POLITECNICO DI TORINO

Master's Degree in DATA SCIENCE AND ENGINEERING

Master's Degree Thesis

Anomaly Detection

Supervisors

Santa DI CATALDO

Francesco PONZIO

Alessio MASCOLINI

Candidate

Hesam KHANJANI

APRIL 2025

Summary

The detection of anomalies in the image data is crucial for many real-time computer vision applications, as it is directly related to triggering an alarm for security threats, quality inspection and production on manufacturing lines. Our goal in this thesis is to create a stable customizable model to predict anomalies such as scratches, scuffs, corrosion, etc. We used a multi-model approach to have all advantages of different approaches. Hence, we implemented base on EfficientAD[1] student-teacher type method, where the student learns the distribution of normal images. When it collapses, it is detected as an anomaly. So called changed instruction hypothesis avoid the student from recasting heaps of pictures, diminished measurements and expanded exactness. Furthermore, in another method called reconstruction-based approach we leveraged autoencoder to learn out the distribution of the normal image [1]. Autoencoder provides the anomaly detection that the reconstructed images and highlight the area that does not conform to the normal pattern, so as to achieve the abnormal localization of the parts. This could enable CNNs to prioritize improving detail in a useful way, potentially boosting both the speed and accuracy of candidates for future applications like anomaly detection for a spectrum of scenarios. Overgeneralization is one of the must challenging problems during dealing with anomaly detection in unsupervised strategies. So we tried to add syntetic anomaly to our training and use two loss function in this order. One is to maximize the distance between the normal sample and the nearest abnormal sample (wider than abnormal samples) and second one minimize the overlap between positive and negative space based on Collaborative Discrepancy Optimization [2]. The method has been tested with the Real-IAD[3] dataset on sub-dataset plastic-nut for incidents. This not only a good baselines for pattern recognition but also a good threshold for defect detection in various industries and can be navigate cross sectors on that basis to find region of anomalies and ensure quality during manufacturing of such components by providing a significant data input for that analysis. Our model achieved good performance of finding anomaly region with AUPRO 95.48% and ability to classify abnormal images AUROC 97.07%.

Acknowledgements

We owe our deepest gratitude to Prof. Santa di Cataldo, Francesco Ponzio, and Alessio Mascolini for offering such beneficial guidance, continuous encouragement, and wise suggestions in planning and preparation for this work. With patience, encouragement, and profound expertise, they have played a significant role in shaping this work, and for providing them with valuable time and expertise, we appreciate them immensely. We are also profoundly grateful to Blue Engineering S.R.L. for providing us with the opportunity and resources necessary to conduct this thesis. Their support has been invaluable in facilitating our research. We also owe a deep sense of gratitude to our family and friends for constant encouragement and motivation, whose presence has played a significant role in keeping us motivated during and even post our academic life. We would, in conclusion, appreciate expressing our deepest gratitude towards Politecnico di Torino for providing such a rich and enriching experience and for granting us such a wealth of information during our studies.

Table of Contents

List of Tables

List of Figures

Acronyms

IAD
Industrial Anomaly Detection

AI
Artificial Intelligence

OOD
Out-of-Distribution

CDO
Collaborative Discrepancy Optimization

BTAD
Bottle, Tile, and Anomaly Detection dataset

MPDD
Magnetic Particle Defect Detection dataset

MTD
Metal Texture Defects dataset

UIAD
Unsupervised Industrial Anomaly Detection

FD
Feature Distribution

FUIAD
Fully Unsupervised Industrial Anomaly Detection

PDN

Patch Description Network

AUROC

Area Under the Receiver Operating Characteristic Curve

AU-PRO

Area Under the Per-Region Overlap Curve

AD

Anomaly Detection

MSE

Mean Squared Error

AE

Autoencoder

CNN

Convolutional Neural Network

Chapter 1

Introduction

The manufacturing industry and quality control processes face a significant challenge in detecting defect anomalies and distinguishing them from normal, high-quality products. In many production environments, defects are rare and varied, making it difficult to gather enough examples to train traditional supervised models effectively. For example, in an automotive assembly line, most parts are produced correctly while only a few might have imperfections due to minor misalignments or material inconsistencies. This rarity creates an imbalance that forces the problem to be addressed in a semi-supervised or unsupervised manner.

At the heart of this challenge lies the need to define what a normal image looks like. By understanding the complete range of acceptable variations in defect-free samples, models can later detect even subtle deviations that may indicate anomalies. A simple analogy is learning to recognize a clear, unblemished apple versus one with small spots or dents. Once a robust baseline of normality is established, the detection system must be sensitive enough to flag anomalies without raising false alarms. This balance is critical because an overly sensitive system might mark harmless variations as defects while an insensitive one could miss genuine issues.

Traditionally, anomaly detection relied on statistical methods and rule-based systems such as Principal Component Analysis, k-means clustering, or Gaussian Mixture Models. These methods depended heavily on manually engineered features and clear definitions provided by domain experts. For example, in a textile factory, experts might manually specify that a certain weave pattern is normal while any deviation from that pattern should be flagged. Although these methods can work well for simpler problems, they often struggle with the complex and high-dimensional data encountered in modern manufacturing.

In recent years, machine learning-based techniques have revolutionized anomaly detection. Instead of relying on hand-crafted rules, these approaches learn directly from data. Supervised methods like Support Vector Machines and Decision Trees can classify defects when there is plenty of labeled data. However, in many industrial

applications, new or rare defects might not have been previously observed, which limits the effectiveness of supervised learning.

Unsupervised methods such as Isolation Forests or Local Outlier Factor address this by looking for unusual patterns without the need for labeled data. A practical example is an automated inspection system that learns the usual texture and color distribution of a metal surface and then highlights any spot that deviates from this learned normality. Deep learning approaches including convolutional neural networks and autoencoders further improve this process by automatically extracting hierarchical features from raw images. Autoencoders, for example, are trained to reconstruct normal images and when the reconstruction error is high it indicates a potential defect. Techniques based on generative adversarial networks have also been introduced, where synthetic images are generated to enhance the understanding of normal patterns, making it easier to spot anomalies.

The integration of advanced backbones such as ResNet101 can further enhance feature extraction and representation learning in anomaly detection studies. ResNet101 is known for its deep residual learning architecture that allows the network to capture more complex features. Its ability to learn fine-grained details and hierarchical representations makes it well-suited for identifying subtle variations in defect patterns that might be overlooked by shallower networks. This enhancement in feature representation can lead to improved classification accuracy and more robust detection of anomalies in diverse manufacturing conditions.

The primary objective of this project is to improve both classification and segmentation accuracy. Classification involves distinguishing between defective and non-defective products while segmentation pinpoints the exact location of anomalies within an image. Achieving both high classification accuracy and precise segmentation is crucial in manufacturing where identifying and isolating defects at an early stage can prevent costly production errors. Many traditional methods excel at classification but fail to provide detailed insights into the location of defects. Our approach seeks to bridge this gap by integrating advanced deep learning techniques that allow for both precise identification and accurate defect localization.

In addition, knowledge distillation plays a critical role in achieving a lightweight and fast network. Through knowledge distillation, a smaller network known as the student network is trained to replicate the behavior of a larger, more complex model known as the teacher network. This process transfers the knowledge learned by the teacher to the student, resulting in a model that is both efficient and quick in inference. Lighter networks are especially important in factory settings where real-time predictions are crucial for maintaining production speed and preventing delays in quality control processes.

Another key objective is to ensure the flexibility and adaptability of the model. In real-world manufacturing environments, production conditions change over time, new defect types emerge, and variations in materials and processes can affect the

appearance of products. A robust anomaly detection system must be capable of adapting to these changes without requiring extensive retraining. Transfer learning allows a model developed for one production line to be fine-tuned for another while continual learning methods help the model adapt to new data over time. Domain adaptation techniques further ensure that a model trained in one setting can be deployed in another without significant loss of accuracy.

Practical constraints such as computational efficiency, inference speed, and ease of integration with existing systems are also paramount. Advanced techniques like model compression, knowledge distillation, and hardware acceleration using GPUs or TPUs are applied to ensure that these complex models can run in real-time on production lines. This is especially important in environments where decisions need to be made within milliseconds to avoid halting the entire manufacturing process.

Hybrid approaches that combine traditional statistical methods with modern machine learning techniques are increasingly popular. By integrating the interpretability and simplicity of conventional methods with the adaptability and power of deep learning, these hybrid systems offer a balanced solution. Ensemble methods further enhance detection accuracy by aggregating the strengths of multiple models, ensuring robust performance even in dynamic production environments.

In summary, AI-driven anomaly detection in manufacturing is about building a strong, adaptable model of normality that can detect deviations with high accuracy, efficiency, and resilience. By leveraging a combination of traditional and modern techniques, these systems can effectively identify and isolate defects, ensuring high quality and consistent production standards in real-world industrial applications. This project aims to push the boundaries of existing anomaly detection frameworks by enhancing classification accuracy, improving segmentation capabilities, and ensuring that the model remains flexible and adaptable to the ever-changing conditions of the manufacturing industry.

Chapter 2

Background

2.1 Machine learning and Deep learning

Machine learning (ML) is a subset of artificial intelligence (AI) that enables computers to learn from data and make decisions without being explicitly programmed. It is widely used in applications such as image recognition, natural language processing, fraud detection, and recommendation systems. ML models identify patterns in data and make predictions or decisions based on those patterns.

The history of machine learning dates back to the mid-20th century, with Alan Turing's foundational work on machine intelligence. Turing proposed the Turing Test to assess a machine's ability to exhibit intelligent behavior. In 1959, Arthur Samuel coined the term 'Machine Learning' while developing self-learning algorithms for playing checkers. The 1960s and 1970s saw the development of early neural networks, particularly Frank Rosenblatt's perceptron, which introduced the concept of supervised learning.

By the 1980s, statistical learning theory led to the development of decision trees and support vector machines, which provided structured approaches to pattern recognition. The 1990s marked the rise of ensemble methods such as boosting and bagging, which improved predictive model performance by combining multiple weak learners. The introduction of kernel methods further expanded ML applications, particularly in high-dimensional data analysis [4].

The early 2000s saw a major shift with big data and increased computational power, leading to a resurgence of neural networks. Deep learning breakthroughs, particularly Convolutional Neural Networks (CNNs) [5] for image recognition and Recurrent Neural Networks (RNNs) for sequential data processing, enabled AI-driven applications in multiple domains. More recently, transformers like BERT [6] and GPT [7] have revolutionized natural language processing, enabling machines to comprehend and generate human-like text. The advancement of ML continues,

fueled by cloud computing, GPUs, TPUs, and other specialized hardware.

ML has given rise to specialized domains, including computer vision, which allows machines to interpret visual data. CNNs have driven progress in self-driving cars, medical imaging, and industrial automation. Another critical area is natural language processing (NLP), which enables machines to comprehend, generate, and interact using human language. Models like BERT and GPT have significantly improved applications such as chatbots, sentiment analysis, and automated translation.

Additionally, reinforcement learning (RL) has advanced decision-making tasks, enabling agents to learn optimal strategies through rewards and penalties. RL is widely used in game AI, robotics, and automated trading systems. Speech processing, another ML subfield, has improved voice recognition, speech synthesis, and virtual assistants like Siri and Alexa, enhancing AI-driven communication systems.

Machine learning models improve over time by analyzing large datasets, making them highly adaptable. Unlike traditional rule-based programming, where explicit instructions dictate behavior, ML algorithms develop their own logic, ensuring robustness in real-world applications. Supervised learning techniques such as decision trees and support vector machines are applied in tasks like medical diagnosis, while unsupervised learning methods like clustering aid in customer segmentation and anomaly detection.

Big data and computational advancements have led to more complex ML applications. The ability to process vast structured and unstructured datasets has spurred innovations in healthcare, where ML is used for disease diagnosis, patient risk assessment, and drug discovery. Robotics, autonomous systems, and real-time decision-making applications further benefit from reinforcement learning, enhancing adaptability and performance [4].

Deep learning, a subfield of ML, has revolutionized AI by employing multi-layered neural networks to extract complex data patterns. CNNs have transformed computer vision tasks such as image classification and object detection, while RNNs and Long Short-Term Memory (LSTM) [8] networks have greatly improved speech recognition and machine translation.

ML is now an essential part of AI-driven technologies, powering applications such as speech recognition, autonomous vehicles, predictive analytics, and cybersecurity. Fraud detection in banking, for example, relies on anomaly detection algorithms to flag suspicious transactions. The field continues to evolve with new optimization techniques, interdisciplinary research, and real-time decision-making applications, making ML one of the most dynamic areas in computer science today.

One of the key driving factors in ML evolution is feature extraction. Early methods relied on handcrafted features, such as edge detection for images and frequency-based attributes for speech recognition. The advent of deep learning has

enabled automatic feature extraction, reducing the need for manual engineering. Techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and wavelet transforms were initially used to preserve essential data structures while reducing dimensionality. However, their domain-specific constraints often required human expertise.

With deep learning, feature extractors like CNNs and Autoencoders have transformed data processing. CNNs learn hierarchical representations from raw pixel data, eliminating the need for manually designed filters. Autoencoders aid in anomaly detection and data compression by efficiently encoding high-dimensional inputs. More recently, Vision Transformers (ViTs) and transformer-based NLP models such as BERT have further enhanced feature extraction capabilities using attention mechanisms.

These advancements have improved performance across biomedical imaging, autonomous systems, and predictive analytics. The ability to extract robust features with minimal human intervention has increased accuracy and broadened ML applications. Future research is expected to refine feature extraction methodologies, making AI systems even more powerful and adaptable.

Historically, ML development has been shaped by statistical learning, probabilistic models, and neural network research. The integration of ML with cloud computing and large-scale data frameworks has further enhanced its impact. Parallel computing and distributed processing have enabled complex model training, making real-time ML applications more feasible. The proliferation of open-source libraries, such as Scikit-Learn, TensorFlow, and PyTorch, has democratized ML technology, accelerating innovation across multiple disciplines.

Today, ML drives advancements in fields as diverse as robotics, personalized marketing, and cybersecurity. Its ability to uncover hidden data patterns has made it indispensable in scientific research, financial forecasting, and industrial automation. The increasing adoption of automated ML (AutoML) has simplified model development, allowing non-experts to leverage powerful ML techniques for their applications. As computing capabilities continue to grow, the future of ML promises even more sophisticated applications, improved efficiency, and broader societal impact. Ethical considerations and explainability research are shaping responsible AI development to align with human values and regulatory requirements.

2.2 Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables machines to interpret and process visual data from the world, much like humans do. It involves developing algorithms and models that can analyze, understand, and

extract meaningful information from images and videos. One of the example of computer vision can be ability to classify the objects such as distinguish between like the sample you can see in 2.1 cat and dog or with tree. Hence, The concept of computer vision dates back to the 1960s when researchers first began exploring ways for machines to interpret visual information. Early experiments involved simple edge detection and pattern recognition techniques. Over the years, advancements in computing power and mathematical models have driven the rapid evolution of computer vision. Today, it has applications across numerous industries, including healthcare, automotive, surveillance, agriculture, and entertainment.

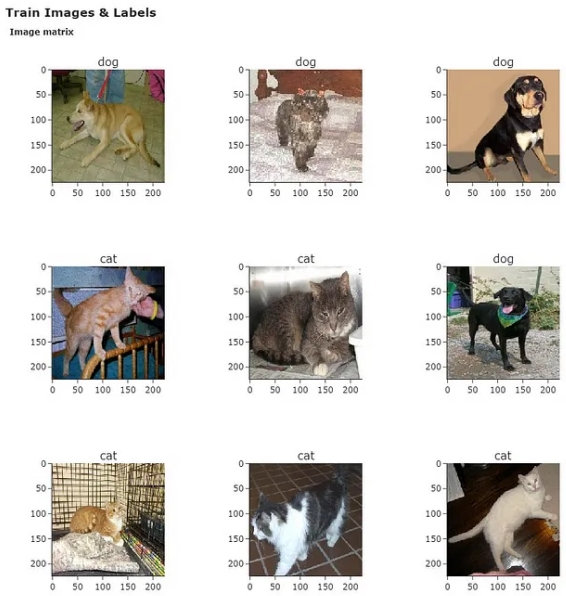


Figure 2.1: Image Classification Samples.

The foundation of computer vision lies in image processing, pattern recognition, and machine learning. Early computer vision systems relied on handcrafted features and rule-based approaches to identify objects and detect patterns in images. In the 1980s and 1990s, advancements in statistical modeling and feature extraction techniques such as Scale-Invariant Feature Transform (SIFT) [9] and Histogram of Oriented Gradients (HOG) [10] significantly improved object recognition. However, with the advent of deep learning in the 2010s, modern computer vision systems have achieved unprecedented accuracy and efficiency. Convolutional neural networks (CNNs) have become the backbone of many computer vision tasks, allowing machines to automatically learn relevant features from large datasets without manual intervention. This shift from handcrafted features to learned representations has

enabled breakthroughs in tasks like facial recognition, medical image analysis, and autonomous navigation.

One of the most common applications of computer vision is image classification, where an algorithm assigns a label to an image based on its content. Object detection extends this concept by not only recognizing objects within an image but also localizing them with bounding boxes. Facial recognition, another widely used application, identifies individuals based on facial features, with uses ranging from security systems to smartphone authentication.

Beyond static images, computer vision is also used in video analysis. Action recognition and event detection enable systems to identify human activities, monitor surveillance footage, and analyze sports events in real time. Self-driving cars rely heavily on computer vision to perceive their surroundings, detect obstacles, recognize traffic signs, and make informed driving decisions. These autonomous systems depend on advanced image processing techniques such as semantic segmentation and sensor fusion, which combine data from cameras, LiDAR, and radar to create a comprehensive understanding of the environment. Medical imaging is another critical area where computer vision assists in diagnosing diseases through analysis of X-rays, MRIs, and CT scans. Deep learning-based methods, such as convolutional neural networks (CNNs) and vision transformers, have greatly improved accuracy in detecting abnormalities, aiding radiologists in faster and more reliable diagnoses. In particular, automated anomaly detection has led to early identification of conditions such as tumors and cardiovascular diseases, significantly improving patient outcomes, which also plays a fundamental role in enhancing the quality and efficiency of computer vision systems. Techniques such as image denoising, contrast enhancement, and edge detection help preprocess raw data to improve recognition accuracy. Additionally, advanced backbones, such as ResNet[11], EfficientNet[12], and Swin Transformer[13], provide powerful feature extraction capabilities, enabling deep learning models to process high-resolution images efficiently.

In recent years, self-supervised learning and generative models have revolutionized computer vision by reducing dependence on labeled data. Methods like contrastive learning and Vision Transformers (ViTs) have improved generalization, making models more robust and adaptable to real-world conditions. These advancements, along with continual improvements in hardware acceleration and edge computing, are driving the next wave of innovation in industrial and consumer applications of computer vision. Despite its advancements, computer vision faces several challenges. Variability in lighting conditions, occlusions, and viewpoint changes can affect model performance. The future of computer vision is promising, with ongoing research exploring areas such as 3D vision, zero-shot learning, and self-supervised learning. One particularly impactful area is anomaly detection, where computer vision systems are being developed to identify rare and unusual patterns in data that may indicate defects, fraud, or health concerns. Industrial

applications of anomaly detection include quality control in manufacturing, where computer vision systems analyze products on assembly lines to detect defects in real time, reducing waste and ensuring high standards of production.

As hardware continues to improve and datasets grow, computer vision will play an increasingly crucial role in shaping the next generation of intelligent systems and automation. The continued development of AI-driven anomaly detection methods will further enhance reliability in various industries, from financial fraud detection to predictive maintenance in industrial equipment, ensuring more efficient and proactive decision-making.

A significant milestone in the advancement of computer vision was the creation of ImageNet [14], a large-scale visual database designed for use in image recognition research. Introduced by Fei-Fei Li and her team in 2009, ImageNet provided researchers with a massive, well-labeled dataset that fueled groundbreaking progress in deep learning-based image classification. in 2.2 you can see some samples from this dataset. The annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) became a benchmark for evaluating computer vision models, inspiring the development of increasingly sophisticated neural networks.The breakthrough moment came in 2012 when AlexNet [15], a deep convolutional neural network (CNN), significantly outperformed traditional machine learning approaches in the ImageNet competition. This achievement demonstrated the power of deep learning and accelerated research in neural network architectures, leading to the creation of models such as VGG, ResNet, and EfficientNet, which have since pushed the boundaries of computer vision applications. The impact of ImageNet extends far beyond academic research. By providing a standardized dataset and challenge, ImageNet not only revolutionized image recognition but also laid the foundation for modern AI-driven anomaly detection methods, enabling more robust and accurate identification of irregular patterns across diverse industries. As computer vision continues to evolve, the legacy of ImageNet remains a cornerstone in the pursuit of intelligent, automated systems.

Figure 2.2: ImageNet sub-dataset samples.

