Data Synthesis and Stability-Guided Training for Industrial Visual Inspection with Neural Networks

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

Muen Jin, M.Sc.

geb. in Jiangsu, VR China

Tag der mündlichen Prüfung:	2. Oktober 2026
Hauptreferent:	Prof. Dr.-Ing. Michael Heizmann, KIT
Korreferent:	Prof. Dr.-Ing. Marco Huber, Uni Stuttgart



This document—excluding parts marked otherwise, the cover, pictures and graphs—is licensed under a Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0): https://creativecommons.org/licenses/by-sa/4.0/deed.en

Preface

Completing this thesis marks the end of an important stage of my academic journey. Looking back on the years leading to this point, I realize how strongly this journey has been shaped by the people I had the privilege to learn from, work with, and rely on along the way.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr.-Ing. Michael Heizmann. His guidance and support have accompanied me not only throughout my doctoral studies, but already during my Master's studies, when his lectures and teaching had a lasting influence on the way I approached scientific questions and research. Over the years, I have benefited greatly from his broad knowledge, thoughtful advice, and willingness to discuss both fundamental and practical aspects of my work.

I would also like to thank Prof. Dr.-Ing. Marco Huber for serving as co-supervisor of this thesis and for the many constructive comments and suggestions that helped improve its quality.

My sincere thanks also go to all my colleagues at IIIT for the many discussions, exchanges, and moments of support over the past years. The open and collaborative environment at the institute contributed greatly to making this period both productive and enjoyable. In particular, I would like to thank Abdullah Al-Hammadi, Carsten Schmerbeck, Daniel Diaz Ocampo, Daniel Leyer, Fabian Leven, Luisa Hoffmann, Johannes Baßler, Johannes Steffens, Manuel Bihler, Simon Wezstein, and Thijs van Wijk for carefully proofreading parts of this thesis and for their helpful comments.

I am also grateful to Prof. Dr.-Ing. Sander Wahls for his willingness to support the review of my thesis at short notice if needed. I would further like to thank Manuela Moritz and Patricia Nestl for their continuous administrative support and for helping with the many organizational matters that accompany several years of research work. I am equally grateful to Dieter Brandt and Marvin Winkler for their technical support

and for their assistance with hardware, infrastructure, and the practical aspects of experimental work.

I thank the German Federal Ministry for Economic Affairs and Climate Action (Bundesministerium für Wirtschaft und Klimaschutz, BMWK) for financially supporting the first two years of my doctoral research through the cooperative project "Hybride Bildverarbeitung auf FPGAs" within the Central Innovation Programme for SMEs (Zentrales Innovationsprogramm Mittelstand, ZIM). I am also grateful to the state of Baden-Württemberg for providing access to the bwHPC computing infrastructure.

Above all, I would like to thank my family. I am deeply grateful to my parents, my grandparents, and all other members of my family who have supported me throughout these years. A doctoral journey is a long one, and it inevitably comes with periods of uncertainty and self-doubt. Their encouragement, patience, and trust gave me the strength to continue when progress seemed slow and the destination still distant.

Looking back, I have always considered myself fortunate. At every stage of my life, I have encountered people who offered me help, friends whose paths resonated with my own, and people whose example inspired me to keep moving forward. They shape how my journey of life is experienced and remembered. For all of these encounters, I am sincerely grateful.

Karlsruhe, October 2026 Muen Jin

Declarations

- Generative artificial intelligence tools in the form of large language models were employed for proofreading, improving the style of the manuscript, and writing programming code. All text generated using LLMs was manually reviewed and often further revised. The models were not used to generate new content.
- The author states no conflict of interest.
- This work was partially funded by the German Federal Ministry for Economic Affairs and Climate Action (Bundesministerium für Wirtschaft und Klimaschutz, BMWK) within the Central Innovation Programme for SMEs (Zentrales Innovationsprogramm Mittelstand, ZIM) as part of the cooperative project “Hybride Bildverarbeitung auf FPGAs” (project funding reference: KK5143401LF0).

Abstract / Kurzfassung

Abstract

In this dissertation, synthetic image data for industrial visual inspection are investigated in combination with refinement and stability-guided training strategies for deep learning models. The work addresses three closely related challenges: the limited availability of representative defect data, the domain gap between synthetic and real images, and the difficulty of reliable model selection in validation-free training settings.

First, synthetic datasets are generated for two representative inspection scenarios. For plastic granules, a modeling-based rendering pipeline is developed to create annotated images under different acquisition conditions. For metallic parts, prompt-guided image generation is used to synthesize realistic crack patterns, followed by alignment and annotation. These datasets provide controllable training data for classification, segmentation, and anomaly detection.

Second, synthetic-to-real refinement is investigated in order to improve the visual alignment of synthetic images with real inspection data while preserving task-relevant structures. Neural style transfer is adapted to the considered industrial scenarios and compared with GAN-based image translation methods. The results show that refinement quality is determined not only by fidelity to the real domain but also by the variety retained in the refined data, and that these two aspects must be balanced with respect to the downstream task.

Third, a stability-guided early stopping strategy is developed for encoder-based anomaly detection models trained without labeled validation data. Instead of relying on validation performance or test-based checkpoint selection, the proposed method monitors the temporal consistency of learned feature embeddings during training and identifies suitable checkpoints from the training process itself.

The proposed methods are evaluated across classification, segmentation, and anomaly detection tasks. The results show that refined synthetic data can improve downstream performance when their integration into the training process is carefully controlled. They also show that stability-guided checkpoint selection improves the robustness of model training, particularly in settings with domain shift or without reliable validation data.

Overall, this dissertation demonstrates that synthetic data generation, feature-consistent refinement, and stability-aware training can be combined into a coherent methodology for industrial visual inspection with deep learning. The presented approach reduces the dependence on real defect data and supports more reliable model deployment under practical constraints.

