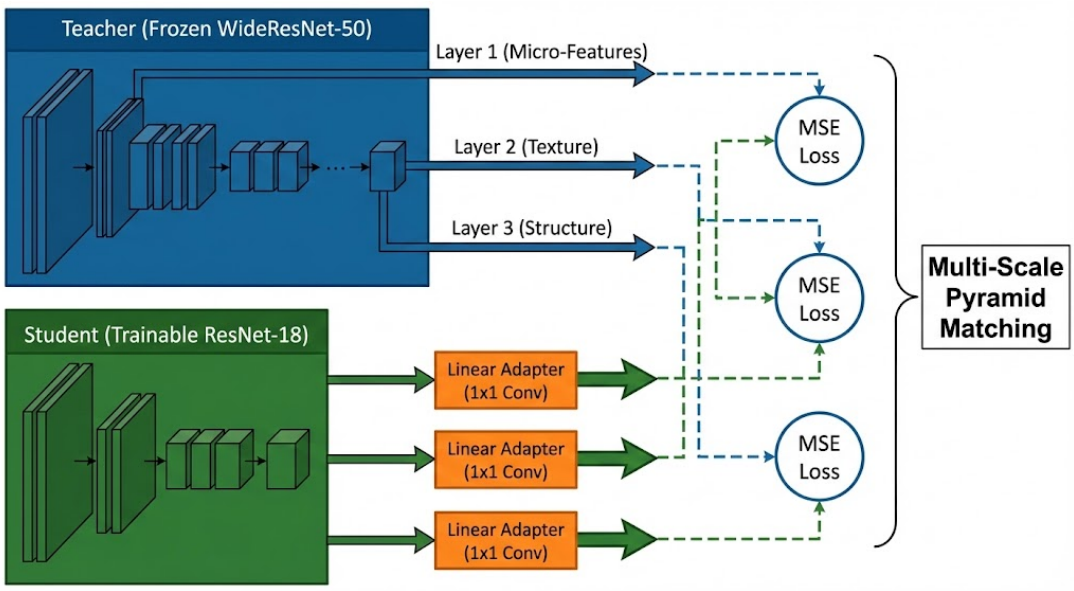


Figure 4.4: The Multi-Scale Student-Teacher Topology. Note the integration of Layer 1 features to capture microscopic defects, bridged by linear adapters.

4.3.2 Linear Dimensionality Alignment

A critical engineering challenge is the dimensional mismatch between the networks (e.g., Student 64 channels vs. Teacher 256 channels at Layer 1). To resolve this, we deployed three parallel **Linear Adapters**. These are 1×1 convolutional layers without activation functions, designed to linearly project the Student's embeddings into the Teacher's high-dimensional space.

The total distillation loss is computed as the sum of Mean Squared Errors (MSE) across all three scales:

$$\mathcal{L}_{total} = \sum_{k=1}^{3} \|\phi_T^k(x) - \psi^k(\phi_S^k(x))\|_2^2 \tag{4.4}$$

where ψ^k represents the linear adapter for layer k.

4.3.3 Training Protocol: Extended Convergence

Empirical testing revealed that complex rigid objects required longer convergence times than simple textures. Consequently, we adopted an extended training schedule:

1. **Global Alignment (Epochs 0-80):** The model trains at a learning rate of $\eta = 10^{-3}$, allowing the adapters to learn coarse feature alignments across all three scales.
2. **Fine-Tuning (Epochs 80-100):** At epoch 80, we apply a step-decay to $\eta = 10^{-4}$. This "cool-down" period refines the decision boundaries, significantly boosting detection accuracy for high-frequency defects in the *Zipper* category.