2.3 Supervised Learning

Artificial Intelligence (AI) has transformed industries by enabling machines to learn from data and make intelligent decisions. One of the most commonly used techniques in AI is supervised learning, a type of machine learning where models are trained on labeled datasets. This method allows systems to map inputs to outputs with high accuracy, making it essential for applications such as image recognition, natural language processing, fraud detection, and medical diagnosis. Supervised learning is a machine learning paradigm that involves training a model using labeled data. The input consists of features, also called independent variables, while the output consists of labels, also known as dependent variables. The model learns a function that maps inputs to outputs based on patterns in the training data. This technique is widely used in classification and regression tasks, making it one of the fundamental methodologies in AI development.

The workflow of supervised learning involves several steps. The first step is data collection, where a dataset consisting of input-output pairs is gathered. Next, data preprocessing is performed, which includes cleaning the data, handling missing values, and normalizing features. Once the data is prepared, it is split into training and testing subsets, often in an 80-20 ratio, to evaluate model performance. After data splitting, an appropriate algorithm is chosen based on the nature of the problem. The selected model is then trained using the training data to learn patterns and relationships. The trained model is evaluated using performance metrics such as accuracy, precision, recall, or root mean squared error. To enhance performance, hyperparameter tuning is carried out to optimize model parameters. Once the model achieves satisfactory accuracy, it is deployed for real-world applications.

Supervised learning can be categorized into two primary tasks: classification and regression. In classification, the model predicts discrete labels or categories, such as determining whether an email is spam or not. Examples of classification algorithms include logistic regression, decision trees, random forests, support vector machines, and neural networks. In regression, the model predicts continuous values, such as estimating house prices based on features like square footage and location. Common regression algorithms include linear regression, polynomial regression, ridge regression, and neural networks.

Several supervised learning algorithms play a critical role in various applications. Linear regression is a simple yet effective algorithm used for predicting numerical values based on a linear relationship between input variables. Logistic regression is employed for binary classification problems, predicting probabilities that map inputs to discrete labels. Decision trees are versatile algorithms that split data into branches based on feature values, making them useful for both classification and regression tasks. Random forests enhance decision trees by combining multiple trees to improve accuracy and reduce overfitting. Support vector machines classify data by

finding an optimal hyperplane that separates different categories. Neural networks, particularly deep learning models, can learn complex patterns and relationships in data, making them powerful for tasks like image and speech recognition.

The success of supervised learning depends on the quality and quantity of labeled data. A well-labeled dataset provides the model with clear examples to learn from, leading to better generalization. However, obtaining high-quality labeled data can be expensive and time-consuming. Furthermore, techniques like cross-validation ensure that models are trained and tested effectively to avoid overfitting and underfitting.

Despite its advantages, supervised learning has some limitations. It requires large labeled datasets, which may not always be available or feasible to obtain. Models trained on biased or imbalanced datasets may produce skewed results, affecting their reliability in real-world applications. Moreover, supervised learning models may struggle with generalizing to unseen data if they are overly complex or trained on insufficiently diverse datasets. Overfitting, where a model learns noise instead of actual patterns, is another challenge that needs to be addressed using techniques like regularization and dropout.

Supervised learning is widely used in various real-world applications. In healthcare, it assists in diagnosing diseases by analyzing medical images and patient records. In finance, it helps detect fraudulent transactions by identifying unusual spending patterns. In e-commerce, recommendation systems use supervised learning to suggest products based on user preferences. Autonomous vehicles rely on supervised learning for object detection and decision-making, ensuring safer navigation. Natural language processing applications, such as sentiment analysis and chatbot development, benefit from supervised learning models trained on text data.

Future trends in supervised learning focus on improving model efficiency and reducing data dependency. Advances in semi-supervised learning aim to leverage both labeled and unlabeled data to reduce the need for extensive manual labeling. Transfer learning enables models to apply knowledge learned from one domain to another, reducing training time and data requirements. Federated learning enhances privacy by allowing models to be trained across decentralized devices without sharing raw data. Additionally, explainability and interpretability of supervised learning models are gaining importance to ensure ethical and transparent AI decision-making.

Supervised learning remains a cornerstone of artificial intelligence and machine learning. By leveraging labeled data, it enables models to achieve high accuracy in various applications, from healthcare to finance and autonomous systems. While challenges such as data dependency and overfitting persist, advancements in AI continue to refine and expand the capabilities of supervised learning. As research progresses, new methodologies will further enhance the efficiency, interpretability, and ethical considerations of supervised learning models, shaping the future of

AI-driven technologies.

we can consider that supervised learning is one of the most widely used approaches in computer vision for defect detection in industrial manufacturing, quality control, and medical imaging. Unsupervised approaches enables models to learn precise defect detection patterns and achieve high accuracy in automated inspections.

2.4 Unsupervised learning

Unsupervised learning has emerged as a pivotal approach in computer vision, particularly for defect detection in industrial settings, manufacturing, and quality control. In contrast to supervised methods that require vast amounts of annotated data, unsupervised techniques learn intrinsic representations directly from unlabeled data. This approach exploits the inherent structure present in the data, such as clusters, manifolds, or latent representations, without relying on explicit input-output mappings. The central idea is to model the underlying probability distribution $p(x)$ of the data and to learn an approximation $\hat{p}(x)$. When a new input x' deviates significantly from $\hat{p}(x)$, it is flagged as an anomaly an especially useful property in defect detection, where normal samples are abundant and defects are both rare and diverse.

A fundamental goal in unsupervised learning is to approximate the true distribution $p(x)$ of data samples x. With a robust model for $\hat{p}(x)$, any input x' residing in a low-density region can be considered anomalous. This notion is critical for applications in quality control where subtle deviations from normality may indicate defects.

One widely adopted technique in this domain is the autoencoder [16]. An autoencoder is composed of two main components: an encoder $f : \mathbb{R}^n \to \mathbb{R}^m$ that compresses the input x into a latent representation z, and a decoder $g : \mathbb{R}^m \to \mathbb{R}^n$ that reconstructs the original input from z. The training objective is to minimize the reconstruction loss:

$$\min_{f,g} \ \mathbb{E}_{x\sim p(x)} \left[\|x - g(f(x))\|^2 \right]. \tag{2.1}$$

When trained on defect-free images, the autoencoder learns to reconstruct these images with low error. However, if a defective image is presented, the reconstruction error become:

$$\|x' - g(f(x'))\|$$

Despite the promising results, challenges remain. High false positive rates may arise due to minor variations in texture, lighting, or manufacturing tolerances. Moreover, deep unsupervised models often lack transparency, which can hinder trust in safety-critical applications. The computational complexity required to train models such as GANs and autoencoders necessitates access to large datasets and significant computational resources. Additionally, the representativeness of the training data is crucial; incomplete data can lead to poor generalization in real-world scenarios. Hybrid approaches that combine unsupervised methods with semi-supervised or human-in-the-loop techniques are emerging as promising solutions to these challenges. Future research is also exploring transformer-based

models and diffusion models to improve both the quality and interpretability of unsupervised learning.

Consequently, the technical foundations of unsupervised learning ranging from density estimation and latent space modeling to clustering and contrastive representation learning provide a robust framework for defect detection in computer vision. By focusing on the intrinsic structure of defect-free data, these methods offer scalable and adaptable solutions for detecting anomalies in a variety of industrial and medical applications.

2.5 Metrics

2.5.1 Accuracy

Accuracy is one of the most commonly used evaluation metrics in computer vision and machine learning models. It measures the proportion of correctly classified instances in a dataset, including both true positives (TP) and true negatives (TN). Mathematically, it is defined as:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{True Positives (TP)} + \text{False Positives (FP)} + \text{True Negatives (TN)} + \text{False Negatives (FN)}} \quad (2.2)$$

In simple terms, accuracy represents the ratio of correctly predicted samples to the total number of samples in the dataset. It provides an intuitive measure of model performance and is widely adopted in applications such as image classification, object recognition, and segmentation.

Despite its usefulness in balanced datasets, accuracy has significant limitations when applied to real-world problems where class distributions are imbalanced or when different types of classification errors have varying degrees of importance. One major limitation arises in datasets with severe class imbalance. If one class is significantly more frequent than another, accuracy can be misleading. For instance, consider a binary classification problem where 95% of instances belong to Class A and only 5% belong to Class B. A naive model that always predicts Class A would achieve 95% accuracy while completely failing to identify Class B instances. Even though the accuracy is high, the model lacks the ability to detect minority class samples, which can be problematic in high-stakes applications such as medical diagnosis, fraud detection, and industrial defect detection.

Accuracy is particularly ineffective in anomaly detection problems such as fraud detection, cybersecurity threat identification, and network intrusion detection. These tasks often involve datasets where anomalies are extremely rare compared to normal instances. A fraud detection system, for example, might have only 0.5% fraudulent transactions in a dataset. If a model simply classifies all transactions as non-fraudulent, it would achieve 99.5% accuracy but fail to detect any actual fraud cases. In such scenarios, accuracy does not provide meaningful insight into model performance, and alternative metrics must be considered to evaluate how well anomalies are detected.

Another domain where accuracy fails to provide a reliable evaluation metric is medical imaging and disease detection. Consider an automated system designed to detect cancer in mammograms. If cancer-positive cases account for only 1% of the dataset, a model that classifies all cases as "healthy" would yield 99% accuracy while completely failing to diagnose cancer-positive patients. In medical applications, the cost of false negatives (misdiagnosing a diseased patient as healthy) is significantly

higher than the cost of false positives (incorrectly diagnosing a healthy patient as diseased). Since accuracy does not differentiate between these errors, it is an inadequate measure of performance in such scenarios.

In industrial applications, particularly in automated quality control and defect detection on manufacturing lines, accuracy can be highly misleading. Manufacturing datasets are typically heavily imbalanced, as most products pass quality inspection, and only a small fraction contain defects. Suppose a factory produces one million units per day, with only 500 defective units. A model that predicts "no defect" for every product would achieve 99.95% accuracy while failing to identify any defective units. In this case, accuracy provides a false sense of security, as the model does not serve its primary purpose of detecting faulty products. Failing to catch defective units can result in significant financial losses, product recalls, and even safety hazards. For industrial applications, recall is a far more critical metric, as it ensures that the model captures defective units even at the cost of some false positives, which can be manually verified through secondary inspections. In object detection-based defect identification models, Intersection over Union (IoU) is often a more meaningful measure of performance.

Although accuracy is a simple and widely used metric, it should not be the sole evaluation criterion for model performance, particularly in applications where class imbalances exist or where different types of misclassification errors have drastically different consequences. Relying solely on accuracy can lead to misleading conclusions about a model's true effectiveness. In cases such as defect detection, medical diagnosis, fraud detection, and autonomous systems, optimizing for recall or F1 score is often more critical than maximizing accuracy alone. Therefore, accuracy should always be analyzed in conjunction with other performance metrics to ensure that machine learning models perform reliably in practical, real-world applications.

2.5.2 F-Score

The F1 score is a fundamental metric in machine learning and computer vision, particularly useful for evaluating classification models in imbalanced datasets or scenarios where false positives and false negatives carry different consequences. Unlike accuracy, which may be misleading when class distributions are uneven, the F1 score balances precision and recall, making it a more reliable performance measure in such cases.

Mathematically, the F1 score is defined as the harmonic mean of Precision and Recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

where:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

Precision represents the proportion of correctly predicted positive instances out of all predicted positives, while Recall quantifies the proportion of actual positives that were correctly classified. The F1 score is particularly useful in defect detection in manufacturing lines, where the occurrence of defects is rare compared to non-defective products, leading to an inherently imbalanced dataset.

In automated manufacturing quality control, computer vision models are widely employed to identify defective products. Since defective products constitute only a small percentage of total production, relying solely on accuracy can be misleading. A naive classifier that labels every product as non-defective may achieve an accuracy of 99.9%, despite failing to detect any actual defects. The F1 score mitigates this issue by considering both false positives (FP) and false negatives (FN), ensuring that misclassified defects impact the evaluation proportionally.

False positives and false negatives in manufacturing defect detection systems have distinct operational implications:

- False Positives (FP): Occur when a non-defective product is incorrectly classified as defective. This leads to unnecessary manual inspections, increased rework, and production inefficiencies.
- False Negatives (FN): Occur when a defective product is mistakenly classified as non-defective. This is far more critical, as defective units may pass through quality control, resulting in customer complaints, recalls, financial losses, and potential safety hazards.

Because false negatives typically pose greater risks than false positives, manufacturing lines often prioritize recall (ensuring defective products are caught).

However, focusing only on recall can lead to an excessive number of false positives, unnecessarily rejecting good products, leading to increased inspection costs and lower production efficiency. The F1 score helps balance these trade-offs, ensuring that both precision and recall contribute equally to the model evaluation.

To optimize the F1 score in defect detection systems, manufacturers employ several strategies. Threshold Tuning adjusting the classification threshold helps balance precision and recall, ensuring that the model neither under-detects nor over-detects defects. Furthermore, rather than a simple classification approach, unsupervised and semi-supervised anomaly detection methods can improve the ability to detect rare defect cases. In addition, by combining the F1 score with other metrics such as the Matthews Correlation Coefficient (MCC) and Area Under the Precision-Recall Curve (AUC-PR) provides a more holistic assessment of model performance.

Although the accuracy appears extremely high, the low F1 score indicates that the model struggles with precision—it detects most defective products (high recall) but misclassifies a large number of good products as defective (low precision). Increasing precision while maintaining recall would improve the F1 score, making the system more reliable for defect detection.

Ultimately, the F1 score plays a critical role in manufacturing defect detection, offering a robust evaluation metric that accounts for both false positives and false negatives. Unlike accuracy, which can be misleading in class-imbalanced scenarios, the F1 score provides a balanced assessment, ensuring that quality control processes effectively identify defects while minimizing unnecessary rework and costs. By leveraging techniques such as threshold tuning, anomaly detection, and multi-metric analysis, manufacturers can optimize defect detection systems, leading to higher product quality, reduced operational waste, and improved customer satisfaction.

2.5.3 AUROC

The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC-ROC) metric are fundamental tools for evaluating machine learning models used in defect detection for manufacturing lines. These metrics provide insight into how well a model distinguishes between defective and non-defective items across different classification thresholds. Unlike accuracy, which can be misleading in cases of class imbalance, AUC-ROC offers a more threshold-independent assessment of model performance.

The ROC curve is a graphical representation of a model's behavior at varying thresholds. It is plotted by evaluating the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold levels. The True Positive Rate (TPR), also known as recall, measures the proportion of actual defective items correctly identified:

$$TPR = \frac{TP}{TP + FN} \tag{2.5}$$

Similarly, the False Positive Rate (FPR) quantifies the proportion of non-defective items that are mistakenly classified as defective:

$$FPR = \frac{FP}{FP + TN} \tag{2.6}$$

A highly effective model produces an ROC curve that bends sharply toward the upper-left corner, where TPR is maximized while FPR remains minimal. The ideal model would achieve a point at (0,1) on the ROC plot, meaning it correctly detects all defective items without falsely classifying any non-defective ones. However, real-world models rarely achieve perfect classification, making AUC-ROC a valuable comparative metric.

The Area Under the ROC Curve (AUC-ROC) quantifies a model's ability to rank defective items higher than non-defective ones. Mathematically, AUC represents the probability that a randomly selected defective item will be assigned a higher defect score than a randomly chosen non-defective item. A perfect classifier has AUC = 1.0, meaning it flawlessly distinguishes between the two classes, while an AUC of 0.5 indicates performance equivalent to random guessing.

AUC-ROC is particularly important for defect detection in manufacturing, where datasets are often highly imbalanced. Since defects typically constitute a small fraction of total production, accuracy can be misleading. A model that simply predicts "non-defective" for every item may achieve high accuracy but fail to detect actual defects. AUC-ROC avoids this issue by evaluating ranking ability rather than absolute classification performance, ensuring the model is assessed across various operating conditions.

In real-world defect detection, threshold selection plays a critical role, as manufacturers must decide between minimizing false positives (FPR) and false negatives (FNR) and the implications of these errors vary significantly

Depending on business priorities, different thresholds along the ROC curve may be optimal:

1. If false positives are costly but false negatives are acceptable, a higher threshold should be used to minimize FPR, reducing unnecessary defect labeling.

2. If false negatives are critical, such as in aerospace or medical device manufacturing, a lower threshold is preferable to maximize recall (TPR), ensuring defective items are not overlooked.

3. If both errors have similar costs, an intermediate threshold balancing precision and recall may be optimal.

Figure 2.3 illustrates a typical AUC-ROC curve, where different points correspond to different trade-offs in classification performance.

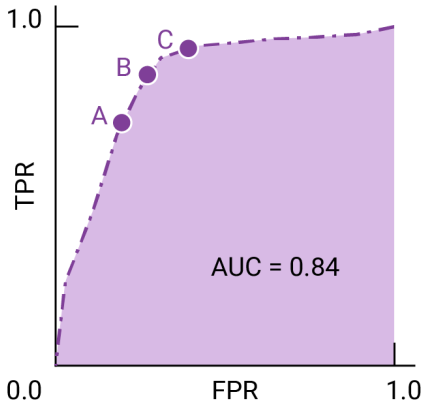


Figure 2.3: Example of an AUC-ROC curve illustrating model performance at varying thresholds.

AUC-ROC is also a valuable tool for comparing multiple defect detection models. With some examples we can see that why this metric is completely related and necessary for industrial anomaly detection task, consider two models:

- **Model A:** AUC = 0.65 (Moderate classification performance)
- **Model B:** AUC = 0.93 (Superior classification performance)

Since Model B achieves a significantly higher AUC, it will generally outperform Model A at almost all threshold settings, making it the preferred choice for deployment. Evaluating models based on AUC allows manufacturers to make data-driven decisions, selecting the model that best aligns with business requirements and quality control standards.

A real-world example further demonstrates the importance of AUC-ROC in manufacturing. Consider an automated semiconductor fabrication plant, where defective microchips constitute only 0.1% of total production. A defect classification model achieves an accuracy of 99.8%, which appears excellent at first glance. However, a deeper analysis using AUC-ROC reveals key issues:

- AUC = 0.62: The model struggles to differentiate defective chips from functional ones.
- False Negative Rate (FNR) = 30%: Nearly one-third of actual defective chips go undetected, leading to potential failures in downstream applications.
- False Positive Rate (FPR) = 5%: A significant number of good chips are incorrectly flagged as defective, increasing inspection and rework costs.

This analysis highlights that while accuracy appears high, the model's ranking ability (AUC-ROC) reveals its limitations. An improved version of the model with AUC = 0.91 demonstrates a much higher capacity to distinguish defects, reducing false negatives to 3% while keeping false positives manageable at 2%. This directly translates to better quality control, reduced waste, and cost efficiency.

Overall, AUC-ROC is a crucial evaluation metric in defect detection systems deployed in manufacturing environments. It enables more reliable model assessment in class-imbalanced settings, where traditional accuracy-based metrics fail. By analyzing ROC curves and AUC scores, manufacturers can optimize their defect detection to minimize false negatives while keeping false positives within acceptable limits. This leads to higher product reliability, lower operational costs, and improved customer satisfaction, ensuring that automated manufacturing lines operate with maximum efficiency and precision.

2.5.4 AU-PRO

The Area Under the Per-Region-Overlap Curve (AU-PRO) is a crucial evaluation metric in anomaly segmentation, especially in applications where precise localization of defects or anomalies is essential. Traditional evaluation metrics, such as accuracy, precision, recall, and F1-score, primarily assess detection capability but fail to account for the quality of segmentation. In contrast, AU-PRO is designed to quantify how well a model delineates anomaly regions, making it a superior metric for tasks requiring precise boundary segmentation.

The foundation of AU-PRO lies in the concept of Per-Region Overlap (PRO), which assesses how well a predicted anomaly region aligns with the corresponding ground-truth region. This is calculated using the Intersection over Union (IoU):

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{2.7}$$

where P represents the predicted anomaly mask, and G represents the ground-truth anomaly mask. Unlike traditional pixel-wise accuracy measures that may disproportionately weigh large anomalies, AU-PRO ensures that both small and large anomalies contribute proportionally to the overall score. This makes it robust in scenarios where minor defects hold critical importance, such as in industrial quality control and medical imaging.

AU-PRO evaluates segmentation performance across different confidence levels rather than at a single decision threshold. Most anomaly segmentation models generate continuous-valued anomaly scores, rather than binary classifications. To create a fair evaluation, the predicted anomaly map is binarized at multiple thresholds. For each threshold, the per-region overlap is computed, and the results are aggregated to form the AU-PRO curve, where the x-axis represents the threshold values, and the y-axis represents the average per-region overlap value.

The AU-PRO score is determined by integrating the area under this curve:

$$\text{AU-PRO} = \int_{t_{\min}}^{t_{\max}} \text{PRO}(t)\, dt \tag{2.8}$$

A higher AU-PRO score indicates better segmentation accuracy and signifies that the model can consistently delineate anomaly boundaries across varying confidence levels. Unlike single-threshold metrics, AU-PRO is threshold-independent, making it particularly useful for evaluating models without requiring arbitrary threshold selection.