Kurzfassung

In dieser Dissertation werden synthetische Bilddaten für die industrielle visuelle Inspektion in Kombination mit Verfeinerungs- und stabilitätsgeführten Trainingsstrategien für Deep-Learning-Modelle untersucht. Die Arbeit adressiert drei eng miteinander verknüpfte Herausforderungen: die begrenzte Verfügbarkeit repräsentativer Defektdaten, den Domänenunterschied zwischen synthetischen und realen Bildern sowie die Schwierigkeit einer zuverlässigen Modellauswahl in validierungsfreien Trainingsszenarien.

Zunächst werden synthetische Datensätze für zwei repräsentative Inspektionsszenarien erzeugt. Für Kunststoffgranulat wird eine modellierungsbasierte Rendering-Pipeline entwickelt, um annotierte Bilder unter unterschiedlichen Aufnahmebedingungen zu generieren. Für metallische Bauteile wird promptbasierte Bildgenerierung eingesetzt, um realistische Rissmuster zu synthetisieren. Anschließend erfolgen Ausrichtung und Annotation. Diese Datensätze stellen kontrollierbare Trainingsdaten für Klassifikation, Segmentierung und Anomalieerkennung bereit.

Anschließend wird die Synthetic-to-Real-Verfeinerung untersucht, um die visuelle Übereinstimmung synthetischer Bilder mit realen Inspektionsdaten zu verbessern und gleichzeitig aufgabenrelevante Strukturen zu erhalten. Dazu wird Neural Style Transfer an die betrachteten

industriellen Szenarien angepasst und mit GAN-basierten Bildübersetzungsverfahren verglichen. Die Ergebnisse zeigen, dass die Qualität der Verfeinerung nicht nur durch die Fidelity zur realen Domäne bestimmt wird, sondern auch durch die in den verfeinerten Daten erhaltene Varietät, und dass beide Aspekte in Abhängigkeit von der jeweiligen nachgelagerten Aufgabe ausbalanciert werden müssen.

Darüber hinaus wird eine stabilitätsgeführte Early-Stopping-Strategie für encoderbasierte Anomalieerkennungsmodelle entwickelt, die ohne gelabelte Validierungsdaten trainiert werden. Anstatt auf Validierungsleistung oder testdatenbasierte Checkpoint-Auswahl zurückzugreifen, überwacht die vorgeschlagene Methode die zeitliche Konsistenz gelernter Merkmalsrepräsentationen während des Trainings und identifiziert geeignete Checkpoints direkt aus dem Trainingsverlauf.

Die vorgeschlagenen Methoden werden anhand von Klassifikations-, Segmentierungs- und Anomalieerkennungsaufgaben ausgewertet. Die Ergebnisse zeigen, dass verfeinerte synthetische Daten die Leistung nachgelagerter Modelle verbessern können, wenn ihre Einbindung in den Trainingsprozess sorgfältig gesteuert wird. Darüber hinaus zeigt sich, dass die stabilitätsgeführte Checkpoint-Auswahl die Robustheit des Modelltrainings verbessert, insbesondere in Szenarien mit Domänenverschiebung oder ohne verlässliche Validierungsdaten.

Insgesamt zeigt diese Dissertation, dass sich synthetische Datengenerierung, merkmalskonsistente Verfeinerung und stabilitätsbewusstes Training zu einer kohärenten Methodik für die industrielle visuelle Inspektion mit Deep Learning verbinden lassen. Der vorgestellte Ansatz verringert die Abhängigkeit von realen Defektdaten und unterstützt eine zuverlässigere Modellanwendung unter praktischen Randbedingungen.

Contents

Preface i

Abstract / Kurzfassung v

Nomenclature xiii

1 Introduction 1
1.1 Contributions of this Dissertation 3
1.2 Outline of the Dissertation 4

2 Related Works 7
2.1 Tasks in Industrial Visual Inspection 7
2.1.1 Classification 8
2.1.2 Segmentation 9
2.1.3 Anomaly Detection 11
2.1.4 Other Tasks 13
2.2 Neural Network Architectures for Visual Inspection . . . 14
2.2.1 Artificial Neural Networks 14
2.2.2 Convolutional Neural Networks 17
2.2.3 Transformer-Based Vision Models 20
2.3 Synthetic Data Generation for Industrial Visual Inspection 24
2.3.1 Modeling-Based Image Generation 24
2.3.2 Learning-Based Image Generation 26
2.4 Synthetic-to-Real Domain Gap and Image Refinement . . 30
2.4.1 Domain Gap and Refinement Criteria 31
2.4.2 GAN-Based Image Refinement 33
2.4.3 Neural Style Transfer for Image Refinement 34
2.4.4 Diffusion-Based Image Refinement 37

2.5 Training Stability and Model Selection under Limited Data . . . 38
2.5.1 Overfitting and Training Instability in Deep Models . . . 38
2.5.2 Early Stopping and Regularization Strategies . . . 40
2.5.3 Model Selection without Validation Data . . . 42

3 Image Synthesis for Materials and Defects . . . 45
3.1 Introduction . . . 45
3.2 Real Datasets . . . 47
3.3 Scene Rendering Based on 3D Modeling . . . 48
3.3.1 Geometry Modeling . . . 49
3.3.2 Material Assignment . . . 51
3.3.3 Environment Modeling . . . 52
3.3.4 Scene Composition and Annotation . . . 53
3.3.5 Results . . . 55
3.4 Prompt-Guided Image Generation via LLVMs . . . 56
3.4.1 Model Benchmarking and Selection . . . 57
3.4.2 Batch Generation and Annotation . . . 61
3.4.3 Results . . . 64

4 Feature-Consistent Synthetic Image Refinement . . . 67
4.1 Method . . . 68
4.1.1 Refinement of Synthetic Granule Images . . . 69
4.1.2 Refinement of Synthetic Crack Images . . . 71
4.2 Evaluation of Refined Images . . . 77
4.2.1 Evaluation Setup . . . 78
4.2.2 Evaluation of Fidelity . . . 80
4.3 Conclusion . . . 90