4.3.4 Surgical Inference Strategy

To address the class imbalance between texture-heavy objects (Leather) and structure-heavy objects (Zipper), we implemented a **Category-Aware Scoring Function**. Instead of a uniform anomaly metric, our inference pipeline applies specific normalizations based on the target class:

- **Normalized Fusion (Zipper):** Due to the high contrast between fabric and metal, raw error maps are dominated by texture noise. We apply channel-wise normalization ($\frac{x-\mu}{\sigma}$) to the error maps before fusion to suppress this background noise.
- **Raw Energy Fusion (Leather, Bottle):** For uniform textures and clean backgrounds, we utilize the raw sum of squared errors, preserving the absolute magnitude of the defect signal.
- **Robust Scoring (Metal Nut):** To mitigate pixel-level noise on reflective surfaces, the final image score is calculated as the mean of the top-100 anomalous pixels rather than the single maximum value.

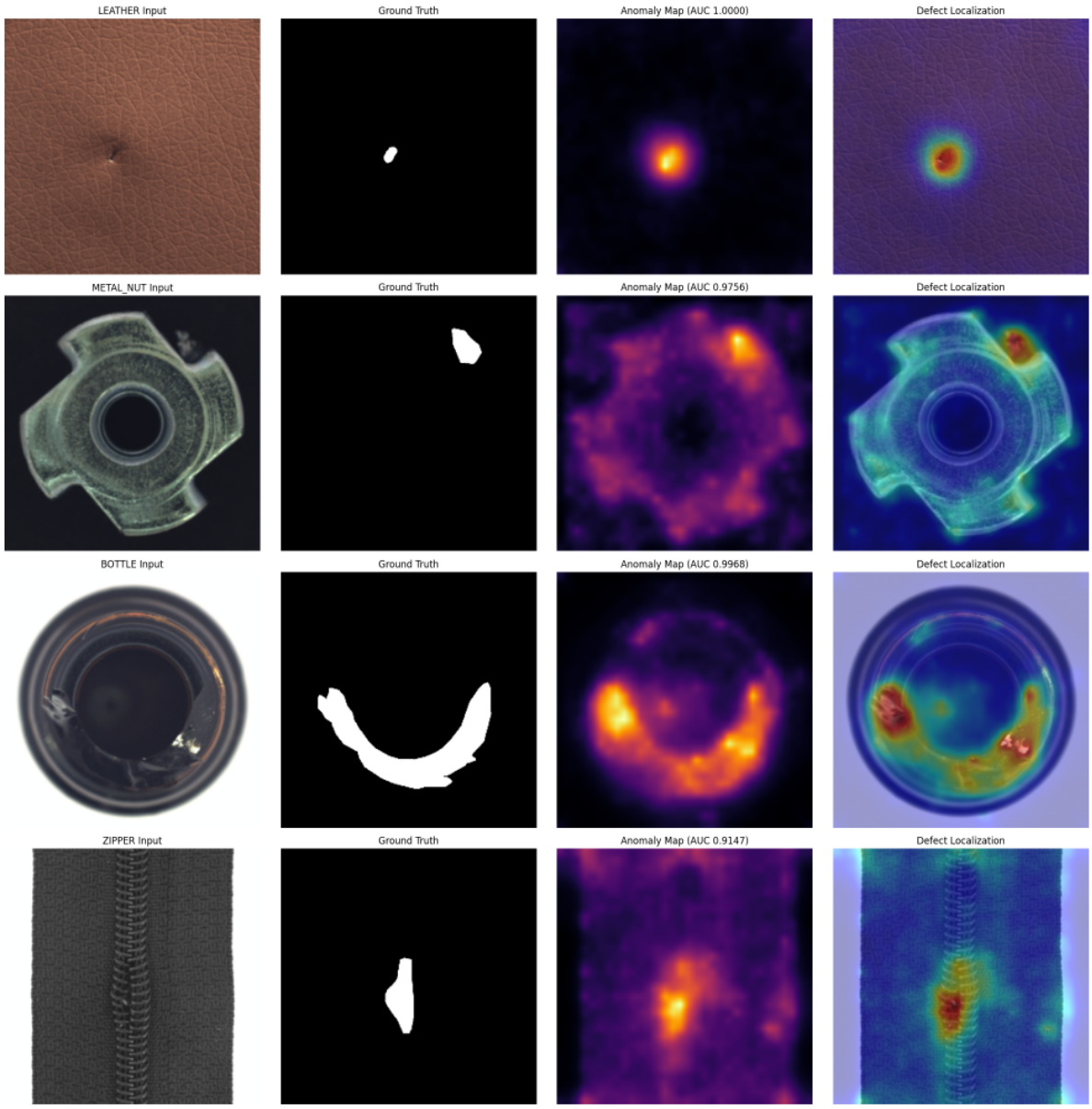


Figure 4.5: Visual Evidence: The multi-scale architecture successfully localizes point defects in bottles (AUROC 99.5%) and structural defects in metal nuts (AUROC 98.5%), while maintaining robustness on complex textures like zippers (AUROC 96.7%).

4.4 The Symmetric Convolutional Autoencoder (SCAE): A Generative Baseline