AU-PRO has significant implications in real-world applications that demand precise anomaly segmentation. In industrial defect detection, precise defect localization is critical for automated manufacturing and quality control. Standard pixel-wise evaluation methods often fail to detect tiny or irregularly shaped defects, leading to

unreliable assessments. With this metric provides a rigorous evaluation by ensuring that both major and minor defects are equally considered. In industries such as semiconductor manufacturing, PCB inspection, and material surface analysis, the ability to assess segmentation fidelity at the per-region level directly impacts production quality and defect classification reliability.

In medical image analysis, where the accurate segmentation of tumors, lesions, and other pathological structures is essential, AU-PRO ensures fine-grained localization accuracy. Traditional classification metrics, such as precision-recall and Dice coefficient, fail to capture how well an anomaly is segmented. This metric provides pixel-level accuracy evaluation, which is crucial in applications such as cancer detection, brain lesion segmentation, and ophthalmic disease diagnosis. Since missegmentation can lead to misdiagnosis and improper treatment planning, AU-PRO is particularly valuable in ensuring reliable and interpretable medical AI systems.

Autonomous systems, such as self-driving cars and robotic inspection systems, also benefit significantly from AU-PRO. These systems rely on accurate anomaly segmentation for obstacle detection and safety-critical decision-making. If an autonomous system fails to segment an object accurately, it may misinterpret hazardous obstacles, road defects, or foreign objects, leading to potential accidents. AU-PRO helps evaluate the segmentation quality of such models across different confidence levels, ensuring that anomaly boundaries are well-defined, leading to more reliable object detection and decision-making.

Another key area where AU-PRO plays an essential role is in multimodal anomaly detection. In many advanced applications, systems analyze data from multiple modalities, such as combining RGB images with infrared, depth maps, or 3D point clouds. Evaluating the segmentation accuracy of such complex data representations requires a metric that captures segmentation quality independently of modality-specific noise and resolution differences. AU-PRO enables fair comparisons and ensures that multimodal anomaly detection systems are evaluated holistically.

Despite its advantages, AU-PRO comes with computational challenges. Calculating per-region overlaps across multiple thresholds increases computational cost, especially for high-resolution images and large datasets. Additionally, the accuracy of AU-PRO is highly dependent on the quality of the ground-truth masks. If the annotation process introduces errors, such as inconsistent segmentation boundaries or mislabeled anomaly regions, the AU-PRO score may not accurately reflect model performance. Careful curation of ground-truth data is necessary to ensure trustworthy evaluations.

AU-PRO stands out as a powerful and sophisticated metric for evaluating anomaly segmentation models, particularly in applications that require high localization accuracy. By considering segmentation performance across multiple thresholds, AU-PRO provides a more comprehensive and robust evaluation than

traditional metrics. Its applicability in industrial inspection, medical imaging, autonomous systems, and multimodal anomaly detection demonstrates its versatility. While computationally intensive, its ability to assess fine-grained segmentation fidelity makes it an indispensable tool for researchers and engineers developing high-precision anomaly detection models.

Chapter 3

Related Work

Anomaly and defect detection in industrial manufacturing lines has witnessed substantial progress over the past decade, evolving from early computer vision techniques based on handcrafted features to contemporary deep learning frameworks that can accurately pinpoint subtle defects in complex environments. This evolution has been driven largely by the critical need for quality assurance and operational efficiency in manufacturing, where the cost of undetected defects can be extremely high. Early approaches in this field relied heavily on traditional image processing methods, which, despite their computational efficiency, often fell short when confronted with the variability and subtlety of defects in real world settings. These classical techniques were primarily based on statistical modeling and thresholding methods that attempted to capture the distribution of normal features, yet they frequently struggled to generalize across different scenarios due to variations in object alignment, lighting conditions, and environmental factors.

The advent of large scale, annotated datasets has played a pivotal role in advancing the state-of-the-art. Notably, the introduction of the MVTec Anomaly Detection (AD) dataset [17, 18] marked a significant milestone by providing a diverse set of inspection scenarios in which training data consisted exclusively of defect free images while the test sets included a variety of anomalies. This dataset, along with its successors such as the Visual Anomaly (VisA) [19] and MVTec Logical Constraints (LOCO) datasets [18], has spurred considerable research interest. These datasets encompass both structural anomalies—such as scratches, stains, or cracks—and logical anomalies where the spatial or contextual relationships between objects are disrupted. The pixel level annotations provided by these datasets have enabled researchers to benchmark both the detection and localization of anomalies with unprecedented precision. Since, usage of industrial anomaly detection became more and more, we needed more challenging datasets to evaluate the performance of our algorithms. Hence, in 2024 Real-IAD [3] introduced a new version of challenging dataset suitable to explore the performance of models in industrial environment.

Industrial anomaly detection techniques overall divided into two main approaches, supervised AD, unsupervised AD. Furthermore, you can find a good overview of all state of the arts related to this computer vision task in 3.1.

In supervised AD we have different subset of challenge. Zero-Shot and Few-Shot Anomaly Detection Recent works have focused on enabling anomaly detection with minimal or no labeled data. Learning based approaches, such as the Unsupervised Metaformer model for anomaly detection [20] and hierarchical transformation discriminating generative models [21], aim to improve feature representations for low-data regimes. Few-shot anomaly detection techniques using registration-based methods [22] has pushed the limits of industrial anomaly detection. Additionally, WinCLIP [23] introduced a zero-few-shot approach for anomaly classification and segmentation, while hybrid prompt regularization techniques [24] further improved segmentation performance without explicit training. Handling noisy data is critical for robust anomaly detection. The TrustMAE framework [25] leverages memory-augmented autoencoders to improve defect classification in the presence of noise. Other approaches include Latent Outlier Exposure [26], which enhances detection with contaminated datasets, and Deep One-Class Classification using Interpolated Gaussian Descriptors. The SoftPatch method focuses on unsupervised anomaly detection in noisy environments, while self-supervised refinement techniques [27] aim to iteratively improve anomaly detection performance. With the growing importance of 3D data, several studies have addressed anomaly detection in 3D point clouds. Deep geometric descriptors have been applied to 3D anomaly detection [28], while asymmetric teacher-student networks [29] offer robust industrial applications. Classical 3D feature based methods [30] highlight the effectiveness of traditional geometric approaches. Hybrid fusion based multimodal methods [31] have demonstrated improved detection capabilities. Additionally, networks like EasyNet [32] and datasets such as Real3D-AD [33] contribute to benchmarking and advancing 3D anomaly detection. Synthetic data augmentation plays a crucial role in improving anomaly detection performance. GAN-based techniques, such as defect image sample generation [34] and highfidelity defect synthesis [35], have been employed for surface defect detection. Simulation-based few-shot learning methods [36] and deep learning-driven synthetic data augmentation have further enhanced defect segmentation and classification.

Moreover, research on continual adaptation [37] focuses on mitigating catastrophic forgetting while maintaining anomaly detection performance. Unified models, such as the approach for multi-class anomaly detection and OmniAL [38], a CNN framework for unsupervised anomaly localization, propose holistic solutions for anomaly detection across multiple domains.

These advancements collectively contribute to improving anomaly detection across various industrial and research applications, paving the way for more efficient, scalable, and adaptive detection systems. In response to the challenges posed by

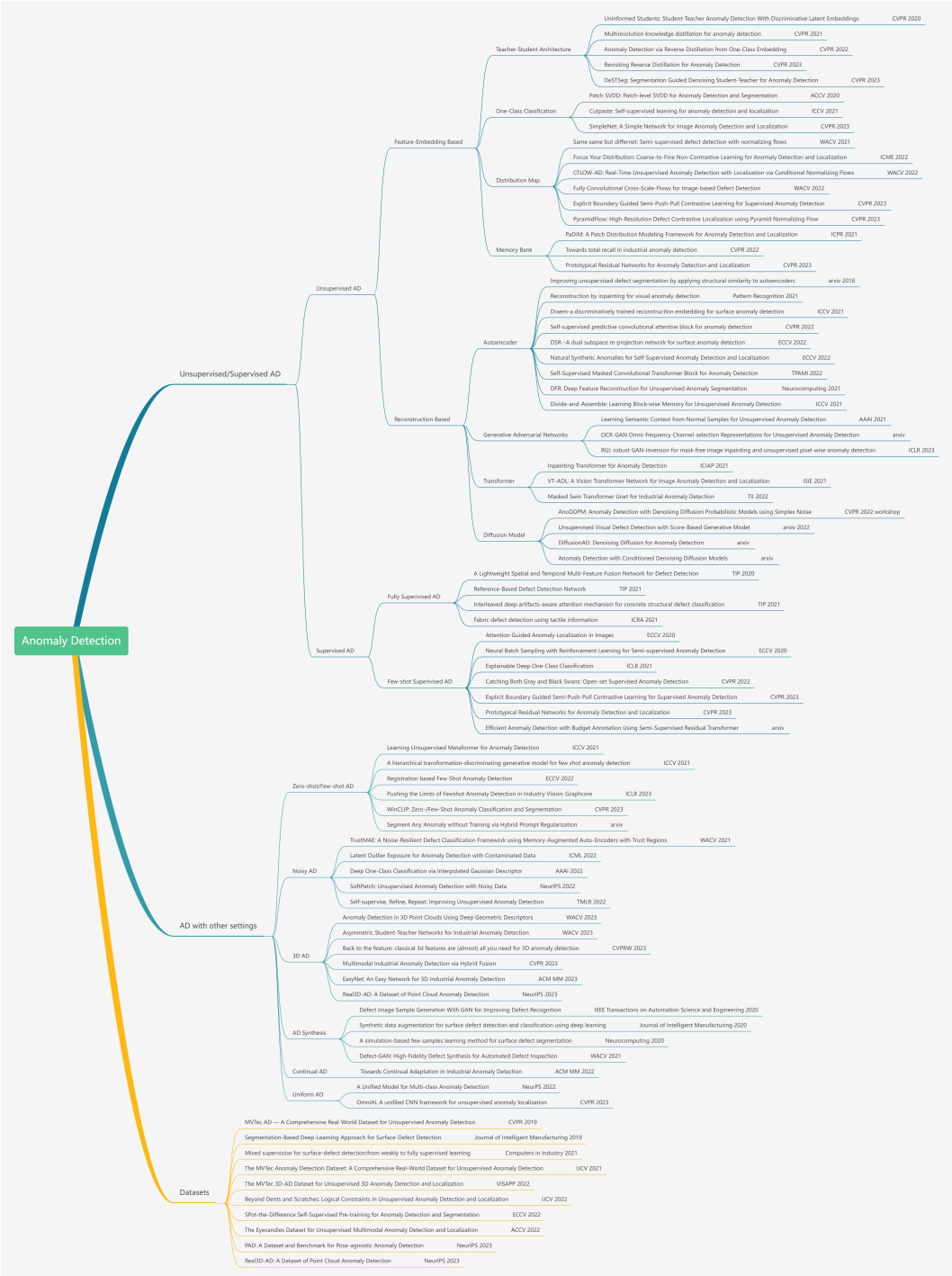


Figure 3.1: Overall overview of industrial anomaly detection strategies till 2024.

Source: `https://github.com/M-3LAB/awesome-industrial-anomaly-detection`

these datasets, deep learning techniques have become the preferred approach for anomaly detection in manufacturing. One influential line of research has focused on leveraging convolutional neural networks (CNNs) as feature extractors. By mapping input images into high-dimensional feature spaces, these methods facilitate the modeling of the distribution of normal features. Various techniques have been employed to model these distributions, including the use of multivariate Gaussian models, Gaussian Mixture Models, and normalizing flows [39, 40]. For example, methods like PatchCore [41] have demonstrated that clustering features and conducting k-nearest neighbor searches on a reduced set of representative features can lead to significant improvements in both detection accuracy and computational efficiency [41]. This approach capitalizes on the idea that anomalies can be detected as deviations from the learned normal feature distribution, thereby offering a statistically grounded framework for defect detection.

Another prominent avenue of research involves reconstruction based methods. These approaches are predicated on the concept that a model trained exclusively on normal data should be capable of accurately reconstructing normal images, while it would struggle with anomalous inputs. Autoencoders and generative adversarial networks (GANs) have been widely used for this purpose [42, 43]. The reconstruction error, which measures the difference between the input and its reconstruction, is then interpreted as an indicator of anomaly. Although this method is intuitively appealing and provides a direct visual interpretation of anomalies, it is not without its drawbacks. A common issue is that models sometimes generate blurry or imprecise reconstructions even for normal inputs, which can lead to false positive detections. To mitigate such challenges, later research has introduced additional constraints or memory modules to enforce the reconstruction of only the most representative normal features. Techniques such as AESSIM and MemAE have incorporated these ideas by either adding auxiliary constraints or utilizing a memory bank of normal features to guide the reconstruction process [44, 45]. Furthermore, some approaches have experimented with generating synthetic anomalies during training to enhance the network's ability to distinguish between normal and abnormal patterns. This dual focus on both accurate reconstruction and robust anomaly highlighting has become a defining characteristic of modern reconstruction based methods.

A third significant research direction has been the application of knowledge distillation techniques. In these methods, a large, pretrained "teacher" network provides high quality feature representations that encapsulate the complexity of normal images, while a smaller "student" network is trained to mimic these features. The fundamental assumption is that the student network, having been exposed only to normal data, will produce outputs that deviate from the teacher's when processing anomalous images. These discrepancies, which manifest as elevated reconstruction or prediction errors, can be effectively used to score and localize

anomalies [46]. Early implementations of this paradigm, such as those employed in US and MRKD, utilized multi resolution feature representations and compared outputs across different scales to detect even subtle defects [46, 47]. Subsequent innovations have sought to address the challenge of overgeneralization where the student network inadvertently learns to mimic the teacher even for anomalous inputs by incorporating mechanisms such as feature pyramid matching and reverse distillation [48, 49]. These enhancements have led to more pronounced discrepancies between the normal and anomalous feature distributions, thereby improving both the reliability and precision of anomaly localization.

More recently, there has been a trend towards integrating the strengths of the aforementioned approaches into unified frameworks. Such hybrid methods aim to model the normal data distribution more accurately while simultaneously ensuring that the deviations caused by anomalies are captured effectively. By optimizing both the normal and abnormal feature distributions collaboratively, these approaches explicitly enlarge the margin between them, reducing prediction uncertainty and enhancing localization performance [29]. Advances in network architecture, including the adoption of residual connections and attention mechanisms, have further bolstered the capacity of these systems to detect subtle defects even in the presence of challenging imaging conditions. The recent exploration of transformer based models in this context has also opened new avenues for capturing long range dependencies and contextual cues, which are essential for accurately identifying defects in complex industrial environments [50, 13].

In industrial applications, where the requirements for real time processing and high accuracy are paramount, the continuous evolution of these methods is essential. The integration of distribution based, reconstruction based, and knowledge distillation–based techniques not only addresses the inherent challenges posed by limited anomalous training data but also enhances the overall robustness of the detection systems. Each approach offers unique advantages: distribution based methods provide a strong statistical foundation, reconstruction based methods offer intuitive visual cues through reconstruction errors, and knowledge distillation approaches leverage the power of pretrained networks to implicitly capture the nuances of normal feature distributions. Despite the progress, each method also comes with its own set of challenges, such as high computational demands, risk of overgeneralization, or the difficulty of balancing reconstruction fidelity with anomaly sensitivity.

Collectively, the body of work in this field reflects a rich tapestry of ideas that have progressively pushed the boundaries of what is achievable in automated defect detection. As researchers continue to refine these techniques and develop more sophisticated models, the integration of these various paradigms is expected to lead to systems that are not only more accurate and robust but also more adaptable to the dynamic and often unpredictable conditions of real world industrial environments.

The pursuit of improved methods for anomaly and defect detection remains a vibrant and critical area of research, promising to deliver significant benefits in terms of quality control and operational efficiency in manufacturing processes [46, 41, 49].

Chapter 4

Methods

Obtaining labeled abnormal data is challenging or even infeasible. Industrial environments often present complexities that make the prediction of abnormal images extremely difficult, particularly when such images are rare or highly variable. In addition, industrial images may exhibit significant noise and environmental variability factors that can degrade the performance of conventional detection methods. To overcome these obstacles, our model is designed to be both fast and accurate, with the dual capability of classifying anomalous objects and segmenting the regions where anomalies occur with unsupervised method which skip the need of labeled data during training.

A key insight from our research is that the teacher-student architecture exhibits a strong ability to distinguish between normal and anomalous images. In parallel, autoencoder architectures have proven effective in localizing the regions of anomalies by reconstructing images and highlighting discrepancies. Recognizing the individual strengths of these architectures, our proposed method integrates a multi-modal approach that combines the discriminative power of the teacher-student framework with the localization capability of autoencoders. This integrated approach is inspired by the EfficientAD method [1], which leverages both autoencoder and student-teacher paradigms to enhance anomaly detection performance.

EfficientAD originally suggested the use of a loss function (denoted as $\mathcal{L}_{\text{OOD}}$) to prevent overgeneralization in the model. However, when evaluated under the constraint of a 30% false positive rate (FPR), this approach did not sufficiently mitigate overgeneralization. Moreover, the reliance on an additional dataset beyond our primary plastic-nut subdataset introduced further complications. In response to these challenges, we modified the training architecture by using a novel loss function, termed $\mathcal{L}_{\text{CDO}}$[2], and by generating synthetic anomalies to simulate realistic defect scenarios. These synthetic anomalies, as illustrated in the accompanying figure, enable the model to optimize its behavior in detecting anomalies while retaining the autoencoder's proficiency in localizing defect regions.

Our final training pipeline is composed of three parallel components. The first component employs the student-teacher framework using normal images to learn robust representations of standard operating conditions. The second component applies the same framework to synthetically generated abnormal images, thereby enhancing the model's sensitivity to potential defects. The third component integrates the autoencoder with the teacher network to fine-tune the localization of anomalous regions. Although EfficientAD originally recommended an additional loss term to align the outputs of the autoencoder and the student network aimed at reducing false alarms in noisy backgrounds we found that this was unnecessary for our application. Our dataset(plastic-nut) from Real-IAD[3] is characterized by clear backgrounds, and the primary challenge lies in the variability of lighting conditions. We addressed these lighting challenges through a comprehensive data augmentation strategy, which included techniques such as image enhancement and adjustments to brightness and contrast.

In summary, our thesis presents a flexible and efficient method for industrial anomaly detection that harnesses the complementary strengths of the teacher-student and autoencoder architectures. By innovating on existing approaches and introducing new loss functions and training strategies, we have developed a model that is capable of both classifying anomalies and accurately segmenting their regions. This work not only contributes to the advancement of unsupervised anomaly detection in industrial settings but also provides a framework that can be adapted to meet the evolving needs of quality control and defect management in diverse manufacturing environments.

4.1 Dataset

High-quality datasets are essential for advancing computer vision, particularly in the realm of industrial anomaly detection. In such applications, the quality and diversity of image data directly influence the development of robust algorithms that can operate under the variable and often challenging conditions found in industrial environments. Industrial images are subject to a range of complexities, including variable lighting, reflections from metallic surfaces, occlusions by machinery, and the inherent requirement for high-resolution capture to detect subtle defects. Furthermore, the multi-view nature of production processes means that images captured from a single angle may fail to reveal all the critical details of a defect, thereby necessitating a dataset that not only encompasses a large volume of images but also a diverse set of perspectives.

Historically, early anomaly detection research relied on datasets such as KolektorSDD [51], which, while pioneering, contained only a single category and thus imposed significant limitations on both the evaluation and further development of algorithms. As research progressed, new datasets like MTD [52], MPDD [53], and BTAD [54] were introduced. Despite these advances, the relatively small number of categories and limited total image count in these datasets continued to restrict comprehensive algorithm evaluation. The advent of the MVTec AD dataset [17], which includes 15 industrial products divided into two types with a total of 5,354 images, marked a turning point. This dataset provided a more robust platform for research on conventional sensory industrial anomaly detection (IAD) tasks and spurred broader interest among researchers and practitioners alike. Building on this momentum, the VisA dataset [19] expanded the scale further by covering 12 objects in three types with a total of 10,821 images, thereby elevating the IAD dataset volume to the 10K image level and increasing the number of categories to 15.

Subsequent efforts have sought to address the limitations of these earlier datasets. For instance, Zhou et al. [55] proposed a synthetic pose-agnostic anomaly detection dataset intended to broaden the scope of research; however, the inherent differences between synthetic data and real-world samples have led to inconsistencies in evaluation metrics. Other datasets have attempted to incorporate three-dimensional information such as MVTec 3D AD [18], Eyecandies [56], and Real3D [33] to improve defect detection, yet these remain confined to relatively small scales and limited industrial scenarios.