5 Training Stability and Early Stopping for Industrial Visual Anomaly Detection Models . . . 91
5.1 Introduction . . . 91
5.2 Limitations of State-of-the-Art Methods . . . 92
5.3 Embedding-Stability-Guided Early Stopping . . . 94
5.3.1 Embedding Extraction . . . 95
5.3.2 Stability Divergence of Embeddings . . . 96

5.3.3 Adaptive Threshold for Convergence Detection . 97
5.4 Integration into Anomaly Detection Frameworks 98
5.5 Experimental Setup 100
5.5.1 Datasets 101
5.5.2 Frameworks and Training Protocol 101
5.5.3 Early Stopping Hyperparameters 102
5.5.4 Evaluation Metrics 103
5.6 Experimental Results 103
5.6.1 Anomaly Detection Performance on Industrial Benchmarks 103
5.6.2 Analysis of Hyperparameter Choices 114
5.7 Conclusion 117

6 Assessing Synthetic Data and Model Stability Across Downstream Tasks 119
6.1 Introduction 119
6.2 Classification on Synthetic and Real Granule Images . . . 120
6.2.1 Task Description and Experimental Setup 121
6.2.2 Experimental Results 122
6.2.3 Summary of Classification Tasks 128
6.3 Crack Segmentation with Synthetic Supervision 130
6.3.1 Task Description and Experimental Setup 131
6.3.2 Experimental Results 132
6.4 Anomaly Detection Across Synthetic and Real Domains . 141
6.4.1 Task Description and Experimental Setup 143
6.4.2 Anomaly Detection on Granule Datasets 144
6.4.3 Anomaly Detection on Crack Dataset 159
6.4.4 Conclusion 162

7 Summary and Outlook 165
7.1 Summary of Contributions 165
7.2 Limitations 167
7.3 Future Work 168

A Evaluation Metrics 173
A.1 Distribution-Level Image Similarity Metrics 173
A.1.1 Fréchet Inception Distance 174

A.1.2 Maximum Mean Discrepancy 175
A.2 Pairwise Image Similarity Metrics 176
A.2.1 Mean Squared Error 177
A.2.2 Peak Signal-to-Noise Ratio 177
A.2.3 Structural Similarity Index Measure 177
A.2.4 Learned Perceptual Image Patch Similarity 178
A.3 Metrics for Binary Classification 179
A.3.1 Accuracy 180
A.3.2 Precision 180
A.3.3 Recall 181
A.3.4 F1 Score 181
A.4 Metrics for Binary Segmentation 181
A.4.1 Foreground Intersection over Union 182
A.4.2 Background Intersection over Union 182
A.4.3 Mean Intersection over Union 183
A.5 Metrics for Anomaly Detection 183
A.5.1 Receiver Operating Characteristic Curve 184
A.5.2 Image AUROC 184
A.5.3 Pixel AUROC 185

Bibliography 187
List of publications 216
List of supervised theses 217

Nomenclature

Common abbreviations

Symbols

Mathematical operators

1 Introduction

Industrial visual inspection plays a central role in modern manufacturing and quality assurance, enabling automated assessment of product integrity, defect detection, and process monitoring. Typical inspection tasks include classification of raw materials, segmentation of defect regions, and detection of abnormal samples that deviate from nominal appearance. With increasing production speed and complexity, manual inspection becomes infeasible, motivating the adoption of computer vision–based solutions [49, 57, 172, 216].

In recent years, deep learning has become a dominant paradigm for visual inspection, driven by its strong representation capacity and superior performance across a wide range of vision tasks [118, 119, 159, 263]. In particular, convolutional neural networks and their variants have demonstrated strong effectiveness in learning discriminative features directly from image data, enabling robust inspection models for diverse industrial applications [117, 222, 281, 286, 304]. However, despite these advances, deploying deep learning–based inspection systems in real-world industrial environments remains challenging.

A fundamental difficulty arises from the limited availability of labeled defect data. In many industrial settings, defects occur rarely, are highly diverse, or evolve over time, making comprehensive data collection and annotation prohibitively expensive [5, 103, 197, 289, 298, 313]. While supervised learning benefits strongly from large and representative datasets, industrial inspection often operates under severe data scarcity, particularly for defective samples. This limitation has motivated the exploration of alternative learning paradigms, such as anomaly detection trained primarily on normal data [33, 35, 110], as well as data augmentation [186, 247, 271, 293] and synthetic data generation techniques [1, 71, 124, 166, 207].

Synthetic data generation offers a way to alleviate data scarcity by providing scalable and fully annotated training samples. Simulation-based

rendering pipelines and generative models have been used to create synthetic images that mimic real inspection scenarios, enabling controlled variation of geometry, appearance, and defect characteristics [93, 184, 207, 269]. Nevertheless, synthetic data typically exhibit a domain gap with respect to real images, stemming from imperfect modeling of imaging conditions, material properties, and noise characteristics. This domain gap often limits the direct transferability of models trained on synthetic data to real inspection settings.

To reduce this gap, synthetic-to-real refinement methods have been proposed, aiming to align the appearance statistics of synthetic images with those of real data. Approaches based on generative adversarial networks, neural style transfer, or related image translation techniques have been used to improve visual realism and domain alignment [43, 194, 278, 292]. However, refinement introduces its own challenges: changes in appearance may alter task-relevant features, affect data diversity, or interact with training dynamics in unintended ways. As a result, refinement does not universally guarantee improved downstream performance and must be applied with care.

Beyond data-related issues, training stability and model selection present additional obstacles in industrial visual inspection. Deep learning models, particularly when trained on limited or synthetic data, can exhibit unstable optimization behavior, overfitting, or sensitivity to hyperparameters [15, 250, 267, 270]. Common practices such as selecting the best-performing checkpoint based on validation performance often rely on assumptions that are unrealistic in industrial scenarios, especially for anomaly detection tasks where labeled anomalies are unavailable. These issues highlight that robust inspection performance depends not only on data quality, but also on principled training control and deployable model selection strategies.