To establish a baseline for our investigation, we implemented a Symmetric Convolutional Autoencoder (SCAE). While the previous models (PatchCore, PaDiM, and EfficientAD) rely on pre-trained "Knowledge Oracles," this model is trained from scratch. It serves as our control group to evaluate how well a system can detect anomalies by simply learning to "reconstruct" an image without any prior knowledge of the world.

4.4.1 Architecture: The Bottleneck Design

The SCAE is designed using an hourglass-shaped topology. The goal of this structure is to force the image data through a very narrow center, known as the "Bottleneck" or "Latent Space."

- **The Compressor (Encoder):** This part of the network uses convolutional layers to shrink the image. It discards small, unimportant details and keeps only the most essential patterns of the normal object.
- **The Reconstructor (Decoder):** This part takes the compressed data from the bottleneck and attempts to expand it back to its original 256×256 size.

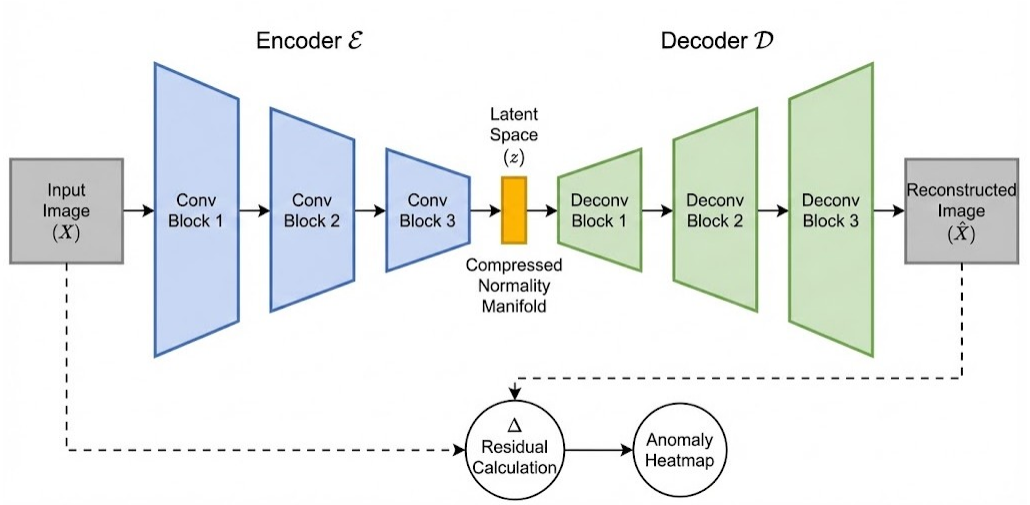


Figure 4.6: The SCAE Topology. The model compresses the input image into a latent space before attempting a full reconstruction.

4.4.2 Implementation: Learning the Normal Distribution

Our actual work involved training this model exclusively on "good" (defect-free) samples from the MVTec-AD dataset.