The standard task in industrial anomaly detection is to determine whether an image of a target class contains an anomaly and, if so, to precisely localize the anomalous region. This task is inherently challenging because anomalous data is typically scarce, leading researchers to frame the problem as an unsupervised learning challenge where only normal data is available during training. A variety of

unsupervised approaches have emerged in recent years, including those based on data augmentation, reconstruction, and embedding techniques. In particular, embedding-based methods have evolved into several subcategories, including memory bank approaches, normalizing flow techniques, knowledge distillation methods, and classification-based frameworks, all of which have achieved commendable results under controlled conditions.

To overcome the limitations of previous datasets and better capture the complexity of industrial imaging, we used the RealIAD [3] dataset. Our experimental evaluations conducted on Real-IAD have revealed that state-of-the-art unsupervised anomaly detection algorithms which perform well on established datasets such as MVTec AD and VisA encounter significant challenges when applied to Real-IAD. This finding underscores the need for more robust algorithms capable of handling the diverse and complex conditions present in real-world industrial environments. The Real-IAD dataset, with its expansive scale, increased category diversity, and multi-view imaging, is poised to serve as a comprehensive resource that not only facilitates fair comparisons across different methods but also drives the development of more effective and practical solutions in industrial anomaly detection. We discussed about training pipeline in details and results of different model and architectures in details in 5.2.

4.1.1 MVTec-AD

The MVTec AD [17] dataset represents one of the most comprehensive and challenging resources for evaluating unsupervised anomaly detection methods in industrial settings. Comprising a total of 5,354 high-resolution images across 15 distinct categories including both objects (e.g., bottle, cable, capsule, hazelnut, metal nut, pill, screw, toothbrush, transistor, zipper) and textures (e.g., carpet, grid, leather, tile, wood) the dataset is designed to closely emulate real-world inspection scenarios encountered in manufacturing processes. The training set consists solely of defect-free images that capture the ideal or "normal" appearance of each category, whereas the test set is intentionally curated to include both pristine and anomalous samples. In the anomalous samples, defects manifest in over 70 unique forms, such as minute scratches, dents, contaminations, missing parts, and subtle structural deformations, with each anomaly annotated at the pixel level to enable precise evaluation of both global classification and localized segmentation performance. Images were acquired using a high-resolution industrial RGB sensor combined with bilateral telecentric lenses (with magnification factors of 1:1 and 1:5), ensuring uniform scale and minimal perspective distortion across the dataset. The resulting images, which typically range in resolution between 700×700 and 1024×1024 pixels, capture a level of detail that is crucial for identifying even the most subtle defects. Controlled illumination conditions during acquisition minimize extraneous variability, thereby

ensuring that the detected anomalies are genuine reflections of physical defects rather than artifacts of inconsistent lighting. In 4.1, sample images are presented to illustrate these characteristics: one panel shows a defect-free image used for training, while other panels exhibit various test images where anomalies such as barely perceptible surface cracks and localized contaminations are evident. These visual examples underscore the dataset's capacity to challenge detection algorithms at both the image and pixel levels.

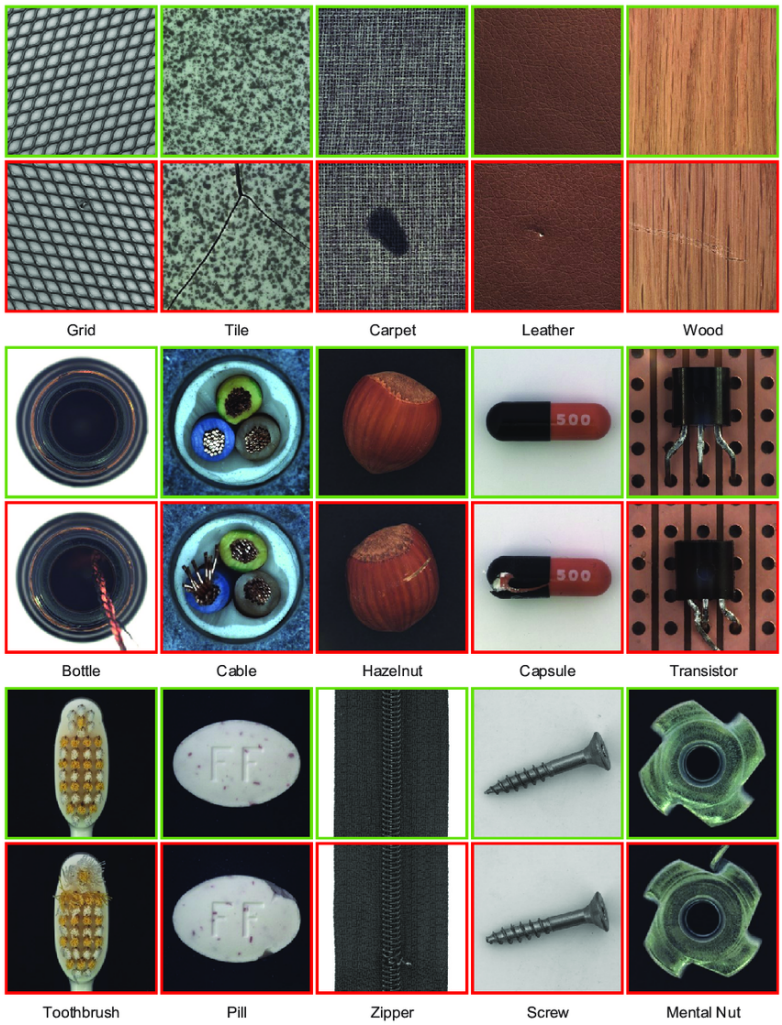


Figure 4.1: Sample images from MVTecAD dataset used in anomaly detection papers.

The structured composition of the MVTec AD dataset allows for a multifaceted evaluation of anomaly detection methods. At the image level, many algorithms assign a single anomaly score per image, with performance commonly quantified using metrics such as the Area Under the Receiver Operating Characteristic

Curve (AUROC). This metric is particularly valuable in settings where a binary decisionnnormal versus anomalous is required. However, the true strength of the dataset lies in its support for pixel-level segmentation. Here, the pixel-precise annotations enable researchers to generate detailed anomaly maps and compute evaluation metrics that assess the spatial accuracy of the segmentation. Metrics such as the per-pixel AUROC and the relative per-region overlap between predicted anomaly regions and the ground truth provide deep insights into the effectiveness of a given method in isolating and localizing defects.

The dataset's relevance extends beyond mere benchmarking; it mirrors the real-world challenges inherent in industrial inspection tasks. In practical applications, defective samples are typically sparse due to stringent quality control protocols, making it imperative for detection systems to learn the concept of "normality" from abundant defect-free data and then generalize effectively to unseen, subtle anomalies. This inherent imbalance between normal and anomalous instances makes the MVTec AD dataset an ideal proxy for testing unsupervised anomaly detection methods. Researchers have leveraged this dataset to develop and refine a variety of approaches including convolutional autoencoders, generative adversarial networks, and methods based on feature extraction from pre-trained convolutional neural networks that are tailored to capture fine-grained deviations from normality.

Moreover, the pixel-level annotations provided in the dataset are critical for advancing research in anomaly segmentation. These annotations allow for a granular assessment of an algorithm's ability to not only flag an image as anomalous but also accurately delineate the exact regions where defects occur. For example, in defect segmentation tasks, even minor discrepancies in reconstruction quality or localization precision can be quantitatively assessed through overlap metrics and per-pixel error rates. Such detailed evaluation is indispensable for applications where precise defect localization directly informs subsequent corrective actions or quality control decisions. The comprehensive nature of MVTec AD is further enhanced by its rigorous acquisition and annotation protocol. Each image depicts a unique physical sample, and the dataset deliberately avoids data augmentation that would result in redundant views of the same object. Instead, every sample is an independent observation, which ensures that the variability within the dataset authentically reflects the diversity encountered in industrial production lines. This characteristic, combined with the high-resolution nature of the images and the controlled acquisition environment, renders the dataset particularly suited for benchmarking state-of-the-art unsupervised anomaly detection and localization methods.

Additionally, the MVTec AD dataset is distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0). This open-access licensing has facilitated its widespread adoption within the research community, making it a de facto standard for comparative evaluations in

the field of anomaly detection. The broad usage of the dataset in numerous studies has also contributed to an evolving understanding of the challenges associated with detecting subtle and complex anomalies in high-resolution images.

Overall, the MVTec AD dataset serves as a robust and challenging benchmark for unsupervised anomaly detection. Its combination of high-quality images, diverse defect types, and meticulously annotated ground truth provides an ideal platform for evaluating both global anomaly classification and detailed defect segmentation. The dataset not only mirrors the complexities of real-world industrial scenarios but also pushes the boundaries of current methodologies, driving innovation toward more sensitive, accurate, and reliable detection systems.

4.1.2 Real-IAD

The Real-IAD dataset has been meticulously developed to provide a more comprehensive benchmark for industrial anomaly detection. As explained in [3], the dataset construction follows a systematic process involving material selection, imaging system design, and data annotation. The dataset includes 30 distinct objects composed of different materials, including metal, plastic, wood, ceramics, and composite materials. To ensure diverse and representative anomaly detection scenarios, various types of defects, such as missing parts, dirt, deformation, pits, cracks, scratches, and structural damage, were manually introduced. These samples were then processed and prepared for image collection using a well-structured imaging system.

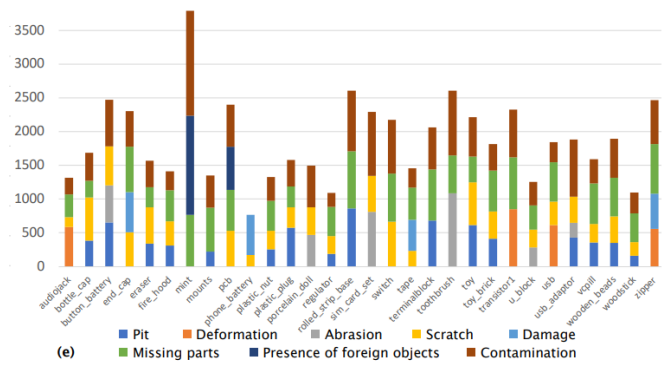


Figure 4.2: Distribution of data volume across different defect categories from Real-IAD paper.

The imaging setup consists of a multi-view camera system, designed to capture high-resolution images from multiple perspectives. The system employs five cameras: one capturing a top-down perspective and four others positioned symmetrically at approximately 45-degree angles. To enhance defect visibility, a ring light source was

positioned above the object to improve image clarity and highlight minor anomalies. In practical industrial applications, automated machinery is often used to flip parts for thorough inspection. However, to maintain consistency in dataset abstraction, the dataset standardizes the multi-view imaging to five fixed perspectives, ensuring that the captured data remains representative while also being computationally feasible. The imaging system utilizes a HIKROBOT MV-CE200-10GC camera with a resolution of 3,648×5,472 pixels [3], which provides detailed images suitable for fine-grained anomaly detection.

Following image collection, the data underwent a rigorous annotation and cleaning process. Manual verification was performed to confirm the correctness of both normal and anomalous images. The dataset was subsequently annotated at the pixel level using LabelMe, providing highly detailed defect segmentation. To ensure high annotation quality, the model's predictions were cross-checked against manual annotations. Any inconsistencies were iteratively reviewed and corrected until the annotation accuracy stabilized, ensuring a clean and reliable dataset for training and evaluation.

A comparison between Real-IAD and existing datasets, such as MVTec AD [17] and VisA [19], reveals significant improvements in dataset scale, diversity, and challenge complexity. The dataset contains a significantly higher number of object categories and provides fine-grained segmentation labels along with multi-view images. Statistical analysis, as demonstrated in 4.2, shows that Real-IAD is notably larger than existing datasets, with an order-of-magnitude increase in both normal and anomalous samples. The dataset also exhibits a higher proportion of defective areas and a broader range of defect types, making it substantially more challenging for anomaly detection algorithms. Furthermore, the dataset maintains a balanced distribution of normal and anomalous samples across different categories, ensuring that models trained on it generalize better to real-world applications.

The dataset offers several key advantages that distinguish it from existing benchmarks. First, its diversity ensures broader coverage of object categories and real-world scenarios, making it a valuable resource for developing robust anomaly detection models. Second, its scale is unprecedented, with over 150,000 images, significantly surpassing the size of previously available datasets. This increase in dataset size allows for more comprehensive evaluations and enhances the statistical significance of experimental results. Third, its complexity introduces a greater level of difficulty, encouraging the development of more advanced and capable anomaly detection algorithms. The inclusion of a wide range of defect types, combined with multi-view imaging, ensures that the dataset reflects real-world challenges more accurately.

Evaluation of the dataset follows two primary settings: Unsupervised Industrial

Anomaly Detection[1] and Fully Unsupervised Industrial Anomaly Detection. The UIAD setup assumes that training data consists solely of normal samples, with both normal and anomalous images included in the test set. This is a widely used protocol in anomaly detection research and provides a standard baseline for evaluation. The FUIAD setting, however, presents a more realistic scenario, allowing for the inclusion of anomalous samples in the training set. This setting is rarely explored in existing datasets due to the difficulty of obtaining sufficient anomalous samples, but Real-IAD provides the necessary data diversity to support such experiments.

The evaluation metrics employed for dataset benchmarking include the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Per-Region Overlap Curve (AU-PRO), following the methodologies described in sections 2.5.4, 2.5.3.

Comparative benchmarking against existing datasets, such as MVTec AD and VisA, further highlights the advantages of Real-IAD. Experimental results, as summarized in [3], show a significant drop in model performance from MVTec (97.9% AUROC) to Real-IAD (85% AUROC) on average, indicating that Real-IAD presents a considerably greater challenge. This increased difficulty is due to the dataset's higher diversity, larger number of object categories, and multi-view complexity. Furthermore, Real-IAD allows for a more meaningful comparison of different anomaly detection methods, as existing datasets often exhibit near-saturated performance (98-99% AUROC), making it difficult to distinguish between algorithmic improvements.

Additionally, for our industrial project, we specifically utilized a sub-dataset from Real-IAD known as Plastic-Nut, which was particularly relevant to our application due to its alignment with common industrial defect patterns. This sub-dataset contained various types of defects, including surface scratch, pit, contamination and missing parts that frequently arise in the manufacturing of small mechanical components. These defects can significantly impact product integrity and operational efficiency, necessitating precise and reliable anomaly detection methodologies. In Figure 4.3 you can see some sample of normal objects that used only for training phase of the project.

To capture the full scope of defect variations, the dataset provides high-resolution images from multiple perspectives, allowing for a more robust evaluation of defect visibility under different lighting conditions and angles. Figure 4.4 illustrates representative examples of Plastic-Nut samples used for evaluation, highlighting the diverse nature of defects and the advantages of multi-view imaging in overcoming occlusions and perspective limitations. The ability to analyze defects from different

[1] `https://github.com/Sunny5250/Awesome-Multi-Setting-UIAD`

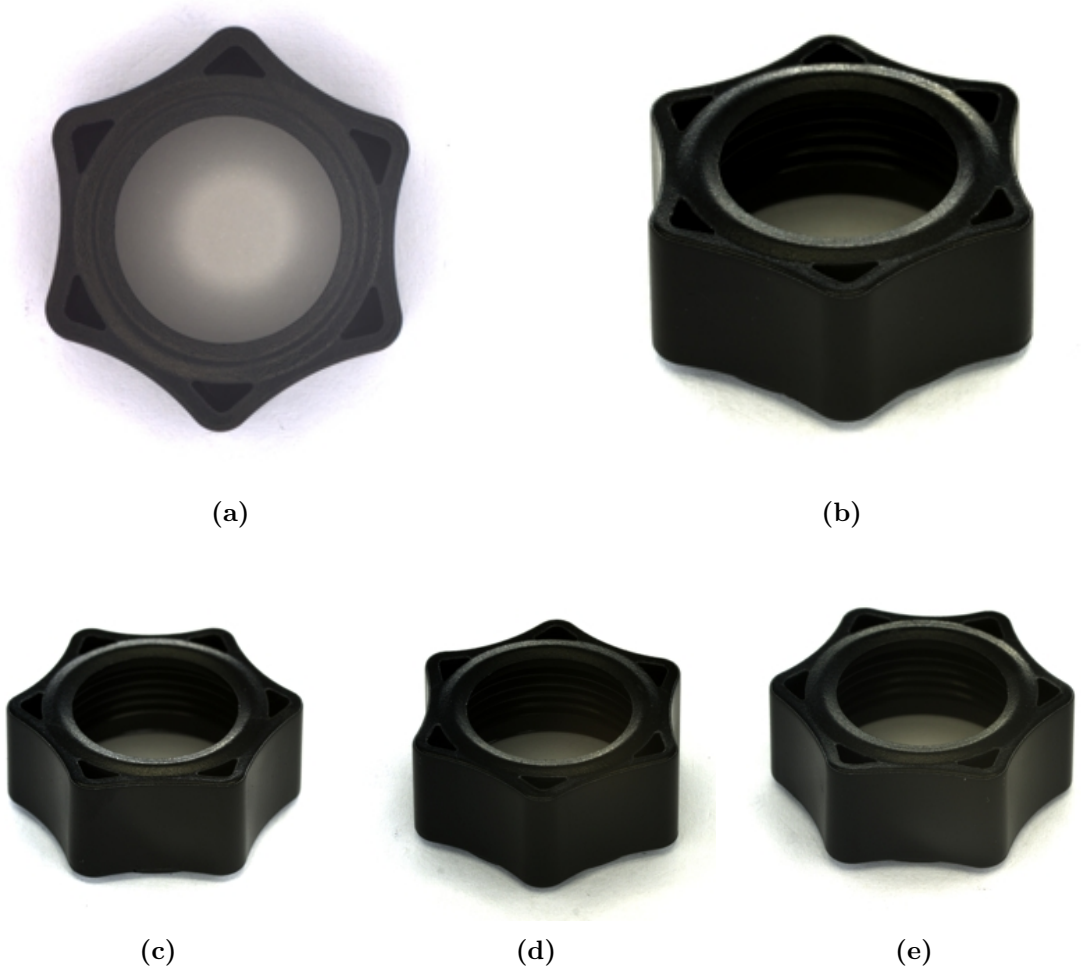
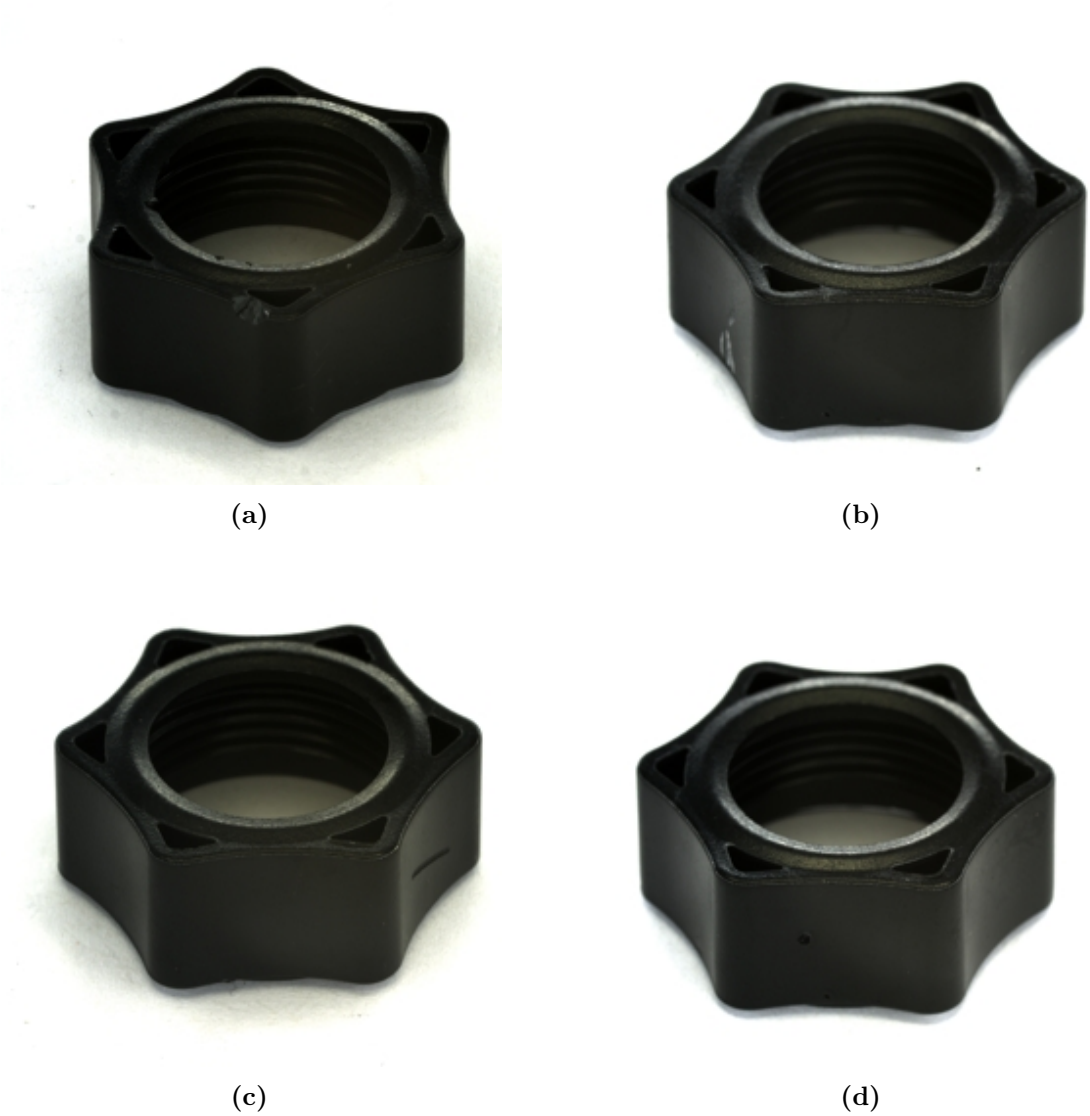


Figure 4.3: Example of anomaly-free images from the dataset plastic-nut used for training. Five different aspects of one object are provided.

viewpoints ensured that subtle defects, which might be invisible in single-view inspection, could be accurately detected. Furthermore, the inclusion of Plastic-Nut in our study enabled the development of a more adaptable and generalizable anomaly detection framework, specifically tuned to the intricacies of real-world manufacturing workflows. This sub-dataset's integration into our research further underscores the practical applicability of Real-IAD in industrial settings, where precise anomaly detection is crucial for minimizing defects and improving operational efficiency. The Plastic-Nut sub-dataset serves as a benchmark for evaluating model performance in detecting intricate structural flaws, which are often overlooked in conventional quality inspection systems. By leveraging multi-view high-resolution imaging and pixel-level annotations, this dataset provides a comprehensive reference for

training deep learning models capable of robust defect identification. Additionally, its inclusion enhances predictive maintenance strategies by facilitating the early detection of structural inconsistencies, thereby reducing downtime and increasing production reliability in automated assembly lines. The systematic evaluation using Plastic-Nut within the Real-IAD framework demonstrates its scalability and effectiveness in diverse manufacturing scenarios, further validating its role in advancing the field of industrial anomaly detection.