Taken together, industrial visual inspection with deep learning is constrained by a combination of data scarcity, domain shift, and training instability. Addressing these challenges requires a holistic perspective that considers data generation, data refinement, and training dynamics as interconnected components rather than isolated problems. This dissertation is situated within this context and aims to develop and analyze such an integrated methodology for industrial visual inspection.

1.1 Contributions of this Dissertation

Existing research on deep learning–based industrial visual inspection typically addresses individual aspects of the inspection pipeline in isolation, such as data generation, domain adaptation, or model training. In contrast, this dissertation adopts a holistic perspective in which synthetic data generation, synthetic-to-real refinement, and training control are treated as interdependent components. The main contributions of this work are summarized as follows:

- Systematic synthesis of industrial inspection data with controllable structure and annotation: This dissertation develops multiple synthetic data generation pipelines tailored to industrial visual inspection, including physically motivated rendering-based synthesis and prompt-guided image generation using large language vision models. These pipelines enable explicit control over geometry, material appearance, environmental conditions, and can provide precise annotations including pixel-level labels where required. Unlike ad hoc data augmentation, the generated datasets serve as controlled experimental variables for studying domain shift and training behavior across classification, segmentation, and anomaly detection tasks.

- Feature-consistent refinement of synthetic images with explicit analysis of fidelity and variety: A feature-consistent synthetic image refinement framework based on neural style transfer is introduced and adapted to industrial inspection data. The refinement strategy is designed to preserve task-relevant structural information while aligning appearance statistics with real images. Through systematic evaluation, the dissertation identifies a clear trade-off between refinement fidelity and data variety and contrasts the behavior of neural style transfer–based and GAN-based refinement approaches, providing a comparative analysis of their respective strengths and limitations.

- Stability-guided early-stopping based on embedding dynamics: The dissertation proposes a validation-free early-stopping strategy that monitors the temporal stability of learned feature embeddings

during training. By detecting convergence through embedding consistency rather than validation performance, the method addresses the practical constraints of industrial inspection scenarios where labeled defect data or validation sets are unavailable. The approach introduces only limited computational overhead and is compatible with existing deep learning frameworks.

- Generalization of stability-guided training control across inspection tasks: While stability-guided early stopping is initially motivated by anomaly detection, this dissertation demonstrates that the proposed criterion generalizes to supervised inspection tasks, including image segmentation and classification. The results show that stability-based checkpoint selection improves robustness against overfitting and training instability in the presence of synthetic and refined synthetic data, thereby extending its applicability beyond unsupervised anomaly detection.
- Comprehensive evaluation of synthetic data, refinement strategies, and training stability in industrial visual inspection: An extensive experimental study is conducted across multiple datasets, model architectures, and inspection tasks. The evaluation systematically investigates the interaction between synthetic data composition, refinement strategies, and training control mechanisms, and assesses their impact on deployable performance under realistic industrial constraints. This integrated analysis provides practical guidance for designing robust inspection pipelines based on synthetic data and deep learning.

1.2 Outline of the Dissertation

The remainder of this dissertation is structured as follows.

Chapter 2 reviews the state of the art in industrial visual inspection and its methodological foundations. It introduces the primary inspection tasks, including classification, segmentation, and anomaly detection, followed by an overview of neural network architectures relevant to visual inspection, ranging from basic feedforward networks and convolutional

neural networks to transformer-based models. The chapter further discusses modeling-based and learning-based image generation techniques, as well as synthetic image refinement strategies with emphasis on domain gap, fidelity, and variety. Finally, it analyzes challenges related to overfitting, training instability, regularization, and validation-free model selection in data-limited industrial settings, thereby establishing the theoretical background for the subsequent chapters.

Chapter 3 introduces the synthetic data generation pipelines developed in this dissertation. Physically motivated rendering-based synthesis and prompt-guided image generation using large language vision models are presented, along with the design choices that enable controllable variation and precise annotation for industrial inspection tasks.

Chapter 4 investigates synthetic-to-real refinement strategies for industrial inspection data. Feature-consistent neural style transfer–based refinement methods are introduced and compared to alternative refinement paradigms. The chapter analyzes refinement quality in terms of fidelity, consistency, and variety, and examines how these factors influence downstream inspection performance.

Chapter 5 addresses training stability and model selection under realistic industrial constraints. A stability-guided early-stopping criterion based on embedding dynamics is proposed, motivated by the limitations of conventional validation-based checkpoint selection. The chapter details the formulation of the stability metric and its integration into training workflows.

Chapter 6 evaluates the proposed methodology across multiple industrial visual inspection tasks, including classification, segmentation, and anomaly detection. The experimental analysis systematically studies the interaction between synthetic data composition, refinement strategies, and stability-guided training, and assesses their impact on deployable inspection performance under different acquisition conditions.

Finally, Chapter 7 summarizes the contributions of the dissertation, discusses its limitations, and outlines directions for future research.

2 Related Works

This chapter reviews prior work that forms the conceptual and methodological foundation of this dissertation. Section 2.1 introduces the main task formulations in industrial visual inspection, namely classification, segmentation, and anomaly detection, and discusses their characteristics and practical constraints. Section 2.2 provides an overview of neural network architectures commonly used for visual inspection, with a focus on convolutional and transformer-based models. Section 2.3 surveys approaches for generating synthetic image data for industrial inspection, including modeling-based, learning-based, and augmentation-driven methods. The problem of domain shift between synthetic and real images, along with existing image refinement strategies, is discussed in Section 2.4. Finally, Section 2.5 reviews prior work on training stability and model selection under limited or biased data, motivating the need for alternative training control strategies in industrial inspection scenarios. The discussion in this chapter establishes the conceptual and methodological background for the contributions presented in Chapters 3 to 6.

2.1 Tasks in Industrial Visual Inspection

Industrial visual inspection encompasses a range of task formulations that differ in supervision requirements, output granularity, and practical deployment constraints. This section reviews the three task types most commonly addressed with deep learning in industrial inspection—classification, segmentation, and anomaly detection—and highlights their respective characteristics and typical application scenarios.