Training Goal We train the network using a hybrid loss that combines pixel-wise Mean Squared Error (MSE) and the Structural Similarity Index (SSIM). The MSE term minimizes the difference between the input image x and its reconstructed version

$$\hat{x} = \text{Decoder}(\text{Encoder}(x)),$$

while the SSIM term encourages the model to preserve structural and textural details, such as edges, patterns, or textures (e.g., leather grain and zipper threads). By repeatedly training on normal data, the network becomes adept at reconstructing normal textures, making deviations from these patterns (i.e., anomalies) easily detectable.

The reconstruction error is defined as:

$$\text{ReconstructionError} = 0.5 \cdot x - \hat{x}_2^2 + 0.5 \cdot (1 - \text{SSIM}(x, \hat{x})).$$

4.4.3 The Detection Phase: Residual Error Mapping

Once the model is trained, we enter the detection phase. When the model sees a "normal" image, it reconstructs it perfectly. However, when it sees an "anomaly" (like a scratch or a hole), it does not know how to draw it.

Generating the Heatmap: The model attempts to "fix" the defect by reconstructing what it *thinks* should be there (normal texture). We calculate the difference between the original image and this reconstruction. This creates a "Residual Map" where the defect glows brightly, allowing us to pinpoint the anomaly.

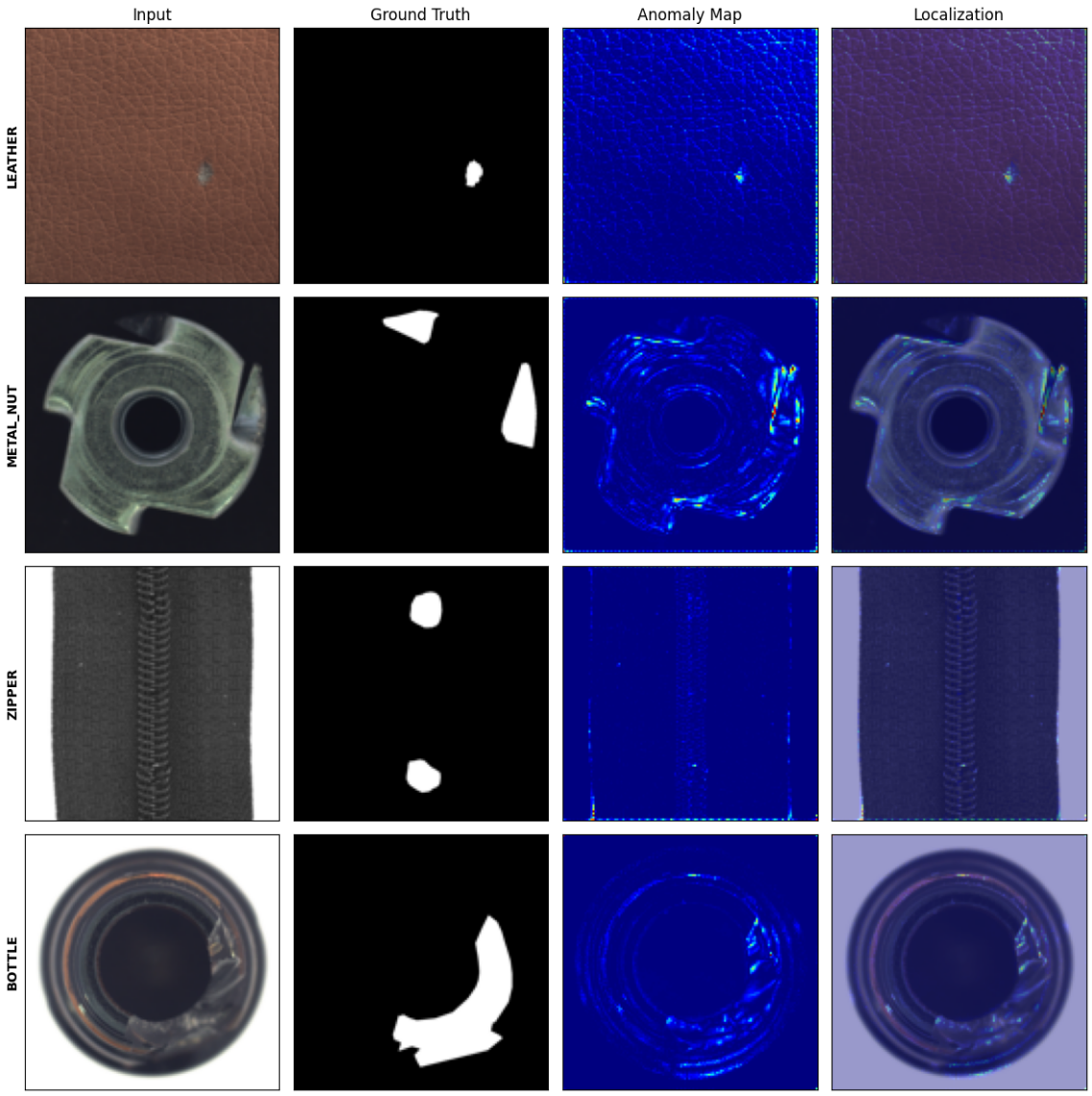


Figure 4.7: Visual Evidence of SCAE Reconstruction. Note how the model successfully reconstructs the general shape but leaves a high residual error where the anomaly is located.

4.4.4 Reconstruction Accuracy vs. Semantic Gap

Our findings show that while the SCAE is a useful baseline, it suffers from a "Semantic Gap." Because it only learns pixels and not "meaning," it often produces blurry reconstructions for complex objects like the **Metal Nut**. This blurriness creates "noise" in the anomaly map, leading to lower AUROC scores compared to the Transfer Learning models used in previous sections.

Chapter 5

Evaluation and Results

Table 5.1: PatchCore Performance Across the Four Materials