(a) **(b)**

(c) **(d)**

Figure 4.4: Example of abnormal images from the dataset plastic-nut used for training. Missing parts (a), contamination (b), scratch (c), and pit (d) are illustrated.

4.1.3 Data Augmentation

This section we provide an in-depth explanation of the data augmentation techniques used in our anomaly detection system. The goal of these augmentations is to increase the diversity of the training data, making the model more robust to variations in real-world images. We explain each concept in detail below.

Default Transform

The first step in our pipeline is to standardize all images using a default transform. This ensures that every image, regardless of its source, has the same size, structure, and range of pixel values before any further processing is applied. three most important step is implemented in this augmentation which we explain it below:

- **Resizing:** Every image is resized to a fixed resolution of 256×256 pixels. This guarantees that all images have the same dimensions, which is crucial because the neural network expects inputs of a consistent size. In some cases of anomaly detection we would prefer to use image with size 512×512. Hence, in this case we shall change this size of augmentation depends on situation.

- **Conversion to Tensor:** After resizing, each image is converted from its original format (usually a PIL image or a NumPy array) into a tensor. A tensor is a multi-dimensional array that serves as the basic data structure for computation in PyTorch. This step also scales the pixel values from the typical range of 0 to 255 down to 0 to 1, which is more manageable for learning algorithms.

- **Normalization:** Finally, the pixel values are normalized using the mean and standard deviation values derived from the ImageNet dataset. This normalization aligns the image statistics to those that the network is commonly trained on, helping to stabilize and speed up the learning process.

This default transform is essential as it creates a consistent starting point for every image before any further augmentations are applied.

Appearance Augmentation

Appearance augmentation introduces controlled variations in the image's visual characteristics. This technique is used to simulate different lighting conditions, camera settings, or environmental factors. By doing so, the model learns to focus on the important structural features rather than getting distracted by minor differences in color or brightness.

- **ColorJitter:** This operation randomly changes the brightness, contrast, and saturation of the image. For example, an image might be made slightly brighter or darker, or its colors might be intensified or muted. These small changes help the model learn to ignore superficial color differences.
- **AdjustSharpness:** A custom transformation that adjusts the image sharpness. Increasing sharpness can make the edges in the image more pronounced, which may help in detecting fine details.
- **AdjustGamma:** This custom transform applies gamma correction. Gamma correction is a non-linear operation used to encode and decode luminance or tristimulus values, effectively changing the image's overall brightness in a non-linear way.

The transformations are wrapped within a RandomChoice operator. This means that for each image, only one of the available appearance adjustments is applied. This random selection helps maintain variability without over-complicating each individual image.

By applying these random appearance modifications, the model becomes less sensitive to differences in lighting, exposure, and color balance. This encourages the learning of robust, high-level features that are invariant to these types of changes. Furthermore, it is good to consider that the type of augmentation such as changing brightness, sharpness and gamma is completely relative to type of object and it environment that image is taken.

Synthetic Anomaly Injection

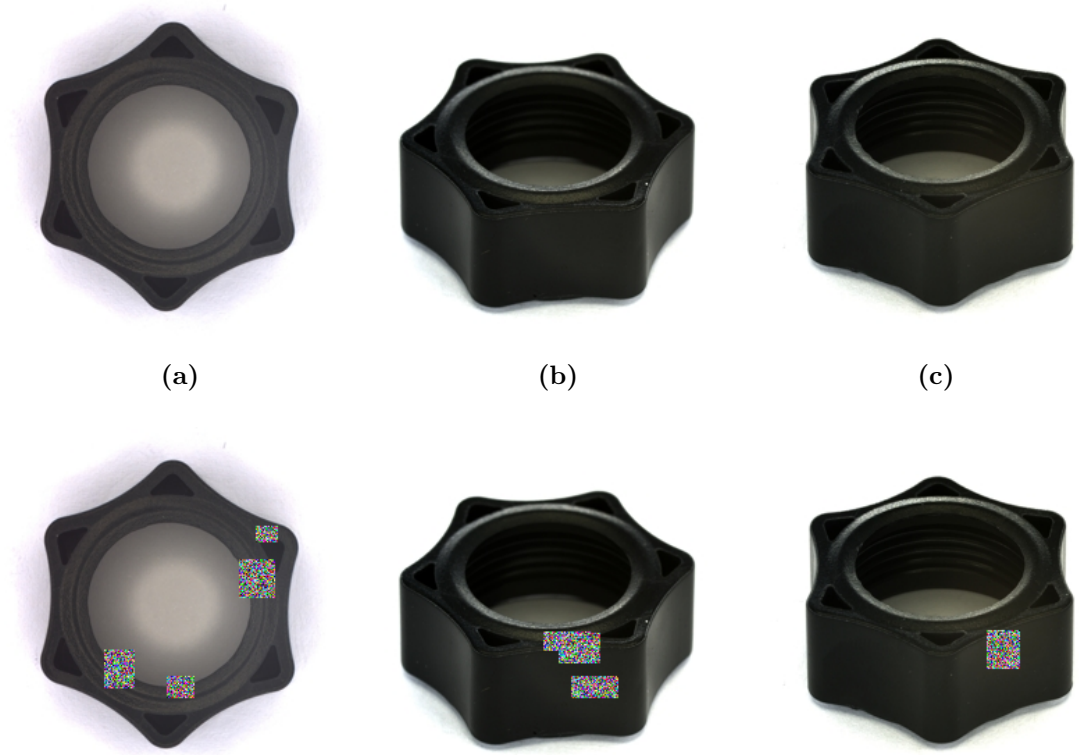
Synthetic anomaly injection is a specialized augmentation technique designed specifically for anomaly detection tasks. In many practical scenarios, real anomalies (defects or unusual patterns) are rare. To address this, we artificially inject anomalies into normal images so that the model can learn what an anomaly looks like.

1. **Decision to Inject:** For each image, a random decision is made based on a predefined probability (in our case, 40%). If the decision is affirmative, the algorithm proceeds to inject anomalies.
2. **Determining the Number of Patches:** When an anomaly is to be added, the system randomly selects between 1 and 4 patches to simulate defects.
3. **Patch Generation and Placement:**
    - The algorithm randomly chooses the size of each patch, typically a small fraction (between $\frac{1}{40}$ and $\frac{1}{10}$) of the image size.

- It then selects a random location for each patch. If the option skip background is enabled, the algorithm uses a simple background estimation method (based on intensity thresholds and morphological operations) to avoid placing patches over background regions.

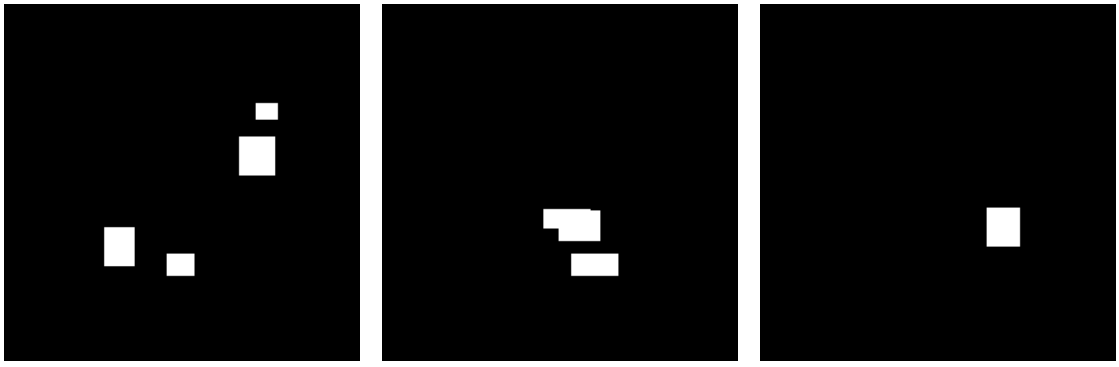
4. **Noise Injection:** The chosen regions in the image are replaced with random noise, simulating the presence of a defect.

Figure 4.5 shows a clear example. The upper portion of the figure depicts an original image, while the lower portion shows the same image after synthetic noise patches have been injected.



(a) (b) (c)

(d) (e) (f)



(g) (h) (i)

Figure 4.5: Example of Synthetic Anomaly Injection and Ground Truth. **Top row:** Original images. **Middle row:** Images with synthetic noise patches added. **Bottom row:** Corresponding ground truth images.

For a complete and detailed explanation of how synthetic anomalies are injected including pseudocode, parameter settings, and further examples please see Section 4.6. This additional section covers every aspect of the algorithm in depth.

In summary, our data augmentation pipeline includes the following:

- **Default Transform:** Standardizes input images by resizing them, converting to tensors, and normalizing the pixel values.
- **Appearance Augmentation:** Introduces variability by randomly adjusting brightness, contrast, saturation, sharpness, and gamma.
- **Synthetic Anomaly Injection:** Creates artificial defects by inserting noise patches into images, helping the model learn to detect anomalies.

4.2 Patch Description Network

The concept of patch-based processing has become a fundamental approach in modern computer vision, enabling more efficient and accurate image analysis. A Patch Description Network (PDN) is a term that can be used to describe models that focus on extracting meaningful representations from small image regions (patches). These networks are crucial in tasks like image matching, feature extraction, and object recognition. Various architectures leverage patch-based processing, and several studies highlight its importance. One of the primary applications of patch-based networks is patch matching, where small regions from different images are compared to determine their similarity. Traditional methods like SIFT [9] and ORB[57] have been widely used for this purpose, but deep learning-based approaches have demonstrated superior performance. For example, Melekhov [58] introduced a neural network-based descriptor designed specifically for patch matching. Their approach maps raw image patches to a low-dimensional feature space, ensuring that visually similar patches are closer in this learned space. This deep-learning-based descriptor outperformed classical methods in keypoint matching and image retrieval tasks [58].

Another notable contribution to patch-based processing is the Patch-Level Vision Similarity Compare Network (PL-VSCN)[59] . This network tackles image matching by breaking down the challenge of comparing whole images into a patch-by-patch comparison. Instead of focusing on global image structures, the PL-VSCN ensures that corresponding patches between two images are identified with high accuracy. This method is particularly effective in scenarios where images may have significant occlusions or background noise, as it reduces the influence of non-corresponding regions [59]. With the rise of Transformer-based architectures in vision tasks, patch-based representations have gained even more prominence. Vision Transformers (ViTs), introduced by Dosovitskiy [60]. in 2020, process images by dividing them into fixed-size patches, treating each patch as an individual token, similar to words in NLP tasks. These patches are embedded and passed through a transformer model, capturing long-range dependencies and providing a global understanding of the image. This patch-based approach has proven to be a strong alternative to traditional convolutional neural networks (CNNs), achieving state-of-the-art results in classification and object detection [60].

Recent anomaly detection methods commonly use the features of a deep pre-trained network, such as a WideResNet-101. However, we used a Patch Description Network (PDN) with a drastically reduced depth as a feature extractor [1]. This PDN consists of only four convolutional layers, where each output neuron has a receptive field of 33×33 pixels, making each output feature vector correspond to a specific patch. Due to this clear correspondence, the network is termed a patch description network [1]. The PDN is fully convolutional and can process images of

variable sizes in a single forward pass.

Unlike traditional deep networks, PDN introduced by EfficientAD reduces computational overhead using strided average-pooling layers after the first and second convolutional layers. This downsampling approach improves runtime and memory efficiency compared to existing methods like Student–Teacher, which lack such mechanisms [1].

To make the PDN generate expressive features, we used the idea introduced in paper EfficientAD [1]. By distill a deep pretrained classification network into it. Specifically, we use the same pretrained features as PatchCore from a WideResNet-101 and train the PDN on ImageNet by minimizing the mean squared difference between its output and the features extracted from the pretrained network. This approach ensures that the PDN retains useful representational capabilities while being computationally efficient. The PDN's design also provides another advantage: each feature vector depends only on its respective 33×33 patch, eliminating long-range dependencies that are common in pretrained classifiers. Unlike PatchCore's feature extractors, which allow anomalies in one region of the image to influence distant feature vectors, the PDN provides highly localized anomaly detection, improving precision.

In addition to its efficiency, the PDN framework allows seamless integration into various image analysis tasks, including object localization and segmentation. By leveraging its well-defined receptive fields, it can be adapted for use in multi-scale feature extraction, improving performance in applications requiring fine-grained spatial details [1]. Furthermore, the PDN's adaptability enables it to serve as a general-purpose feature extractor, which can be fine-tuned for specialized domains such as medical imaging and remote sensing. As deep learning continues to evolve, PDN-like architectures hold significant potential for optimizing computational efficiency while maintaining high feature expressiveness, providing a promising direction for future research in computer vision.

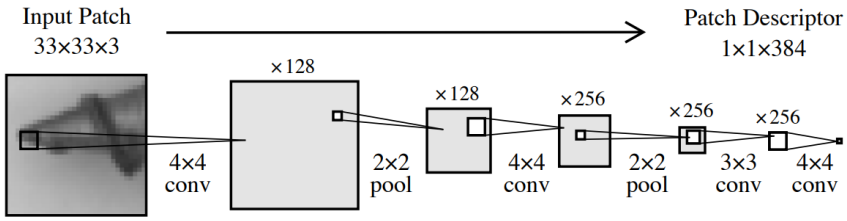


Figure 4.6: Patch description network (PDN) architecture of EfficientAD-S [1]. Applying it to an image in a fully convolutional manner yields all features in a single forward pass.

Expanding on its applications, the PDN can be used in real-time video processing,

where efficiency and speed are critical. Its ability to generate compact yet descriptive features makes it ideal for video anomaly detection, object tracking, and event recognition. Additionally, PDN-based architectures can be integrated into robotics, enabling autonomous navigation and scene understanding in environments with limited computational resources. With increasing demands for lightweight, high-performance deep learning models, the PDN presents an opportunity for more scalable and deployable computer vision solutions across diverse fields, including security surveillance, augmented reality, and industrial automation. Future research will continue exploring ways to refine PDN architectures, optimizing them for even greater efficiency while maintaining accuracy and robustness. Finally, you can find the pipeline of this network in details and how it will be used in this thesis in order to have patching in Section 5.

4.3 Pretraining

In the following, we describe how to distill the WideResNet-101 [61] features used by PatchCore [41] into the teacher network T. The distillation training algorithm is presented in Algorithm 1. The process is analogous for other pretrained feature extractors.

There are only a few requirements regarding the output shape of the feature extractor. The feature extractors used by PatchCore output features of shape $384 \times 64 \times 64$ for an input image size of 512×512 pixels. Therefore, the teacher and the autoencoder also output 384 channels we will discuss about it more in next sections. If a pretrained feature extractor outputs a different number of channels, this default of 384 output channels of the teacher and the autoencoder can be adjusted flexibly. During distillation, we resize input images to 512×512 for the pretrained feature extractor and to 256×256 for the teacher network that is being trained. This results in an output shape of $384 \times 64 \times 64$ for the teacher network as well. If a feature extractor outputs feature maps of a size other than 64×64, we can adjust its input image size to achieve an output feature map size of 64×64. Alternatively, we can adjust the input image size of the teacher network because it is fully convolutional and operates separately on patches of size 33×33. A feature map size of 53×71, for example, can be achieved by applying the teacher network to images of size 212×284.

We use a batch size of 16 for the distillation training and use ImageNet [14] as the pretraining dataset. We use the official implementation of PatchCore [41] and its default values if not stated otherwise. We use the feature postprocessing of PatchCore as well, which includes pooling features from two layers and projecting each feature vector to a reduced dimensionality of 384 dimensions, as described in [1]. The features used for our distillation training are the final features used by PatchCore, i.e., those given to the coreset subsampling algorithm when training PatchCore. We denote the WideResNet-101-based feature extractor, including the feature postprocessing, as

$$\Psi : \mathbb{R}^{3\times512\times512} \rightarrow \mathbb{R}^{384\times64\times64}.$$

Distillation Training Algorithm

Algorithm 1 shows the overall procedure to distill the features of a pretrained backbone (WideResNet-101 in this case) into a new teacher network T. In practice, this is implemented in PyTorch, but below we provide a more concise and professional pseudo-code version. The main idea of this algorithm is based on [1] paper.

Algorithm 1 Distillation Training for Teacher Network T

Require: Pretrained backbone Backbone (WideResNet-101), feature-extractor module extractor that outputs $384 \times 64 \times 64$ features, teacher network T with 384 output channels, dataset $\mathcal{D}$ (e.g., ImageNet), batch size $B = 16$, total steps $N = 60000$, Adam optimizer with learning rate $\eta = 10^{-4}$ and weight decay 1×10^{-5}.

Define transformations:
  $\text{transform}_{\text{teacher}}(\cdot)$: Resize images to 512×512 and apply standard normalization.
  $\text{transform}_T(\cdot)$: Resize images to 256×256 and apply the same normalization.
  Optionally, apply random grayscale augmentation to both.
Compute normalization statistics for teacher features:
  (1) Initialize arrays for storing feature means and variances.
  (2) For a subset of S images from $\mathcal{D}$ (e.g., $S = 10000$):
    (a) Transform each image with $\text{transform}_{\text{teacher}}(\cdot)$.
    (b) Extract teacher features $f \leftarrow \text{extractor}(\text{image}) \in \mathbb{R}^{384\times64\times64}$.
    (c) Accumulate mean μ_f and variance σ_f^2 across channels.
  (3) Compute final μ^Φ and σ^Φ by averaging.
Prepare teacher network T for training:
  Execute `T.train()` and move to GPU if available.
Initialize the Adam optimizer:
  Set optimizer $\leftarrow \text{Adam}(T.\text{parameters}(), \eta, \text{weight\_decay} = 1 \times 10^{-5})$.
Main training loop:
for $i = 1$ **to** N **do**
  Load a batch of images $(x_1, x_2, \ldots, x_B)$ from $\mathcal{D}$.
  Transform for teacher feature extraction: $x^{\text{teacher}} \leftarrow \text{transform}_{\text{teacher}}(x)$.
  Transform for teacher network training: $x^T \leftarrow \text{transform}_T(x)$.
  Compute target features: $F^{\text{target}} \leftarrow \text{extractor}(x^{\text{teacher}}) \in \mathbb{R}^{384\times64\times64}$.
  Normalize: $F^{\text{target}} \leftarrow \dfrac{F^{\text{target}} - \mu^\Phi}{\sigma^\Phi}$.
  Forward pass teacher network: $F^T \leftarrow T(x^T)$, with $F^T \in \mathbb{R}^{384\times64\times64}$.
  Compute distillation loss: $\mathcal{L} \leftarrow \text{MSE}(F^{\text{target}}, F^T)$.
  Backpropagation and update:
    `optimizer.zero_grad()`
    $\mathcal{L}$.backward()
    `optimizer.step()`
  Checkpoint (optional): Every k iterations, save the teacher network T.
end for
Save final teacher network weights. **return** Trained teacher network T.

some comments on Algorithm 1:

- **Infinite Dataloader:** For practical purposes, the code may wrap a standard `DataLoader` in an *infinite* loader that loops over the dataset indefinitely until N training steps are reached.
- **Feature Normalization:** The `feature_normalization` function estimates the mean and variance of the teacher features channel-wise, which are then applied during training to match the distribution of the teacher.
- **Patch-Based Network:** The `PatchMaker` class and subsequent modules (`Preprocessing`, `Aggregator`, etc.) are used to break the teacher's intermediate features into patches and produce consistent dimensionalities, ensuring the final output is $384 \times 64 \times 64$.
- **Saving Models:** The code periodically saves both the full model (`.pth` file) and the state dictionary (`.pth` file) for easier reloading. Based on which type of PDN we use, the final teacher model will be medium sized teacher (medium PDN) or small sized teacher (small PDN).