2.1.1 Classification

Classification is one of the most fundamental tasks in industrial visual inspection. Given an input image depicting an object or scene, the objective is to assign a discrete label indicating the corresponding class, such as material type, product category, or quality state. Typical industrial applications include sorting of manufactured parts, inspection of surface conditions, and quality grading in high-throughput production lines.

Formally, let $\mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ denote the input image space and let $\mathcal{Y} = \{1, \dots, K\}$ denote a finite set of K class labels. A classification model with trainable parameter θ is defined as a mapping

$$F_\theta : \mathcal{X} \to \mathbb{R}^K, \tag{2.1}$$

which assigns to each input image $\mathbf{X} \in \mathcal{X}$ a class-probability vector $\hat{\mathbf{y}} = F_\theta(\mathbf{X})$, where $\hat{\mathbf{y}} \in [0,1]^K$ and $\sum_{j=1}^{K} \hat{y}_j = 1$. Here, $\hat{y}_j$ denotes the j-th component of the vector $\hat{\mathbf{y}}$. The predicted class label is obtained as

$$\hat{y} = \arg \max_{j=1,\dots,K} \hat{y}_j . \tag{2.2}$$

Training is performed using a labeled dataset $\mathcal{D} = \{(\mathbf{X}^i, y^i)\}_{i=1}^N$. The ground-truth label y^i can be represented equivalently as a one-hot encoded vector $\mathbf{y}^i \in \{0,1\}^K$. The categorical cross-entropy loss for classification is defined as

$$\mathcal{L}_{\mathrm{CCE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} (\mathbf{y}^i)_j \log(\hat{\mathbf{y}}^i)_j . \tag{2.3}$$

In industrial inspection scenarios, classification is often performed under constraints that distinguish it from generic image classification. First, the number of defect samples is typically small compared to the number of normal samples [18, 113, 114, 171, 261, 262, 316], resulting in class imbalance and increased risk of overfitting. Second, unknown or previously unseen defect types may appear during deployment [18, 274, 279, 316]. In such cases, these samples are sometimes assigned to an additional "unknown" or "defective" class, or handled through rejection mechanisms rather than standard multi-class classification.

Moreover, industrial classification tasks frequently assume controlled acquisition conditions, such as fixed viewpoints, homogeneous backgrounds, or centered objects [18, 102, 104, 113, 126, 241, 262, 316, 327]. While these assumptions simplify the classification problem, variations in illumination, surface texture, or production batches can still induce significant domain shifts. As a result, the robustness of classification models with respect to data variability and distribution changes is a central concern in industrial applications.

Overall, image classification serves as a foundational task in industrial visual inspection, providing coarse but efficient decision-making. Its formulation and limitations motivate the exploration of complementary task settings, such as segmentation and anomaly detection, which offer finer-grained or more flexible inspection capabilities.

2.1.2 Segmentation

Segmentation is another central task in industrial visual inspection when spatially resolved information about objects or defects is required. In contrast to classification, which assigns a single label to an entire image, segmentation predicts labels at a finer spatial resolution, typically at the pixel level. This enables precise localization of objects, defect regions, or background areas. Hence, segmentation is essential for applications such as defect localization, dimensional measurement, and downstream actuation.

Let $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ denote an input image. In semantic segmentation, the model produces a class-probability tensor

$$F_\theta(\mathbf{X}) \in [0,1]^{H \times W \times K}.$$

For each pixel location (u, v), the corresponding class-probability vector is

$$\hat{\mathbf{y}}_{u,v} = F_\theta(\mathbf{X})_{u,v}, \qquad \hat{\mathbf{y}}_{u,v} \in [0,1]^K, \tag{2.4}$$

where $\sum_{j=1}^{K} (\hat{\mathbf{y}}_{u,v})_j = 1$ for all (u, v). The predicted dense label map $\hat{\mathbf{Y}} \in \mathcal{Y}^{H \times W}$ is obtained by assigning to each pixel

$$\hat{Y}_{u,v} = \arg \max_{j=1,\dots,K} (\hat{\mathbf{y}}_{u,v})_j. \tag{2.5}$$

Training segmentation models requires annotated data $\mathcal{D} = \{(\mathbf{X}^i, \mathbf{Y}^i)\}_{i=1}^N$, where $\mathbf{Y}^i \in \mathcal{Y}^{H \times W}$ denotes the ground-truth segmentation mask. For the categorical cross-entropy formulation, the ground-truth labels are equivalently represented in one-hot encoded form as $\mathbf{y}_{u,v}^i \in \{0,1\}^K$ at each pixel location (u, v). Thus, the pixel-wise categorical cross-entropy loss is defined as

$$\mathcal{L}_{\text{CCE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{u=1}^{H} \sum_{v=1}^{W} \sum_{j=1}^{K} (\mathbf{y}_{u,v}^i)_j \log(\hat{\mathbf{y}}_{u,v}^i)_j \,, \tag{2.6}$$

where $(\mathbf{y}_{u,v}^i)_j$ denotes the j-th component of the ground-truth class vector at pixel location (u, v) of sample i.

However, in many industrial scenarios this loss alone is insufficient, particularly when target regions are small or unevenly distributed. To address class imbalance, overlap-based loss functions are commonly employed. For binary segmentation, the Dice loss over the dataset is defined as

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \sum_{i=1}^{N} \sum_{u,v} (\hat{\mathbf{y}}_{u,v}^i)_1 \, (\mathbf{y}_{u,v}^i)_1}{\sum_{i=1}^{N} \sum_{u,v} (\hat{\mathbf{y}}_{u,v}^i)_1 + \sum_{i=1}^{N} \sum_{u,v} (\mathbf{y}_{u,v}^i)_1} \,, \tag{2.7}$$

where the formulation is shown for the binary case with foreground class index 1. Dice-based losses emphasize region overlap and are particularly effective when foreground regions are sparse.