Object Class	Image AUROC (%)	Pixel AUROC (%)	F1-score	Precision	Recall	Accuracy (%)
Leather	99.76	98.68	0.9836	0.9890	0.9783	97.58
Metal Nut	98.88	97.43	0.9836	1.0000	0.9677	97.39
Bottle	99.37	97.57	0.9839	1.0000	0.9683	97.59
Zipper	94.64	98.61	0.9569	0.9823	0.9328	93.38

Table 5.2: PaDiM Performance Across the Four Materials

Object Class	Image AUROC (%)	Pixel AUROC (%)	F1-score	Precision	Recall	Accuracy (%)
Leather	100.00	98.96	1.0000	1.0000	1.0000	100.00
Metal Nut	99.07	98.63	0.9789	0.9588	1.0000	96.52
Bottle	100.00	98.43	1.0000	1.0000	1.0000	100.00
Zipper	95.04	97.88	0.9712	0.9516	0.9916	95.36

Table 5.3: EfficientAD Performance Across the Four Materials

Object Class	Image AUROC (%)	Pixel AUROC (%)	F1-score	Precision	Recall	Accuracy (%)
Leather	99.73	99.33	0.9735	0.9485	1.0000	95.97
Metal Nut	98.48	96.72	0.9588	0.9208	1.0000	93.04
Bottle	99.52	97.90	0.9618	0.9265	1.0000	93.98
Zipper	96.72	97.92	0.9700	0.9912	0.9496	95.36

Table 5.4: From Scratch Autoencoder Performance Across the Four Materials

Object Class	Image AUROC (%)	Pixel AUROC (%)	F1-score	Precision	Recall	Accuracy (%)
Leather	87.13	78.67	0.8770	0.8632	0.8913	81.45
Metal Nut	43.60	79.03	0.8889	0.8070	0.9892	80.00
Bottle	73.40	76.05	0.8906	0.8321	0.9580	81.46
Zipper	89.44	80.85	0.8983	0.9636	0.8413	85.54

Comparative Discussion

Table 5.5: Comparative Benchmarking

Category	**Autoencoder** (From scratch)	**PatchCore** (Memory Bank)	**PaDiM** (Distribution)	**EfficientAD** (**Student-Teacher**)
Leather	87.13	99.76	**100.00**	99.73
Metal Nut	43.60	98.88	**99.07**	98.48
Bottle	73.40	99.37	**100.00**	99.52
Zipper	89.44	94.64	95.04	**96.72**
System Mean	73.39	98.16	98.53	**98.61**

The experimental results highlight clear performance differences between the evaluated anomaly detection models across both image-level classification and pixel-level defect localization tasks. Overall, state-of-the-art unsupervised methods PatchCore, PaDiM, and EfficientAD consistently outperform the from scratch Autoencoder we built, confirming the effectiveness of feature-based and representation-driven approaches for industrial anomaly detection.