4.4 Student-Teacher

The result of 5.2.2 will be our Teacher in the connection of student-teacher part which some papers called it expert and apprentice. The student-teacher (S-T) architecture is an advanced and highly efficient framework widely used in anomaly detection, particularly in industrial and real-world applications where computational efficiency is crucial. This method leverages the interaction between two neural networks: the teacher network, which serves as a pre-trained reference model, and the student network, which is trained to closely approximate the teacher's outputs on normal (non-anomalous) data. The fundamental idea is that since the student is only exposed to normal data during training, it fails to generalize properly when encountering anomalies. This discrepancy between the teacher's and student's outputs during inference is utilized as an anomaly indicator. By capturing deviations, this framework effectively detects various types of anomalies, including structural inconsistencies and logical errors that may arise in different application domains.

One of the most significant advantages of the student-teacher approach is that it does not require labeled anomalous data, making it particularly useful in scenarios where anomalies are rare or difficult to define. Unlike traditional supervised learning methods that depend on pre-labeled datasets containing both normal and anomalous instances, the student-teacher model is self-supervised and learns only from normal data. This eliminates the challenge of obtaining diverse anomalous samples for training. Furthermore, since inference involves only forward passes through the student and teacher networks, the computational overhead is significantly reduced compared to other anomaly detection approaches. Additionally, this method exhibits strong generalization to diverse types of anomalies without requiring explicit modeling of each anomalous scenario, as the deviation between the teacher and student networks naturally captures deviations from expected normal behavior.

The student-teacher model operates through a structured process involving multiple steps. First, the teacher network, typically a pre-trained deep neural network such as a convolutional neural network (CNN), extracts meaningful feature representations from input data which we explained this phase in 5.2.2. The student network is then trained to mimic these feature representations using normal data exclusively. At inference time, anomalies are identified based on the deviation between the student's and teacher's outputs, measured through predefined distance metrics or loss functions. The greater the deviation, the more likely the input is anomalous. This approach is particularly effective in domains such as industrial inspection, medical diagnostics, and cybersecurity, where detecting subtle deviations is critical.

Training the student-teacher model is performed using various loss functions

that shape how the student network learns to replicate the teacher's outputs. The most commonly used loss function is the Mean Squared Error (MSE) loss, which minimizes the difference between the feature representations of the teacher and student. However, to enhance the model's ability to detect anomalies, additional loss mechanisms can be introduced. Hard feature loss focuses on challenging feature components where the student struggles to match the teacher, forcing the student to learn only the essential aspects of normal data while remaining sensitive to anomalies. Furthermore, in [1] mentioned an out-of-distribution penalty loss is sometimes employed to penalize the student for attempting to generalize beyond its trained distribution, ensuring that it does not learn unintended patterns from unseen data. but during training on plastic-nut dataset we found this loss function unrelated.

Hard feature loss focuses on challenging feature components where the student struggles to match the teacher, forcing the student to learn only the essential aspects of normal data while remaining sensitive to anomalies.

The Mean Squared Error (MSE) loss is formulated as:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} |T(x_i) - S(x_i)|^2 \tag{4.1}$$

where:

$T(x_i)$ represents the feature output of the teacher network for input x_i.

$S(x_i)$ represents the corresponding feature output of the student network.

N is the number of training samples.

The Hard Feature Loss is designed to emphasize the most challenging feature components where the student deviates significantly from the teacher. It is expressed as:

$$L_{hard} = \frac{1}{|H|} \sum_{(c,w,h)\in H} (T(x)c, w, h - S(x)c, w, h)^2 \tag{4.2}$$

where H is the set of feature dimensions with the highest reconstruction errors, forcing the student to focus on difficult-to-replicate aspects of normal data.

Once trained, anomaly detection is performed through a systematic inference process. Given an input image or data sample, both the teacher and student networks process it, and the squared difference between their outputs is computed to generate an anomaly map. The anomaly score is then derived from this map, and a threshold is applied to determine whether an input is anomalous. In order to choose the hardest ones we choose just 99% upper quantile of this distance of student with teacher.

The student-teacher model has demonstrated remarkable success across a variety of application domains. In industrial settings, it is used for quality control and

defect detection in manufacturing processes, where even minor deviations from normal production patterns can indicate defects. In medical imaging, the model aids in detecting abnormalities in X-rays, MRIs, and other scans by highlighting regions that deviate from expected normal anatomy. In cybersecurity, it is applied to identify unusual network behavior, helping to detect potential cyber threats and intrusions. The versatility and efficiency of this framework make it a powerful tool in anomaly detection, particularly in environments where obtaining labeled anomalous data is impractical or infeasible. In summary, the student-teacher framework provides a robust, efficient, and generalizable approach to anomaly detection. By leveraging a teacher network as a reference model and training a student network to replicate its outputs on normal data, this method effectively identifies anomalies based on deviations observed at test time.

4.5 Autoencoder-Teacher

Logical anomalies in images can arise from various inconsistencies, including missing, misplaced, or surplus objects, as well as violations of geometrical constraints, such as an incorrect screw length. To effectively model and detect such logical inconsistencies, we utilize an autoencoder-based approach, inspired by the recommendations of the MVTec LOCO dataset [18]. The primary function of the autoencoder is to learn and encode the logical constraints inherent in the training images, thereby allowing violations of these constraints to be detected when presented with anomalous samples.

The core anomaly detection mechanism in EfficientAD [1] consists of a student-teacher network along with an autoencoder. The autoencoder, denoted as A, is trained to replicate the output of the teacher network. Mathematically, given an input training image I, the autoencoder generates an output representation $A(I) \in \mathbb{R}^{C \times W \times H}$. The corresponding loss function, ensuring the proper training of the autoencoder, is formulated as:

$$L_{AE} = \frac{1}{CWH} \sum_{c} \|T(I)_c - A(I)_c\|_F^2, \tag{4.3}$$

where $T(I)$ represents the output of the teacher network, and the Frobenius norm is used to measure the reconstruction error. This loss ensures that the autoencoder learns to approximate the teacher's outputs effectively.

The architecture of the autoencoder follows a standard convolutional design, utilizing strided convolutions in the encoder phase and bilinear upsampling in the decoder phase. Detailed layer hyperparameters are provided in 5.2. Unlike the patch-based student model, which processes smaller local regions, the autoencoder must encode and decode entire images through a bottleneck of 64 latent dimensions. This constraint poses challenges, particularly when dealing with logical anomalies. Since the autoencoder is optimized on normal images, its latent representation for anomalous images typically diverges significantly from expected reconstructions. Furthermore, autoencoders are known to struggle with reconstructing fine-grained patterns, leading to systematic reconstruction artifacts even for normal images, such as blurry textures in background grids [62].

To mitigate the risk of false-positive anomaly detections caused by systematic reconstruction artifacts, EfficientAD introduces an additional component to its loss function. Specifically, the student network is extended to predict both the output of the teacher network and the output of the autoencoder. Denoting the student's additional output channels as $S'(I) \in \mathbb{R}^{C \times W \times H}$, we define an auxiliary loss function as:

$$L_{STAE} = \frac{1}{CWH} \sum_{c} \|A(I)_c - S'(I)_c\|_F^2 \tag{4.4}$$

By training the student network to capture the systematic reconstruction errors of the autoencoder on normal images, it learns to disregard such artifacts while preserving sensitivity to true anomalies. Importantly, the student network is not exposed to anomalous examples during training, ensuring that it does not learn their reconstruction characteristics. Consequently, the discrepancy between the autoencoder's output and the student's output serves as a meaningful anomaly map.

This discrepancy-based anomaly detection approach is structured in two forms: a local anomaly map derived from the student-teacher network and a global anomaly map obtained from the student-autoencoder interaction. The final combined anomaly map is computed as their average, with the maximum anomaly score across the image determining the final anomaly classification. By leveraging shared hidden layers in the student network, this method achieves computational efficiency while maintaining robust detection of both structural and logical anomalies, making EfficientAD a powerful framework for unsupervised anomaly detection.

4.6 Collaborative Discrepancy

Collaborative Discrepancy Optimization (CDO) is a novel framework introduced on 2023 [2] designed to address a key limitation in conventional unsupervised image anomaly localization methods, namely the overgeneralization problem. In traditional approaches based on knowledge distillation, an apprentice network is trained to mimic the expert network by minimizing the discrepancies between corresponding feature descriptors (FDs) extracted from anomaly-free images. Specifically, for a pixel x in an input image, the expert network produces a feature $f_T(x)$ and the apprentice network produces a feature $f_S(x)$; the discrepancy between these features is measured by a function $d(\cdot, \cdot)$, typically defined as the pixel-wise mean square error between the normalized features, i.e.,

$$d(f_T(x), f_S(x)) = \left\| \hat{f}_T(x) - \hat{f}_S(x) \right\|_2^2 \tag{4.5}$$

where $\hat{f}$ denotes the normalized feature vector. The training process in conventional methods is based on minimizing the average discrepancy for normal samples, thereby assuming that the discrepancies for abnormal features would naturally remain large. However, due to the high generalization capacity of the student network, even abnormal inputs may yield features that are close to those of the teacher network, resulting in small discrepancies and leading to high prediction uncertainty. This phenomenon, referred to as the overgeneralization problem, causes the discrepancy distributions of normal and abnormal features to exhibit a small margin and significant overlap, both of which impair the reliability of anomaly localization.

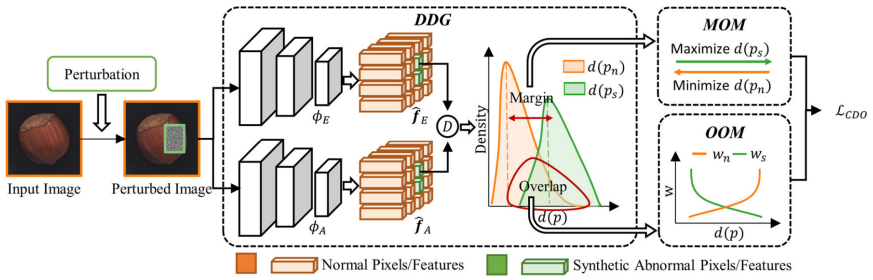


Figure 4.7: CDO loss function to ovoid overgeneralization, introduced by Collaborative Discrepancy Optimization for Reliable Image Anomaly Localization on 2023 [2]

To overcome this shortcoming, CDO introduces a collaborative optimization strategy that leverages both normal data and synthetically generated abnormal data. Synthetic anomalies are produced by applying random perturbations to normal images. Concretely, several square regions within a normal image are randomly selected and their pixel values are replaced by random values sampled

from a Gaussian distribution, thus forming a set of synthetic abnormal pixels as discussed in section 4.1.3. Denoting the set of normal pixels as $\mathcal{N}$ and that of synthetic abnormal pixels as $\mathcal{A}$, the corresponding discrepancy distributions (DDs) can be expressed as $\{d(f_T(x), f_S(x))\}_{x\in\mathcal{N}}$ and $\{d(f_T(x), f_S(x))\}_{x\in\mathcal{A}}$, respectively.

The essence of CDO lies in the simultaneous optimization of these two discrepancy distributions. Unlike previous methods that solely focus on minimizing the discrepancy for normal features, CDO enforces a dual objective: it minimizes the discrepancies of normal FDs while maximizing those of synthetic abnormal FDs. This collaborative optimization directly targets the margin between the two distributions, which can be roughly defined as the absolute difference between their average discrepancy values. Mathematically, the basic formulation of the loss function is given by

$$\mathcal{L} = \sum_{x\in\mathcal{N}} d(f_T(x), f_S(x)) - \sum_{x\in\mathcal{A}} d(f_T(x), f_S(x)) \tag{4.6}$$

This loss function is designed to enlarge the gap (margin) between the average discrepancies of normal and abnormal features, thereby ensuring that the apprentice network not only replicates the expert network's behavior for normal inputs but also produces clearly distinguishable outputs for abnormal inputs.

While optimizing the average discrepancy is beneficial, it does not fully address the influence of tail samples those normal samples with unusually large discrepancies and abnormal samples with unexpectedly small discrepancies that contribute to the overlap between the two distributions. To further reduce this overlap, CDO incorporates a dynamic weighting mechanism inspired by the principles of the Focal Loss. In this scheme, the importance of each pixel is modulated based on the ratio of its discrepancy to the average discrepancy of its corresponding set. Let $\bar{d}_{\mathcal{N}}$ and $\bar{d}_{\mathcal{A}}$ denote the average discrepancies for the normal and abnormal distributions, respectively. The weight for a normal pixel is then defined as

$$w_{\mathcal{N}}(x) = \left(\frac{d(f_T(x), f_S(x))}{\bar{d}_{\mathcal{N}}}\right)^{\gamma} \tag{4.7}$$

and the weight for an abnormal pixel is defined as

$$w_{\mathcal{A}}(x) = \left(\frac{d(f_T(x), f_S(x))}{\bar{d}_{\mathcal{A}}}\right)^{-\gamma} \tag{4.8}$$

where $\gamma \geq 0$ is a hyper-parameter that controls the degree of emphasis on these tail samples. In effect, normal pixels that deviate more from the average (indicating potential outliers even among normal samples) are assigned higher weights, and similarly, abnormal pixels that appear too similar to normal samples (with lower than expected discrepancies) are also given greater importance.

Integrating this dynamic weighting into the loss function, the final CDO loss is formulated as

$$\mathcal{L}_{\mathcal{CDO}} = \frac{\sum_{x\in\mathcal{N}} w_{\mathcal{N}}(x)\, d(f_T(x), f_S(x)) - \sum_{x\in\mathcal{A}} w_{\mathcal{A}}(x)\, d(f_T(x), f_S(x))}{\sum_{x\in\mathcal{N}} w_{\mathcal{N}}(x) + \sum_{x\in\mathcal{A}} w_{\mathcal{A}}(x)} \tag{4.9}$$

This expression ensures that the margin between the normal and abnormal discrepancy distributions is maximized while the overlap is minimized, thereby reducing prediction uncertainty and enhancing anomaly localization performance.

Once the training with CDO is complete, the anomaly score for each pixel is computed by measuring the discrepancy between the normalized features extracted by the expert and apprentice networks. That is, for any pixel x, the anomaly score is given by

$$S(x) = d(f_T(x), f_S(x)) = \left\| \hat{f}_T(x) - \hat{f}_S(x) \right\|_2^2 \tag{4.10}$$

In practical applications, it has been demonstrated that integrating multi-hierarchical representations i.e., combining discrepancy information from various layers of a convolutional neural network can further improve anomaly localization. If discrepancies computed from different hierarchical levels are denoted as $d_h(f_S(x), f_T(x))$ for $h = 1, \ldots, H$, the final anomaly score may be expressed as

$$S(x) = \sum_{h=1}^{H} \alpha_h\, d_h(f_T(x), f_S(x)) \tag{4.11}$$

where α_h are weighting coefficients that balance the contribution of each hierarchical level.

Overall, the proposed CDO [2] framework provides a comprehensive solution to the inherent limitations of previous methods by directly addressing the overgeneralization issue. Through the use of synthetic anomalies and a dual optimization strategy that targets both the margin and the overlap of discrepancy distributions, CDO ensures that the student network produces features that are both accurate for normal inputs and highly sensitive to anomalies. This is achieved without the need for explicitly available abnormal samples during training, as the synthetic abnormalities serve to implicitly guide the network towards a more robust separation between normal and abnormal feature spaces. The resulting improvement in anomaly localization performance is particularly valuable in applications where high reliability and precision are paramount.

4.7 Anomaly Maps Normalization

In our anomaly detection framework, we combine two complementary sources of information a local anomaly map generated by a student model and a global anomaly map produced by an autoencoder to capture both fine-grained and broad structural deviations in images. Since the outputs of these two models naturally have different dynamic ranges and statistical properties, it is essential to bring them to a common scale before fusion. To address this, we implement a robust quantile-based linear normalization strategy suggested in [1], augmented with a tunable hyperparameter α that adjusts the relative influence of the two maps on the final anomaly heatmap which is essential for out project due to flexibiliy needed in industrial environment.

The normalization process begins by analyzing a set of validation images that are known to be defect-free. For each anomaly map, we collect all pixel-wise anomaly scores from these images and compute two key quantiles, q_a and q_b, corresponding to pre-determined probabilities a and b. These quantiles effectively capture the typical distribution of scores in normal conditions and serve as reliable anchors for normalization. A linear transformation is then defined to map q_a to a normalized score of 0 and q_b to 0.1. The selection of 0 and 0.1 is not arbitrary; these values ensure compatibility with standard zero-to-one color scales used for visualization and are robust to variations in the underlying score distribution, whether Gaussian, multimodal, or otherwise. Importantly, this mapping does not impact performance metrics such as the area under the ROC curve (AU-ROC), since these metrics depend solely on the ranking of anomaly scores rather than their absolute magnitudes.

To further enhance the adaptability of our system, we introduce the hyperparameter α, which provides a mechanism to control the balance between the local and global anomaly maps in the final fusion process. If we denote the normalized student output as M_{student} and the normalized autoencoder output as $M_{\text{autoencoder}}$, the final anomaly heatmap H is computed as:

$$H = \alpha \cdot M_{\text{student}} + (1 - \alpha) \cdot M_{\text{autoencoder}} \tag{4.12}$$

with α taking a value in the range $[0, 1]$. A value of α closer to 1 gives more weight to the student model, emphasizing the detection of localized defects a scenario often encountered in high-precision manufacturing where subtle, local irregularities are critical. Conversely, setting α nearer to 0 prioritizes the global perspective provided by the autoencoder, which can be more effective in contexts where broad, structural inconsistencies indicate anomalies. This weighting mechanism allows practitioners to tailor the detection system to the unique requirements of different manufacturing lines or industries, ensuring that the strengths of both models are optimally leveraged.

The use of quantile-based normalization is particularly advantageous because it mitigates the impact of outliers and noise inherent in real-world data. By anchoring the normalization process to the behavior of anomaly scores in defect-free images, the method adapts to varying distributions and preserves the meaningful differences between defect-free and defective regions. Moreover, the fixed mapping targets of 0 and 0.1 not only facilitate a consistent visual interpretation but also ensure that the dynamic ranges of both maps are comparable when they are fused. This uniform scaling is crucial, as an unbalanced fusion could result in one map overshadowing the other, potentially concealing significant defect signals.

Overall, our normalization approach is designed to be both robust and flexible. It addresses the challenge of merging heterogeneous anomaly maps by standardizing their scales through a data-driven, quantile-based linear transformation, and it introduces a customizable hyperparameter α to fine-tune the contribution of each map based on empirical performance and domain-specific considerations. The detailed tuning process for α and further discussions on its impact are provided in Section 5.2. This integrated method ensures that the final combined anomaly heatmap is both accurate and adaptable, making it a key component of our defect detection pipeline in diverse industrial settings.

Chapter 5

Experiments and Results

In this section we would like to discuss about how step by step we study the recent papers related to anomaly detection. Moreover, we will discuss about how each component can effect the results of our industrial thesis. Furthermore, we will explain how we used the details explained in section 4 to achieve best results on our main focused dataset called Plastic-nut from Real-IAD [3] with explaining hardware and software required for implementation.

5.1 Hardware and Software

Our experimental framework is implemented in Python, a language chosen for its versatility and the extensive ecosystem of libraries that support cutting-edge deep learning research. In this section, we provide a comprehensive discussion of the libraries utilized, detailing their roles and the rationale behind their selection.

At the core of our framework lies PyTorch, which serves as the primary deep learning library. PyTorch offers dynamic computation graphs and an intuitive API, enabling rapid prototyping and flexible experimentation. Its efficient handling of tensor operations and automatic differentiation is crucial for training our teacher-student network architecture as well as the associated autoencoder module. Furthermore, PyTorch's native support for CUDA-enabled GPUs allows us to offload computationally intensive tasks such as forward and backward passes and the computation of complex loss functions to high-performance hardware, thereby substantially reducing training time. The modular design of PyTorch also facilitates the seamless integration of custom components, such as our specialized CDO loss function, which performs pixel-wise feature comparisons and adaptive weighting.