Another commonly used objective is the Focal loss [152], which down-weights well-classified pixels and focuses learning on hard examples. For binary segmentation, it is defined as

$$\mathcal{L}_{\text{Focal}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{u,v} \alpha (1 - (\hat{\mathbf{y}}_{u,v}^i)_1)^\gamma \log\left((\hat{\mathbf{y}}_{u,v}^i)_1\right) , \tag{2.8}$$

where $\alpha \in [0, 1]$ is a weighting factor and $\gamma \geq 0$ controls the degree of focusing on misclassified pixels.

In industrial inspection, the characteristics of segmentation tasks depend strongly on the application objective. When segmentation is used for defect localization, such as detecting surface anomalies [3, 12, 31, 60, 115, 261], the target regions often occupy only a small fraction of the

image, leading to severe class imbalance and increased sensitivity to annotation noise. In contrast, when segmentation is applied to localize objects or regions of interest, *e.g.*, to localize individual components in a scene [7, 92, 244, 315, 317], the imbalance issue is less pronounced, but precise boundary delineation and instance separation become critical. Beyond semantic segmentation, instance segmentation is also relevant in industrial inspection scenarios where multiple object instances of the same class must be distinguished. In this setting, the output includes both pixel-wise class assignments and instance identifiers, allowing individual objects to be localized and separated even when they belong to the same semantic category. Instance segmentation is commonly applied in tasks such as object counting [51, 128, 129], precise object localization for pick-and-place systems [40, 90, 273, 296], or high-speed sorting in cluttered scenes [36, 86, 213, 214].

Overall, segmentation provides fine-grained spatial information that is indispensable for many industrial inspection tasks. At the same time, its annotation cost, sensitivity to data variability, and task-dependent challenges motivate the exploration of data-efficient training strategies and robust model selection mechanisms, which are addressed in later chapters of this dissertation.

2.1.3 Anomaly Detection

Anomaly detection is a widely studied task in industrial visual inspection that addresses scenarios in which defective samples are rare, diverse, or unavailable during training. In the literature, this problem is also referred to as one-class classification or novelty detection, reflecting the fact that models are typically trained using data from a single, nominal class and are required to identify deviations from this learned notion of normality at test time.

In this dissertation, anomaly detection is considered under a specific and practically relevant setting: only normal images are available during training, no labeled validation anomalies are assumed, and the test set contains labeled samples from both the normal and the defective classes. This setting reflects common industrial inspection scenarios, where defects occur infrequently and collecting representative anomaly labels for

validation or tuning is impractical, and it is also adopted by widely used benchmark datasets [18, 316, 327].

Formally, let $\mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ denote the input image space and let $\mathcal{Y} = \{0, 1\}$ denote the label space, where 0 represents normal samples and 1 denotes anomalous samples. In the anomaly detection setting considered here, the training data consist exclusively of normal samples, i.e., labeled image pairs $(\mathbf{X}^i, y^i)$ with $y^i = 0$. At test time, both normal and anomalous samples may be encountered.

During inference, an anomaly detection model assigns an anomaly score $s(\mathbf{X}) \in \mathbb{R}$ to an input image $\mathbf{X}$, where larger values indicate a higher likelihood of abnormality. Depending on the inspection objective, anomaly scores can be defined at different output granularities. In image-level anomaly detection, a single scalar score is computed for the entire image. A common formulation is

$$s(\mathbf{X}) = g_{\text{img}}(F_\theta(\mathbf{X})) \ , \tag{2.9}$$

where $F_\theta(\mathbf{X})$ denotes a learned feature representation or reconstruction of the input image and $g_{\text{img}}(\cdot)$ is a scoring function, such as a reconstruction error or a negative log-likelihood under a learned normal data distribution. A binary decision is obtained by thresholding

$$\hat{y} = \begin{cases} 1 & s(\mathbf{X}) > \tau, \\ 0 & \text{otherwise}, \end{cases} \tag{2.10}$$

where $\hat{y} = 0$ denotes a normal image and $\hat{y} = 1$ denotes an anomalous image.

In pixel-level anomaly detection, the model produces a spatial anomaly score map

$$\mathbf{S}(\mathbf{X}) = g_{\text{px}}(F_\theta(\mathbf{X})) \ , \qquad \mathbf{S}(\mathbf{X}) \in \mathbb{R}^{H \times W} , \tag{2.11}$$

where each element $\mathbf{S}_{u,v}(\mathbf{X})$ represents the anomaly score at pixel location (u, v). An image-level score can be derived from the pixel-level representation via an aggregation operator $\phi(\cdot)$, such as the maximum or average:

$$s(\mathbf{X}) = \phi(\mathbf{S}(\mathbf{X})) \ . \tag{2.12}$$

This formulation enables both image-level decision-making and spatial localization of anomalous regions without requiring pixel-wise anomaly labels during training.

A key challenge in anomaly detection under the considered setting is model selection and training control. Since labeled anomalies are unavailable during training and validation, conventional validation-based hyperparameter tuning and early stopping are not directly applicable [30, 64, 73, 85, 168, 174, 192]. As a result, anomaly detection models are particularly sensitive to overfitting and unstable training dynamics, especially when synthetic or augmented data are introduced [198, 288, 306]. These challenges motivate the development of alternative, validation-free training control strategies, which are addressed in later chapters of this dissertation.

2.1.4 Other Tasks

In addition to classification, segmentation, and anomaly detection, several other task formulations are encountered in industrial visual inspection. Common examples include:

- object detection, which localizes objects or defects using bounding boxes and is often applied in scenarios where approximate localization is sufficient, such as coarse defect screening or part presence verification [47, 58, 98, 149, 170, 219, 225, 291, 295, 321].
- instance counting, where the objective is to estimate the number of objects or defects in a scene, *e.g.*, in bulk material inspection or throughput monitoring [2, 48, 142, 290].
- pose estimation, which aims to determine the spatial orientation of objects and is commonly used in pick-and-place systems, robotic assembly, or bin-picking applications [94, 138, 158, 225, 283, 301, 303].
- change detection, which compares images acquired at different times to identify deviations, and is relevant for monitoring process drift, wear, or gradual degradation [13, 165, 173, 201, 232].