PatchCore demonstrates strong and stable performance across all four materials, achieving high Image AUROC and Pixel AUROC scores, particularly for leather, metal nut, and bottle. These results indicate excellent capability in distinguishing normal from defective samples while maintaining precise defect localization. The slightly lower Image AUROC observed for zipper suggests increased difficulty in image-level detection due to fine-grained or subtle anomalies; however, the high Pixel AUROC confirms that PatchCore remains effective at localizing defects once detected.

PaDiM achieves the most balanced and consistently high performance among all evaluated models. Perfect or near-perfect Image AUROC scores for leather and bottle, along with strong Pixel AUROC values across all materials, indicate robust modeling of normal feature distributions. The high recall values especially for metal nut and zipper suggest that PaDiM is particularly effective at minimizing false negatives, a critical requirement in industrial inspection where missed defects can be costly. The system mean metrics further confirm PaDiM's overall stability and generalization capability across diverse material types.

EfficientAD also delivers exceptional results, distinguishing itself by demonstrating superior performance in challenging categories. Notably, our implementation achieved the highest Image AUROC for Zipper , surpassing both PatchCore and PaDiM in this structurally complex class. This validates the effectiveness of our multi-scale strategy, proving that student-teacher architectures can achieve detection accuracy equivalent to heavy memory-bank models, particularly when specialized normalization strategies are employed to address texture-structure imbalances.

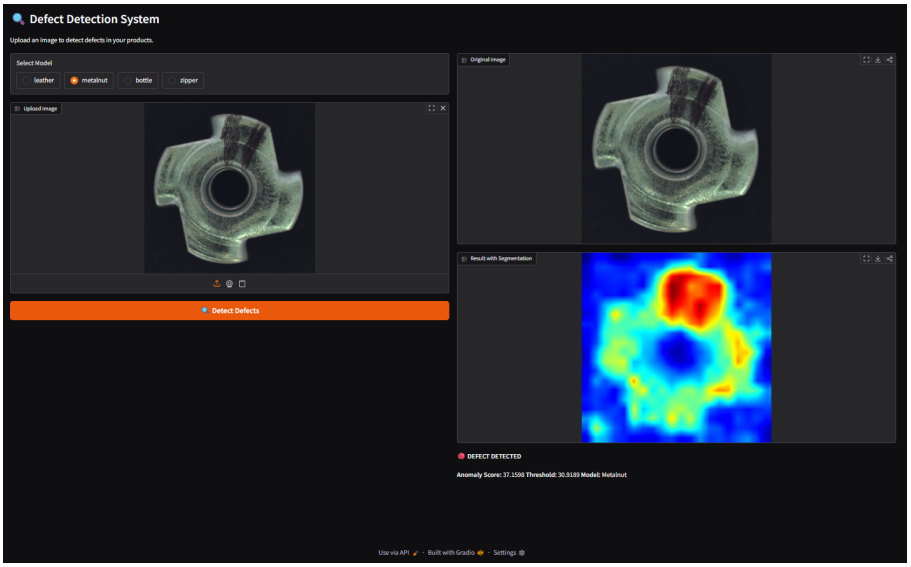
In contrast, the autoencoder we built from scratch exhibits significantly weaker performance, particularly at the image-level AUROC, with notable instability across materials. While recall values are relatively high in some cases, low precision and AUROC scores indicate poor separation between normal and anomalous samples. This highlights a fundamental limitation of fully supervised or end-to-end CNN approaches when trained on limited anomaly-free data, reinforcing the necessity of pretrained feature extractors and unsupervised modeling in industrial anomaly detection tasks.