Complementing PyTorch is NumPy, a fundamental library for numerical computing in Python. NumPy provides efficient n-dimensional array objects and a comprehensive suite of mathematical functions that are indispensable for data

manipulation and statistical analysis. In our pipeline, NumPy is extensively used for tasks such as image normalization, quantile calculations, and general array transformations, all of which are critical in preprocessing the high-dimensional data and interfacing effectively with PyTorch tensors.

Image processing forms a critical component of our experimental setup, and this is where PIL (Python Imaging Library) and tifffile come into play. PIL is employed for a wide array of image manipulation tasks including loading, format conversion, and the application of transformations such as color jitter, sharpness enhancement, and gamma correction. These operations are essential for augmenting the training dataset, thereby simulating diverse imaging conditions that improve the generalizability of the model. On the other hand, tifffile is specifically used for reading and writing TIFF images, ensuring that the high-resolution anomaly maps generated during evaluation are stored with precision and integrity.

Torchvision further extends the functionalities of PyTorch by providing commonly used image transformations, pre-trained models, and dataset utilities. The transform modules available in Torchvision enable us to perform standard preprocessing tasks such as resizing and normalization, and they are seamlessly integrated with our custom transformation pipeline. This pipeline includes both standard augmentations and more advanced modifications (e.g., random application of color jitter or geometric adjustments), which are critical for the robust training of the student network under various perturbations.

The argparse library is used to manage configuration and hyperparameter settings through command-line arguments. This design choice allows researchers to adjust parameters such as dataset paths, model sizes, training iterations, and loss coefficients without altering the source code. By facilitating easy customization, argparse ensures reproducibility and systematic experimentation, which are central to our research methodology.

Another key utility is provided by the itertools library, which is utilized for generating infinite data iterators and managing complex iteration patterns over training samples. This capability is particularly valuable in our context, as it helps prevent interruptions in the training loop due to dataset size limitations and supports the dynamic creation of synthetic anomalies via custom data augmentation techniques.

Real-time progress tracking during training is achieved using the tqdm library. Tqdm offers a visually intuitive progress bar that provides immediate feedback on the training process, including iteration counts, loss values, and checkpointing events. This feedback mechanism is indispensable during extended training sessions, as it allows for real-time monitoring and prompt debugging if necessary In addition to these core libraries, OpenCV (imported as cv2) is employed within our custom dataset classes to perform advanced image processing tasks. OpenCV's efficient routines for background estimation and noise injection are used to simulate realistic

synthetic anomaly patches. These patches are strategically inserted into the training images to enrich the diversity of the dataset and to challenge the student network with varied anomaly conditions, thus enhancing its robustness.

For evaluation purposes, we integrate metrics computation using Scikit-Learn. Scikit-Learn provides reliable implementations for evaluation metrics such as the ROC-AUC score, which quantifies the classification performance of the model. Its modular design allows for seamless incorporation into our evaluation pipeline, ensuring that the performance metrics are both accurate and reproducible.

On the hardware side, our system is engineered to dynamically detect and utilize CUDA-enabled GPUs through PyTorch's device query functions. When a compatible GPU is available, acceleration is leveraged to significantly enhance performance by parallelizing computations across thousands of cores, making both training and inference substantially faster compared to CPU-only execution. In this project, we employed an NVIDIA RTX 4000 Ada Generation GPU. These specifications provide ample computational power for deep learning tasks, especially when handling high-resolution image data and complex neural network architectures.

To further optimize performance, multi-threading is exploited via PyTorch's DataLoader configuration, which utilizes multiple worker processes and pinned memory. This ensures efficient data transfer between the CPU and GPU, minimizing bottlenecks caused by slow memory access. The system also benefits from mixed-precision training with Tensor Cores, which enhances efficiency by reducing memory usage while maintaining model accuracy. This hybrid approach leveraging both the CPU for preprocessing and the GPU for heavy computations ensures that our framework can effectively manage the high computational demands of deep learning workflows, particularly in scenarios requiring real-time or near-real-time processing.

5.2 Training and Results

In this section, we provide a comprehensive overview of our training procedure for EfficientAD, originally designed for the MVtecAD dataset and subsequently adapted to the more challenging Real-IAD dataset. Specifically, we focus on the plastic-nut subset of Real-IAD, which presents diverse industrial anomalies and demands a more robust detection approach than MVtecAD. Our experiments explore different network architectures (small and medium PDNs), pretraining strategies, loss functions, and augmentation techniques. We evaluate the interplay between **classification** (normal vs. abnormal) and **segmentation** (precise localization of anomalous regions), aiming for a method that excels in both accuracy and efficiency.

5.2.1 Adaptation from MVtecAD to Real-IAD

Originally, EfficientAD [1] was validated on the MVtecAD dataset [17], a widely known benchmark for anomaly detection on industrial objects and textures. However, Real-IAD [3] features significantly larger and more varied data. To ensure compatibility with unsupervised anomaly detection, we restructured the MVtecAD-like splits in Real-IAD to include only normal samples for training, leaving anomalous samples exclusively for validation and testing. This allows the network to learn the normal data distribution more effectively and detect any deviations during inference. Notably, plastic-nut exhibits several unique manufacturing defects and imaging inconsistencies, making it a representative subset of industrial challenges. As a result of modifications in the data structure of Real-IAD to become similar to MVTecAD (training, test and groundtruth which train contain only anomaly free images). in table 5.1 we can see the number of images.

	Abnormal	Normal
Number of Images	1246	2511

Table 5.1: Number of image in plastic-nut dataset after structural modifications

5.2.2 Pretraining and Teacher Networks

Pretraining serves as a foundation for both the medium and small teacher networks, distilled into corresponding student networks medium PDN and small PDN as described in 4.2. The architectural details for these PDNs are listed in Table 5.3 and 5.2. Our approach follows the design principles in [1], which incorporate lightweight yet powerful encoder-decoder structures conducive to real-world industrial settings.

Layer Name	Stride	Kernel Size	Number of Kernels	Padding	Activation
Conv-1	1×1	4×4	128	3	ReLU
AvgPool-1	2×2	2×2	128	1	-
Conv-2	1×1	4×4	256	3	ReLU
AvgPool-2	2×2	2×2	256	1	-
Conv-3	1×1	3×3	256	1	ReLU
Conv-4	1×1	4×4	384	0	-

Table 5.2: Patch description network architecture of the teacher network for EfficientAD-S. The student network has the same architecture, but 768 kernels instead of 384 in the Conv-4 layer. A padding value of 3 means that three rows, or columns respectively, of zeros are appended at each border of an input feature map.[1]

Layer Name	Stride	Kernel Size	Number of Kernels	Padding	Activation
Conv-1	1×1	4×4	256	3	ReLU
AvgPool-1	2×2	2×2	256	1	-
Conv-2	1×1	4×4	512	3	ReLU
AvgPool-2	2×2	2×2	512	1	-
Conv-3	1×1	1×1	512	0	ReLU
Conv-4	1×1	3×3	512	1	ReLU
Conv-5	1×1	4×4	384	0	ReLU
Conv-6	1×1	1×1	384	0	-

Table 5.3: Patch description network architecture of the teacher network for EfficientAD-M. The student network has the same architecture, but 768 kernels instead of 384 in the Conv-5 and Conv-6 layers. A padding value of 3 means that three rows, or columns respectively, of zeros are appended at each border of an input feature map.[1]

Two pretrained teacher networks were used for pretraining. Medium teacher provides higher capacity and more complex feature representations, leading to enhanced detection and localization capabilities. Small teacher offers reduced memory footprint and faster inference. While slightly less accurate than the medium teacher, it remains well-suited for scenarios with stringent hardware constraints. Based on PDN network we use the Algorithm 1 which explained earlier we distill our teacher based on ImageNet and WideResNet-101 into medium or small size of PDN. As a result, we achieve out teacher in order to use for training and next phases of the project.

5.2.3 Autoencoder for Reconstruction-Based Detection

Additionally, we integrated an autoencoder with a dedicated decoder to capture texture and structural deviations. By comparing reconstructed images with the

originals, the autoencoder highlights abnormal regions that deviate from the learned "normal" distribution. The architectures are enumerated in Table 5.4, ensuring they remain computationally efficient while retaining reconstruction accuracy.

Layer Name	Stride	Kernel Size	Number of Kernels	Padding	Activation
EncConv-1	2×2	4×4	32	1	ReLU
EncConv-2	2×2	4×4	32	1	ReLU
EncConv-3	2×2	4×4	64	1	ReLU
EncConv-4	2×2	4×4	64	1	ReLU
EncConv-5	2×2	4×4	64	1	ReLU
EncConv-6	1×1	8×8	64	0	-
Bilinear-1	-	-	-	-	Resizes the 1×1 input feature maps to 3×3
DecConv-1	1×1	4×4	64	2	ReLU
Dropout-1	-	-	-	-	Dropout rate = 0.2
Bilinear-2	-	-	-	-	Resizes the 4×4 input feature maps to 8×8
DecConv-2	1×1	4×4	64	2	ReLU
Dropout-2	-	-	-	-	Dropout rate = 0.2
Bilinear-3	-	-	-	-	Resizes the 9×9 input feature maps to 15×15
DecConv-3	1×1	4×4	64	2	ReLU
Dropout-3	-	-	-	-	Dropout rate = 0.2
Bilinear-4	-	-	-	-	Resizes the 16×16 input feature maps to 32×32
DecConv-4	1×1	4×4	64	2	ReLU
Dropout-4	-	-	-	-	Dropout rate = 0.2
Bilinear-5	-	-	-	-	Resizes the 33×33 input feature maps to 63×63
DecConv-5	1×1	4×4	64	2	ReLU
Dropout-5	-	-	-	-	Dropout rate = 0.2
Bilinear-6	-	-	-	-	Resizes the 64×64 input feature maps to 127×127
DecConv-6	1×1	4×4	64	2	ReLU
Dropout-6	-	-	-	-	Dropout rate = 0.2
Bilinear-7	-	-	-	-	Resizes the 128×128 input feature maps to 64×64
DecConv-7	1×1	3×3	64	1	ReLU
DecConv-8	1×1	3×3	384	1	-

Table 5.4: Network architecture of the autoencoder for EfficientAD-S and EfficientAD-M. Layers named "EncConv" and "DecConv" are standard 2D convolutional layers. [1]

5.2.4 Balancing Loss Components in EfficientAD

A key goal of our study was to investigate the impact of multiple sub-losses within the total loss function introduced by EfficientAD. To this end, we introduced four coefficients to vary the relative importance of each loss component:

- **coeff_Hard**: Scales the hard-distillation loss between the teacher and student networks, encouraging the student to mimic the teacher's classification boundaries.
- **coeff_OOD**: A term intended to enhance robustness to out-of-distribution (OOD) samples, preventing overfitting to seen data distributions.
- **coeff_AE**: Governs the reconstruction-based loss from the autoencoder, aligning the learned representations with the normal reconstruction targets.

- **coeff_STAE**: Combines student-teacher distillation signals with autoencoder reconstruction, fostering a unified representation of classification and segmentation.

We systematically varied these coefficients to observe their influence on classification and localization. Interestingly, we found that introducing $\mathcal{L}_{\text{OOD}}$ reduced detection performance by approximately $n\%$, likely because it over-penalized differences from normal data, even when such differences corresponded to valid anomalies. The must important achievement of this Experiment was to see the performance changing behaviour of model with different influence of ImageNet penalty which results are available in Figure 5.1 shows that we can change this loss function in order to improve model performance.

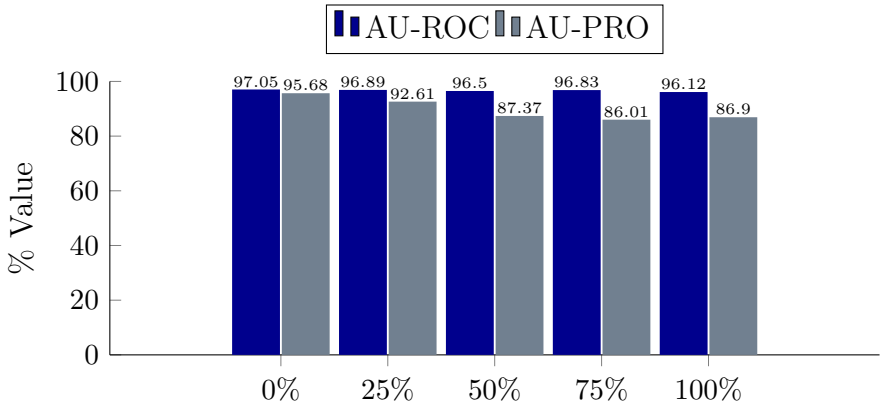


Figure 5.1: Percentage of loss of out of distribution on training.

5.2.5 Training with Limited Data

An industrial environment often restricts the availability of labeled data. To evaluate performance in data-scarce scenarios, we trained our framework on only 500 normal images, while maintaining the full test set (see 5.5). Surprisingly, the model maintained robust classification (normal vs. abnormal) and localization capabilities, underscoring the method's viability in production lines with limited data collection resources.

Train (only normal)	Test	AUROC (%)	AU-PRO (%)
2259	1499	97.85	92.03
500	1499	96.07	95.48
100	1499	92.82	95.67

Table 5.5: Performance metrics with varying training sizes

with considering reduced performance of AU-PRO while having full dataset size we can conclude that we faced with overgeneralization when training the models on more number of data. As a result, still we needed to improve the model to be more robust against overgeneralization which we will discuss about it in Section 5.2.8 in deep.

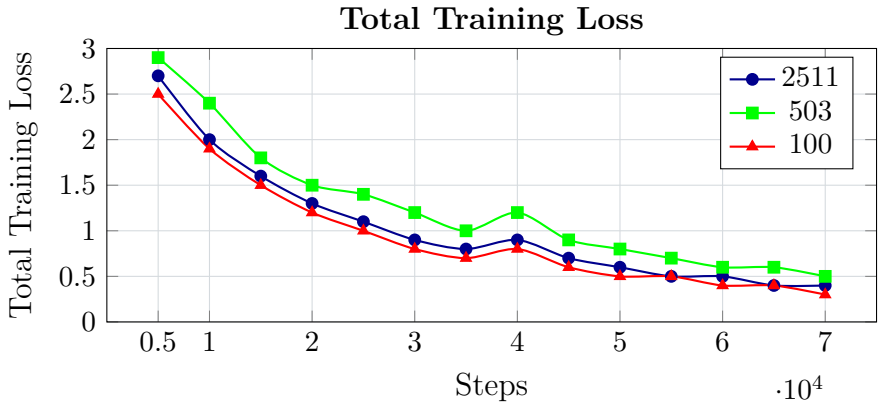


Figure 5.2: Total training loss over training steps for different dataset sizes

5.2.6 Distillation-Only vs. Autoencoder-Only Training

To highlight the complementarity of student-teacher distillation and autoencoder reconstruction, we conducted a controlled experiment using either $\mathcal{L}_{\text{Hard}}$ (student-teacher only) or $\mathcal{L}_{\text{AE}}$ (autoencoder only). As reported in Table 5.6:

Distillation-Only (Student-Teacher) exhibits strong classification performance high AUROC but struggles to precisely localize anomalies lower AU-PRO, losing nearly 10% in segmentation capability.

Autoencoder-Only excels at generating meaningful heat maps for spatial anomalies high AU-PRO, but its global classification ability is comparatively weaker.

This trade-off clearly indicates that distillation-based and reconstruction-based approaches are synergistic. Each method alone addresses only part of the anomaly

Table 5.6: Performance metrics for different methods with and without ImageNet penalty

Method	ImageNet Penalty	AUROC (%)	AU-PRO (%)
Autoencoder	-	91.66	**94.80**1
Student	-	97.60	73.68
Student + Autoencoder	-	**97.85**	92.03
Student + Autoencoder	Yes	96.18	80.86
Student	Yes	95.91	63.18

detection challenge. With combination of both of them we can achieve more Reliable result.

Our main training pipeline till this step needs to combine both student-teacher distillation and autoencoder reconstruction, normalizing and merging each network's anomaly maps to form a unified detector. Illustrative examples in Figure 5.3 reveal that certain samples are better captured by the student's discriminative features, while others benefit from the autoencoder's reconstruction error. Merging these complementary signals yields superior performance across both classification (AUROC) and segmentation (AU-PRO).

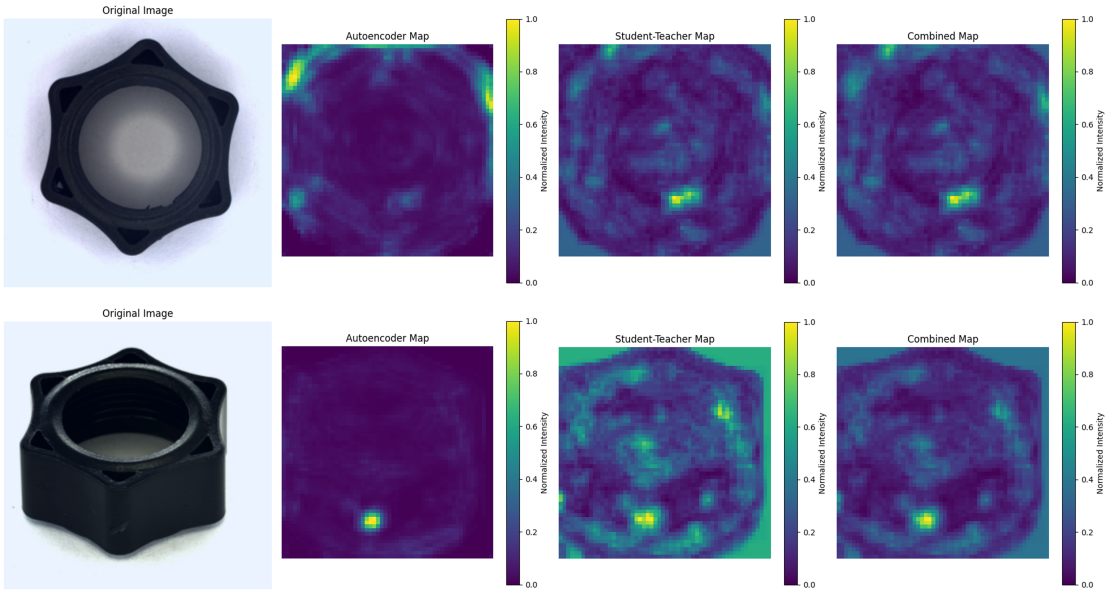


Figure 5.3: Output of models seperately and normalized and combined map.

As industrial point of view, we need more flexibiliy depends on situation which we are interest in classification or segmentation we might need one of the models have more influence on the output result. Furthermore, this output should be flexible depends on environment and type of object. As a result, we add new hyperparameter named α as discussed in Section 4.7 to change our preferences at inference phase.

5.2.7 Performance vs. Model Complexity

We examined both medium PDN and small PDN variants. Although the small PDN is more efficient and suitable for hardware-limited situations, we observed a slight reduction in detection accuracy, particularly in segmentation metrics. Nonetheless, the small PDN remains an attractive option for real-time inspection pipelines where model size and inference speed are critical. In general, the medium PDN offers a better trade-off between detection accuracy and real-time feasibility.

Network Architecture	**AU-PRO**	**Complexity**
SMALL PDN	84.79	1,096,320
MEDIUM PDN	**92.03**	11,615,488

Table 5.7: Performance and complexity of different PDN architectures.

5.2.8 Collaborative Discrepancies Integration

Moreover, we replaced the traditional $\mathcal{L}_{\text{OOD}}$ with a novel $\mathcal{L}_{\text{CDO}}$ [2], intended to mitigate overgeneralization while preserving the model's sensitivity to subtle defects. As shown in Table 5.8, this substitution improved the final detection metrics, particularly in challenging cases. As mentioned in Section 4.1.3 by adding Syntetic anomalies and using Equation 4.9 we added new loss in to the total loss of out training.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Hard}} + \mathcal{L}_{\text{CDO}} + \mathcal{L}_{\text{AE}} + \mathcal{L}_{\text{STAE}} \tag{5.1}$$

Combining perturbation with architecture of reconstruction based and embedding based method make us able to use advantage of each of them. Although implementation of combination was challenging as technical point of view, but we maintain the complexities of the model suitable in compared to initial point and we improved the model ability to localization.

5.2.9 External Validation on Other Datasets

Finally, to ensure our modifications did not overfit to plastic-nut alone, we tested the improved model on additional Real-IAD subsets. The results, summarized in Table 5.8, corroborate that our architecture and training strategies generalize well to other industrial defect types with minimal fine-tuning. Furthermore, Figure 5.4 contain some sample of our model on other subdatasets.

Dataset	AUROC	AU-PRO
Plastic_nut (Real-IAD)	97.07	95.48
Usb (Real-IAD)	97.16	94.11
Metal_nut (MVTec AD)	99.26	94.14
Cable (MVTec AD)	96.92	83.93
Hazelnut (MVTec AD)	96.82	81.80

Table 5.8: Comparison of AUROC and AU-PRO values for different datasets.