- image registration, which aligns images obtained under different viewpoints, times, or sensing conditions and is often used as a preprocessing step in multi-sensor inspection or longitudinal analysis [9, 34, 83, 84, 175, 237].

A detailed introduction is not provided for these tasks, as they are beyond the scope of the work presented in this dissertation.

2.2 Neural Network Architectures for Visual Inspection

Neural networks form the foundation of modern deep learning–based visual inspection systems. This section introduces neural network models from first principles, beginning with basic artificial neural networks and their training, and then focusing on architectural variants that are particularly suited for visual data. Convolutional neural networks are discussed as a dominant architecture for industrial visual inspection, followed by a brief overview of transformer-based vision models and their relevance in generative and representation learning contexts.

2.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are a class of parameterized models inspired by biological neural systems and form the basis of modern deep learning. From a functional perspective, an artificial neural network (ANN) represents a parameterized mapping

$$F_\theta : \mathcal{X} \to \mathcal{Y}\,, \tag{2.13}$$

where $\mathcal{X}$ denotes the input space, $\mathcal{Y}$ the output space and θ the trainable parameters of the network F. In visual inspection tasks, F maps images or image-derived representations to class labels, segmentation masks, or anomaly scores, depending on the specific objective as introduced in Section 2.1.

The basic computational unit of an ANN is the artificial neuron, also referred to as a perceptron, which is illustrated in Fig. 2.1. Given an input

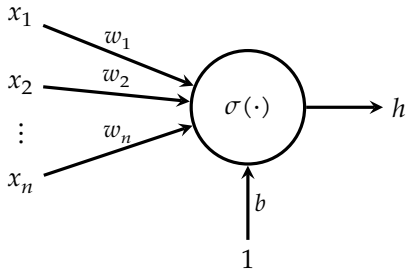


Figure 2.1 A perceptron.

vector $\mathbf{x} \in \mathbb{R}^d$, a perceptron computes a weighted sum followed by a non-linear activation:

$$h = \sigma\big(\mathbf{w}^\top \mathbf{x} + b\big)\,, \tag{2.14}$$

where $\mathbf{w} \in \mathbb{R}^d$ denotes the weight vector, $b \in \mathbb{R}$ is a bias term, and $\sigma(\cdot)$ is a non-linear activation function. Common choices for $\sigma(\cdot)$ include the sigmoid, hyperbolic tangent, and rectified linear unit (ReLU). The activation introduces non-linearity, enabling neural networks to model complex, non-linear relationships.

A multi-layer perceptron (MLP) is constructed by stacking multiple layers of perceptrons. For an MLP with L layers, the forward pass is given by

$$\mathbf{h}^{(1)} = \sigma\Big(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\Big)\,, \quad \mathbf{h}^{(l)} = \sigma\Big(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\Big)\,, \quad l = 2, \dots, L\,, \tag{2.15}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ denote the weights and biases of layer l, and $\sigma(\cdot)$ is a non-linear activation function (assuming the same activation function is applied to all layers). The output of the network is given by $F_\theta(\mathbf{x}) = \mathbf{h}^{(L)}$. By composing linear transformations with non-linear activations, MLPs can approximate a wide range of functions, a property often referred to as the universal approximation capability.

The parameters θ of an ANN are learned from data by minimizing a task-dependent loss function that quantifies the discrepancy between model predictions and ground-truth targets. Given a training dataset

$\mathcal{D} = \{(\mathbf{X}^i, \mathbf{y}^i)\}_{i=1}^N$, the learning objective is commonly formulated as empirical risk minimization:

$$\min_\theta \ \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell\big(F_\theta(\mathbf{X}^i), \mathbf{y}^i\big) \ , \tag{2.16}$$

where $\ell(\cdot, \cdot)$ denotes a sample-wise loss function, such as cross-entropy loss or Dice loss [179] as introduced in Section 2.1.

Optimization is typically performed using gradient-based methods, which iteratively update the parameters in the direction that locally reduces the loss. At each iteration step t, the gradient of the loss with respect to the parameters,

$$\nabla_\theta \mathcal{L} \, , \tag{2.17}$$

is computed at $\theta^{(t)}$, and the parameters are updated according to

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \, , \tag{2.18}$$

where $\eta > 0$ denotes the learning rate controlling the step size of the update. The gradients are efficiently computed using the backpropagation algorithm, which applies the chain rule of calculus to propagate error signals from the network output backward through successive layers.

In practice, the optimization is carried out using stochastic or mini-batch approximations of the loss, where gradients are estimated from small subsets of the training data rather than the full dataset [25, 26, 61, 224]. This reduces computational cost and enables scalable training on large datasets. Variants of gradient descent further introduce mechanisms such as momentum [190, 211, 259], adaptive learning rates [67, 135, 221], or regularization [95, 141, 164, 191, 256] to improve convergence speed and stability. Despite these variations, all such methods follow the same fundamental principle of iteratively adjusting parameters to minimize the empirical loss.

While MLPs provide a general and flexible modeling framework, their direct application to image data is limited. Flattening images into vectors discards spatial structure, and the number of parameters grows rapidly with input resolution, leading to high computational cost and increased risk of overfitting [20, 72, 146, 148]. These limitations motivate the development of architectures that explicitly exploit the spatial properties

of visual data, most notably convolutional neural networks, which are discussed in the following subsection.

2.2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a specialized class of artificial neural networks designed to process data with a grid-like structure, most notably images. Unlike multi-layer perceptrons, which treat inputs as unordered vectors, CNNs preserve the spatial structure of images throughout the network. This is achieved through the use of convolutional layers with local receptive fields and shared parameters, which introduce strong inductive biases tailored to visual data.