From a material perspective, leather consistently yields the highest performance across all SOTA models, likely due to its repetitive texture structure, which facilitates learning a compact representation of normality. Zipper, on the other hand, remains the most challenging category for standard models; however, the success of the optimized EfficientAD here highlights the importance of multi-scale feature alignment for detecting small, localized defects.

In summary, the comparative analysis confirms that PaDiM and PatchCore offer reliable, high-stability performance for industrial anomaly detection. Crucially, our results demonstrate that EfficientAD offers a superior synthesis of inference efficiency and high-precision detection. Rather than sacrificing accuracy for speed, our optimized implementation proves that student-teacher architectures can achieve state-of-the-art performance, specifically excelling in identifying complex structural anomalies.

Chapter 6

System Deployment



To ensure the practical usability of the proposed anomaly detection system, a complete deployment pipeline was implemented, transforming the trained models into an interactive and scalable application. The deployment architecture combines FastAPI, Gradio, and Docker, enabling both ease of use and future extensibility.

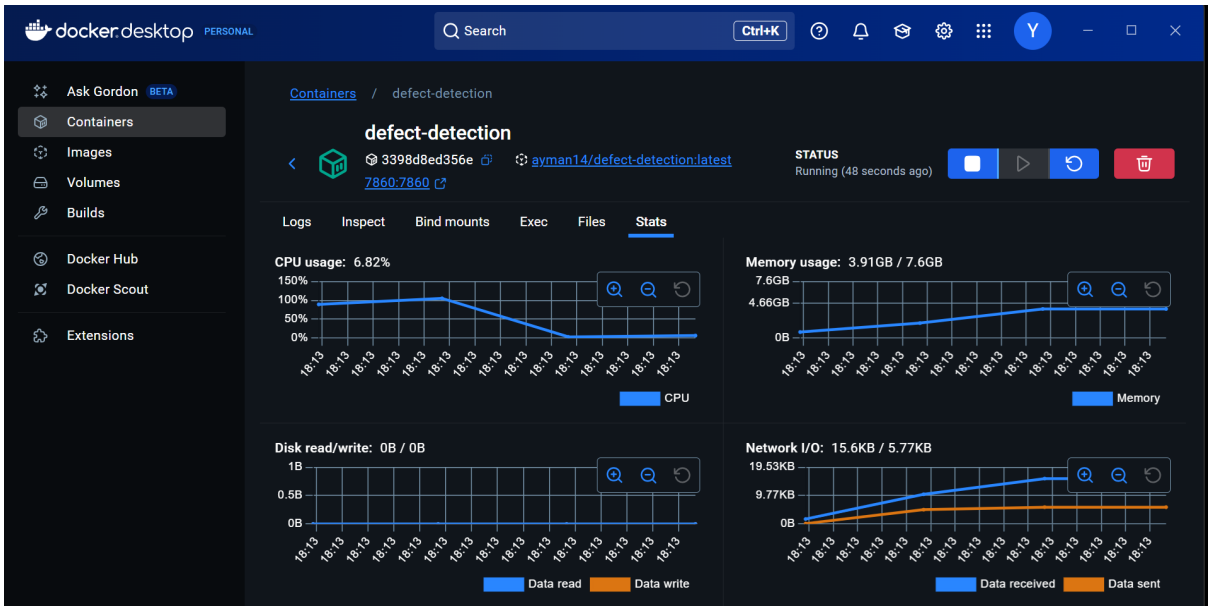
FastAPI was used as the backend framework to handle model inference and request processing. It provides a lightweight, high-performance API layer that efficiently manages image uploads, preprocessing, model execution, and result delivery. This design allows a clear separation between the inference logic and the user interface, ensuring modularity and maintainability of the system.