Overall, our experiments confirm that distillation and reconstruction are complementary, enabling strong classification (via teacher-student) and precise segmentation (via autoencoder). Smaller architectures maintain competitive performance and are well-suited for hardware-constrained environments, although they may sacrifice some accuracy. Data scarcity is tolerable, as even 500 training images yielded satisfactory detection results. Customized OOD losses and augmentation strategies can significantly affect performance in subtle, real-world industrial scenarios and localization. Thus, our modifiations on EfficientAD framework effectively scales from standard anomaly detection benchmarks to demanding industrial applications, balancing detection accuracy and computational efficiency.

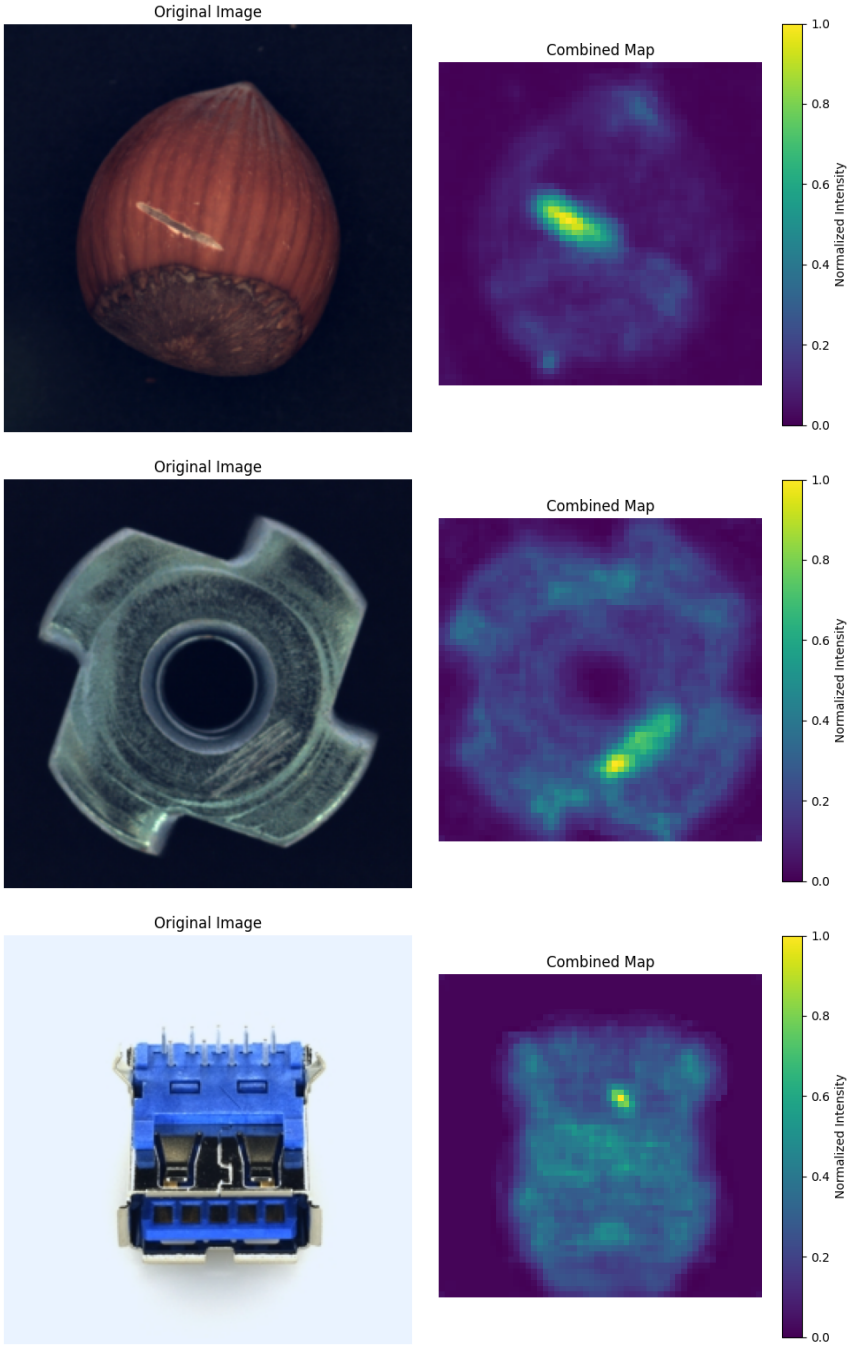


Figure 5.4: Sample output of model on other objects.

Chapter 6

Conclusion and Future works

The work presented in this thesis explored an integrated anomaly detection pipeline designed specifically for complex industrial imaging environments. Our approach combined a teacher-student framework with an autoencoder-based methodology to not only detect but also precisely localize defects in high-resolution images. By leveraging robust data augmentation strategies and sophisticated loss functions such as Collaborative Discrepancy Optimization (CDO), our system demonstrated impressive resilience against variations in lighting conditions, background clutter, and the inherent diversity of defect types. Experimental results on the Plastic-Nut subset of the Real-IAD dataset revealed that our hybrid pipeline substantially reduces both false positives and false negatives when compared to more conventional single-model approaches, confirming the value of integrating multiple detection cues into one unified framework.

A pivotal aspect of our approach has been the incorporation of image tiling techniques within the anomaly detection process. Industrial imaging tasks frequently involve extremely high-resolution images, where anomalies such as minute scratches, pits, or contaminations can be localized to only a few pixels. Processing an entire high-resolution image at once often forces a reduction in image size, which can lead to the loss of critical local details. To address this challenge, our pipeline employs an image tiling strategy that divides each high-resolution image into a series of smaller, manageable patches or tiles. This division allows the model to maintain fine-grained resolution and concentrate on localized regions with greater precision. Each tile is processed independently, ensuring that the local texture and structural nuances are captured accurately by the model. Moreover, by analyzing these smaller sections in detail, the system is better equipped to identify subtle anomalies that might otherwise be overlooked if the image were downsampled or

processed as a whole.

The image tiling technique also offers significant computational benefits. When images are partitioned into tiles, the model can operate on smaller chunks of data that require less memory, enabling the deployment of more complex deep learning architectures even on hardware with limited computational resources. This approach is particularly beneficial in industrial settings, where real-time or near-real-time analysis is essential. The tiling process inherently allows for parallel processing; multiple tiles can be analyzed simultaneously, which further enhances the system's throughput and reduces overall processing time. In addition, tiling facilitates more effective use of spatial context, as overlapping tiles can be utilized to ensure that features located near the edges of one tile are not lost or misinterpreted. This overlapping mechanism helps in creating a seamless fusion of local predictions into a coherent global anomaly map that preserves the spatial integrity of the original image.

Despite its clear advantages, the image tiling technique also introduces several challenges that open up interesting avenues for future research. One such challenge is determining the optimal size and number of tiles for different types of industrial images. The ideal tile size must strike a balance between maintaining sufficient local detail and ensuring that the model retains enough contextual information from the overall image. In scenarios where defects are exceedingly small, smaller tiles may be necessary, but they risk losing the broader contextual cues that can help in distinguishing true anomalies from benign variations. Conversely, larger tiles might capture more context but at the expense of local resolution. Future work could investigate adaptive tiling strategies, where the tile size is dynamically determined based on the content of the image or guided by preliminary assessments of the image's complexity.

Another promising direction lies in refining the fusion strategies used to combine the predictions from individual tiles into a final, coherent anomaly map. Presently, our approach involves a relatively straightforward aggregation of tile-level outputs. However, more sophisticated fusion methods could account for the varying confidence levels across tiles, incorporate edge-aware blending techniques, or even leverage attention mechanisms to weigh the contributions of different regions more effectively. Such enhancements could further reduce false positives at tile boundaries and improve the overall robustness of the anomaly detection system.

Moreover, integrating the image tiling strategy with other components of our pipeline, such as the autoencoder and the teacher-student framework, could unlock additional performance improvements. For instance, the autoencoder's ability to reconstruct images can be optimized by focusing on individual tiles, thereby enhancing its capacity to capture localized reconstruction errors that are indicative of anomalies. Similarly, the teacher-student model can be modified to operate on a tile-by-tile basis, where the discrepancies between the student and teacher

outputs are computed for each tile independently. This approach would enable the detection mechanism to be highly sensitive to small, localized defects while still leveraging the global context provided by the complete image.

the model again during pretraining to enhance its performance. DINO, a self-supervised learning approach based on Vision Transformers In future work, we propose replacing ResNet101 with DINO [63] in our anomaly detection framework and distilling (ViTs), has demonstrated superior feature extraction capabilities without requiring labeled data, making it well-suited for unsupervised anomaly detection. By leveraging DINO, we can improve the model's ability to learn meaningful representations that capture both local and global structures within images, potentially leading to better anomaly detection performance. Additionally, after pretraining the teacher network using DINO, we can further distill its knowledge into a student network, similar to our existing approach but with a more powerful and generalizable feature extractor. This modification may enhance the student model's capacity to detect subtle anomalies, improve generalization across different datasets, and reduce dependence on large-scale labeled data. Future research should focus on evaluating the effectiveness of DINO-based teacher-student distillation in anomaly detection, assessing computational costs, and comparing performance with traditional CNN-based architectures.

In summary, the integration of image tiling techniques into our anomaly detection pipeline has proven to be a powerful strategy for addressing the challenges posed by high-resolution industrial imaging. By preserving local detail and enabling efficient processing, tiling allows the model to detect even the smallest defects with high precision while managing computational resources effectively. Future work that builds on these insights—by refining tiling strategies, improving fusion methodologies, and integrating multi-scale analysis—holds great promise for developing a truly universal anomaly detection framework at the same time by using new backbone we can experience new result. Such a framework would be capable of delivering robust, low-latency, and highly accurate performance across a broad spectrum of industrial use cases, thereby contributing significantly to advancements in quality control and automated inspection systems.

Bibliography

[1] Kilian Batzner, Lars Heckler, and Rebecca König. *EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies*. 2024. arXiv: 2303.14535 [cs.CV]. URL: https://arxiv.org/abs/2303.14535 (cit. on pp. ii, 32, 48, 49, 51, 55, 57, 62, 67–69).

[2] Yunkang Cao, Xiaohao Xu, Zhaoge Liu, and Weiming Shen. «Collaborative Discrepancy Optimization for Reliable Image Anomaly Localization». In: *IEEE Transactions on Industrial Informatics* 19.11 (2023), pp. 10674–10683. DOI: 10.1109/TII.2023.3241579 (cit. on pp. ii, 32, 59, 61, 73).

[3] Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jianning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. *Real-IAD: A Real-World Multi-View Dataset for Benchmarking Versatile Industrial Anomaly Detection*. 2024. arXiv: 2403.12580 [cs.CV]. URL: https://arxiv.org/abs/2403.12580 (cit. on pp. ii, 26, 33, 35, 38–40, 64, 67).

[4] Neelam Dahiya, Sheifali Gupta, and Sartajvir Singh. «A Review Paper on Machine Learning Applications, Advantages, and Techniques». In: *ECS Transactions* 107.1 (Apr. 2022), p. 6137. DOI: 10.1149/10701.6137ecst. URL: https://dx.doi.org/10.1149/10701.6137ecst (cit. on pp. 4, 5).

[5] K O'Shea. «An introduction to convolutional neural networks». In: *arXiv preprint arXiv:1511.08458* (2015) (cit. on p. 4).

[6] Jacob Devlin. «Bert: Pre-training of deep bidirectional transformers for language understanding». In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on p. 4).

[7] Alec Radford. «Improving language understanding by generative pre-training». In: (2018) (cit. on p. 4).

[8] Alex Sherstinsky. «Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network». In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306 (cit. on p. 5).

[9] David G Lowe. «Distinctive image features from scale-invariant keypoints». In: *International journal of computer vision* 60 (2004), pp. 91–110 (cit. on pp. 7, 48).

[10] Navneet Dalal and Bill Triggs. «Histograms of oriented gradients for human detection». In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893 (cit. on p. 7).

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep residual learning for image recognition». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 8).

[12] Mingxing Tan and Quoc Le. «Efficientnet: Rethinking model scaling for convolutional neural networks». In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114 (cit. on p. 8).

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. «Swin transformer: Hierarchical vision transformer using shifted windows». In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022 (cit. on pp. 8, 30).

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. «Imagenet: A large-scale hierarchical image database». In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on pp. 9, 51).

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. «Imagenet classification with deep convolutional neural networks». In: *Advances in neural information processing systems* 25 (2012) (cit. on p. 9).

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. «Reducing the dimensionality of data with neural networks». In: *science* 313.5786 (2006), pp. 504–507 (cit. on p. 14).

[17] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. «MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9592–9600 (cit. on pp. 26, 34, 35, 39, 67).

[18] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. «The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization». In: *arXiv preprint arXiv:2112.09045* (2021) (cit. on pp. 26, 34, 57).

[19] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. «Spot-the-difference self-supervised pre-training for anomaly detection and segmentation». In: *European Conference on Computer Vision*. Springer. 2022, pp. 392–408 (cit. on pp. 26, 34, 39).

[20] Jhih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. «Learning unsupervised metaformer for anomaly detection». In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4369–4378 (cit. on p. 27).

[21] Shelly Sheynin, Sagie Benaim, and Lior Wolf. «A hierarchical transformation-discriminating generative model for few shot anomaly detection». In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 8495–8504 (cit. on p. 27).

[22] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. «Registration based few-shot anomaly detection». In: *European Conference on Computer Vision*. Springer. 2022, pp. 303–319 (cit. on p. 27).

[23] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. «Winclip: Zero-/few-shot anomaly classification and segmentation». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19606–19616 (cit. on p. 27).

[24] Yunkang Cao, Xiaohao Xu, Chen Sun, Yuqi Cheng, Zongwei Du, Liang Gao, and Weiming Shen. «Segment any anomaly without training via hybrid prompt regularization». In: *arXiv preprint arXiv:2305.10724* (2023) (cit. on p. 27).

[25] Daniel Stanley Tan, Yi-Chun Chen, Trista Pei-Chun Chen, and Wei-Chao Chen. «Trustmae: A noise-resilient defect classification framework using memory-augmented auto-encoders with trust regions». In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021, pp. 276–285 (cit. on p. 27).

[26] Chen Qiu, Aodong Li, Marius Kloft, Maja Rudolph, and Stephan Mandt. «Latent outlier exposure for anomaly detection with contaminated data». In: *International conference on machine learning*. PMLR. 2022, pp. 18153–18167 (cit. on p. 27).

[27] Antoine Cordier, Benjamin Missaoui, and Pierre Gutierrez. «Data refinement for fully unsupervised visual inspection using pre-trained networks». In: *arXiv preprint arXiv:2202.12759* (2022) (cit. on p. 27).

[28] Paul Bergmann and David Sattlegger. «Anomaly detection in 3d point clouds using deep geometric descriptors». In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 2613–2623 (cit. on p. 27).

[29] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. «Asymmetric student-teacher networks for industrial anomaly detection». In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2023, pp. 2592–2602 (cit. on pp. 27, 30).

[30] Eliahu Horwitz and Yedid Hoshen. «Back to the feature: classical 3d features are (almost) all you need for 3d anomaly detection». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2968–2977 (cit. on p. 27).

[31] Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. «A diffusion-based framework for multi-class anomaly detection». In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 8. 2024, pp. 8472–8480 (cit. on p. 27).

[32] Ruitao Chen, Guoyang Xie, Jiaqi Liu, Jinbao Wang, Ziqi Luo, Jinfan Wang, and Feng Zheng. «Easynet: An easy network for 3d industrial anomaly detection». In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 7038–7046 (cit. on p. 27).

[33] Jiaqi Liu, Guoyang Xie, Ruitao Chen, Xinpeng Li, Jinbao Wang, Yong Liu, Chengjie Wang, and Feng Zheng. «Real3d-ad: A dataset of point cloud anomaly detection». In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on pp. 27, 34).

[34] Shuanlong Niu, Bin Li, Xinggang Wang, and Hui Lin. «Defect image sample generation with GAN for improving defect recognition». In: *IEEE Transactions on Automation Science and Engineering* 17.3 (2020), pp. 1611–1622 (cit. on p. 27).

[35] Andrei-Timotei Ardelean and Tim Weyrich. «High-fidelity zero-shot texture anomaly localization using feature correspondence analysis». In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 1134–1144 (cit. on p. 27).

[36] Taoran Wei, Danhua Cao, Caiyun Zheng, and Qun Yang. «A simulation-based few samples learning method for surface defect segmentation». In: *Neurocomputing* 412 (2020), pp. 461–476 (cit. on p. 27).

[37] Wujin Li, Jiawei Zhan, Jinbao Wang, Bizhong Xia, Bin-Bin Gao, Jun Liu, Chengjie Wang, and Feng Zheng. «Towards continual adaptation in industrial anomaly detection». In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 2871–2880 (cit. on p. 27).

[38] Ying Zhao. «Omnial: A unified cnn framework for unsupervised anomaly localization». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3924–3933 (cit. on p. 27).

[39] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. «Padim: a patch distribution modeling framework for anomaly detection and localization». In: *International Conference on Pattern Recognition*. Springer. 2021, pp. 475–489 (cit. on p. 29).

[40] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. «Cutpaste: Self-supervised learning for anomaly detection and localization». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 9664–9674 (cit. on p. 29).

[41] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. «Towards total recall in industrial anomaly detection». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 14318–14328 (cit. on pp. 29, 31, 51).

[42] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. «Ganomaly: Semi-supervised anomaly detection via adversarial training». In: *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer. 2019, pp. 622–637 (cit. on p. 29).

[43] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. «Unsupervised anomaly detection with generative adversarial networks to guide marker discovery». In: *International conference on information processing in medical imaging*. Springer. 2017, pp. 146–157 (cit. on p. 29).

[44] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. «Dsr–a dual subspace re-projection network for surface anomaly detection». In: *European conference on computer vision*. Springer. 2022, pp. 539–554 (cit. on p. 29).

[45] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. «Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection». In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1705–1714 (cit. on p. 29).

[46] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. «Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4183–4192 (cit. on pp. 30, 31).

[47] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. «Multiresolution knowledge distillation for anomaly detection». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 14902–14912 (cit. on p. 30).

[48] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. «Student-teacher feature pyramid matching for anomaly detection». In: *arXiv preprint arXiv:2103.04257* (2021) (cit. on p. 30).

[49] Hanqiu Deng and Xingyu Li. «Anomaly detection via reverse distillation from one-class embedding». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 9737–9746 (cit. on pp. 30, 31).

[50] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. «Simplenet: A simple network for image anomaly detection and localization». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 20402–20411 (cit. on p. 30).

[51] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. «Segmentation-based deep-learning approach for surface-defect detection». In: *Journal of Intelligent Manufacturing* 31.3 (2020), pp. 759–776 (cit. on p. 34).

[52] Yibin Huang, Congying Qiu, and Kui Yuan. «Surface defect saliency of magnetic tile». In: *The Visual Computer* 36.1 (2020), pp. 85–96 (cit. on p. 34).

[53] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. «Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions». In: *2021 13th International congress on ultra modern telecommunications and control systems and workshops (ICUMT)*. IEEE. 2021, pp. 66–71 (cit. on p. 34).

[54] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. «VT-ADL: A vision transformer network for image anomaly detection and localization». In: *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. IEEE. 2021, pp. 01–06 (cit. on p. 34).

[55] Qiang Zhou, Weize Li, Lihan Jiang, Guoliang Wang, Guyue Zhou, Shanghang Zhang, and Hao Zhao. «Pad: A dataset and benchmark for pose-agnostic anomaly detection». In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on p. 34).

[56] Luca Bonfiglioli, Marco Toschi, Davide Silvestri, Nicola Fioraio, and Daniele De Gregorio. «The eyecandies dataset for unsupervised multimodal anomaly detection and localization». In: *Proceedings of the Asian Conference on Computer Vision*. 2022, pp. 3586–3602 (cit. on p. 34).

[57] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. «ORB: An efficient alternative to SIFT or SURF». In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571 (cit. on p. 48).

[58] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. «Image patch matching using convolutional descriptors with euclidean distance». In: *Asian Conference on Computer Vision*. Springer. 2016, pp. 638–653 (cit. on p. 48).

[59] Xiong You, Qin Li, Ke Li, Anzhu Yu, and Shuhui Bu. «PL-VSCN: Patch-level vision similarity compares network for image matching». In: *IET Computer Vision* 15.2 (2021), pp. 122–135 (cit. on p. 48).

[60] Dosovitskiy Alexey. «An image is worth 16x16 words: Transformers for image recognition at scale». In: *arXiv preprint arXiv: 2010.11929* (2020) (cit. on p. 48).

[61] Sergey Zagoruyko. «Wide residual networks». In: *arXiv preprint arXiv:1605.07146* (2016) (cit. on p. 51).

[62] Alexey Dosovitskiy and Thomas Brox. «Generating images with perceptual similarity metrics based on deep networks». In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 57).

[63] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: `2104.14294 [cs.CV]`. URL: `https://arxiv.org/abs/2104.14294` (cit. on p. 78).