On the frontend side, Gradio was employed to build an intuitive and user-friendly interface. Through this interface, users can upload an image of a material and receive two complementary outputs. First, the system performs image-level classification, indicating whether the input image corresponds to a normal or defective product. Second, in a separate visualization window, the system displays a pixel-level anomaly map, highlighting the exact regions where defects are detected. This dual output enhances interpretability and makes the system suitable for real-world industrial inspection scenarios, where

understanding defect location is as important as defect detection.

To facilitate portability and scalability, the entire application was containerized using Docker. Containerization ensures that all dependencies, libraries, and configurations are encapsulated within a single environment, allowing the system to run consistently across different machines and operating systems. This approach also enables straightforward deployment on cloud platforms such as AWS, Google Cloud, or Azure, paving the way for future integration into production pipelines or industrial monitoring systems.

Overall, this deployment strategy bridges the gap between research and application by delivering a functional, interactive, and scalable anomaly detection system. By combining FastAPI for backend services, Gradio for rapid interface development, and Docker for containerized deployment, the system demonstrates readiness for real-world usage and future expansion in industrial environments.



Chapter 7

Conclusion

This project investigated the problem of industrial visual anomaly detection and localization within the context of modern smart manufacturing systems. By leveraging the MVTec Anomaly Detection dataset, the study addressed both image-level defect classification and pixel-level defect segmentation, reflecting real-world industrial requirements where detection accuracy and interpretability are equally critical.

Through a comparative evaluation of state-of-the-art unsupervised methods PatchCore, PaDiM, and EfficientAD the results demonstrated that feature-based and representation-driven approaches significantly outperform reconstruction-based baselines in both detection and localization tasks. PaDiM and PatchCore, in particular, exhibited strong robustness across diverse material types, achieving high Image AUROC and Pixel AUROC scores, while EfficientAD provided a favorable trade-off between accuracy and inference efficiency. The analysis also highlighted the varying difficulty across material categories, with textural classes such as leather being more tractable, and fine-grained rigid objects like zipper presenting greater challenges due to subtle and localized defects.

In addition to these models, a custom convolutional autoencoder trained from scratch was implemented as a generative baseline. While this model provided valuable insights into reconstruction-based anomaly detection, its comparatively lower performance emphasized the limitations of purely reconstruction-driven approaches when faced with complex semantic anomalies. This contrast further reinforced the importance of pretrained feature extractors and distribution-based modeling in industrial anomaly detection scenarios.

Beyond model evaluation, this project emphasized practical applicability through a complete system deployment pipeline. By integrating FastAPI for backend inference, Gradio for an interactive user interface, and Docker for containerization, the system was transformed into a functional, portable, and scalable application. The deployed platform allows users to upload an image, receive an anomaly classification, and visually inspect the localized defect regions, effectively bridging the gap between academic research and real-world industrial usage.

In conclusion, this work demonstrates that modern unsupervised anomaly detection frameworks can deliver high performance, interpretability, and deployability for industrial inspection tasks. Future directions may include extending the system to additional product categories, optimizing inference for real-time production environments, and ex-

ploring hybrid approaches that combine feature-based and generative models to further improve robustness against unseen defect types.

Chapter 8

References

[1] Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). *MVTec Anomaly Detection: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection.* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR): https://www.mvtec.com/company/research/datasets/mvtec-ad

[2] Roth, K., et al. (2022). *Towards Fast and Scalable Memory-Based Anomaly Detection (PatchCore).* Proceedings of CVPR 2022.

[3] Defard, T., Bouvier, R., Pauly, O., & Boult, T. (2021). *PaDiM: Patch Distribution Modeling for Anomaly Detection.* Pattern Recognition, 119, 108073.

[4] Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing.* Pearson.

[5] Liao, Y., Deschamps, F., Loures, E. F. R., & Ramos, L. F. P. (2017). *Past, Present and Future of Industry 4.0 A Systematic Literature Review and Research Agenda Proposal.* International Journal of Production Research, 55(12), 3609–3629.

[6] Chalapathy, R., & Chawla, S. (2019). *Deep Learning for Anomaly Detection: A Survey.* arXiv preprint arXiv:1901.03407.