2.2.2.1 Convolutional Layers

The core building block of a CNN is the convolutional layer. Instead of applying a dense linear transformation to the entire input, a convolutional layer applies a set of learnable filters, referred to as convolutional kernels, to local regions of the input feature map. Each kernel is a small tensor (*e.g.*, 3×3) that slides across the spatial dimensions of the input and computes a weighted sum of the values within its receptive field. Concretely, given an input feature map $\mathbf{h}^{(l-1)} \in \mathbb{R}^{H \times W \times C}$, a convolutional layer computes the output feature map $\mathbf{h}^{(l)}$ as

$$\mathbf{h}^{(l)}_{u,v,k} = \sigma\left(\sum_{c=1}^{C} \sum_{i,j} \mathbf{W}^{(l)}_{i,j,c,k} \mathbf{h}^{(l-1)}_{u+i,v+j,c} + \mathbf{b}^{(l)}_{k} \right), \tag{2.19}$$

where u and v denote spatial coordinates of the output feature map, i and j index the spatial extent of the convolutional kernel, c indexes the input channels, and k indexes the output channels. The tensor $\mathbf{W}^{(l)}$ represents the learnable convolutional kernels connecting input channel c to output channel k, and $\mathbf{b}^{(l)}$ denotes the bias vector applied per output channel. The bias term is normally added to each output channel to allow the activation thresholds to be adjusted independently of the input signal. This increases the flexibility of the learned representations and helps compensate for systematic intensity offsets or background variations. Fig. 2.2 shows the mechanism of the convolution operation. Non-linear

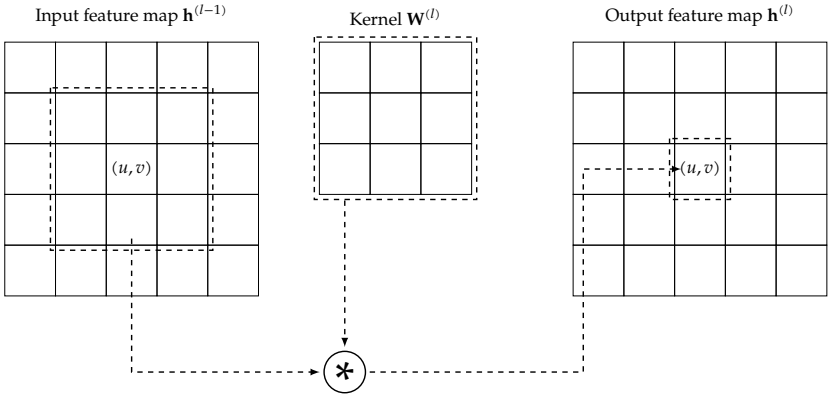


Figure 2.2 Visualization of the convolution mechanism in a CNN. A local region of the input is combined with a kernel to produce a value at the corresponding position in the output feature map. The channel dimensions, bias addition, and activation function are omitted for clarity.

activation functions are applied after convolution to enable the network to model complex, non-linear relationships.

The convolution operation introduces local connectivity, *i.e.*, each output activation depends only on a local neighborhood of the input rather than on all input elements. At the same time, the same kernel parameters are reused at every spatial location, normally referred to as weight sharing. Together, local connectivity and weight sharing reduce the number of learnable parameters compared to fully connected layers, while enforcing sensitivity to spatial patterns that may appear at different positions in the image. By stacking multiple convolutional layers, CNNs form hierarchical feature representations. Shallow layers in CNNs capture simple local structures such as edges or corners, while deeper layers encode increasingly abstract and task-relevant patterns. This hierarchical organization is a key factor in the effectiveness of CNNs for processing data with spatial structure [147, 307].

2.2.2.2 Sampling Layers

Pooling operations are commonly used as means of down-sampling to reduce the spatial resolution of feature maps while retaining the most salient information. Typical pooling mechanisms include max pooling and average pooling [146, 148, 223], which aggregate global feature map values or feature map values within local neighborhoods (*e.g.*, 3×3) by selecting the maximum or computing the mean, respectively. Pooling increases the effective receptive field of subsequent layers, improves robustness to small spatial variations, and reduces computational complexity.

In tasks that require dense predictions, downsampling operations must be complemented by upsampling mechanisms to recover spatial resolution. Common approaches include interpolation-based upsampling and transposed convolution, also referred to as deconvolution [163, 308]. Transposed convolution applies learnable filters to increase spatial resolution and is widely used in encoder–decoder architectures, *e.g.*, U-Net [227], where fine-grained spatial information is progressively reconstructed from compressed feature representations.

2.2.2.3 Normalization and Regularization

To stabilize training and improve convergence, CNN architectures frequently incorporate normalization and regularization techniques. The most commonly adopted normalization for CNNs is batch normalization [122]. It stabilizes training by normalizing intermediate feature activations using statistics computed over each mini-batch. For each feature channel, the activations are centered and scaled according to the corresponding batch mean and variance, followed by a learnable affine transformation that restores representational flexibility. This procedure reduces internal covariate shift, improves gradient flow, and enables the use of higher learning rates during optimization. A small constant is typically added to the variance term to ensure numerical stability. The learnable scaling and shifting parameters allow the network to recover the original representational capacity if normalization is not beneficial for a particular feature channel. Ioffe and Szegedy [122], Santurkar *et al*. [239]

and Bjorck *et al*. [21] show that batch normalization improves gradient flow and contributes to more stable optimization.

Regularization aims to reduce overfitting by discouraging overly complex models. Dropout [106, 256] randomly deactivates a subset of neurons during training, preventing co-adaptation of features and encouraging redundancy in the learned representations. Weight decay [19, 141] introduces a penalty on large parameter values by adding an ℓ_2 regularization term to the loss function, promoting smoother and more generalizable solutions. These techniques are particularly important in industrial inspection scenarios, where labeled training data are often limited.

CNNs can be composed into different architectural patterns depending on the inspection task. For image classification, convolutional layers are typically followed by global pooling and a classification head [140, 151]. For segmentation, encoder–decoder architectures combine progressive downsampling with upsampling stages to recover spatial resolution [163, 227]. In anomaly detection, CNNs are often used as feature extractors [6, 18, 228], where intermediate feature maps or embeddings serve as the basis for anomaly scoring rather than direct prediction.

2.2.3 Transformer-Based Vision Models

Transformer-based vision models [66, 89, 137, 161, 189, 196, 220, 275] constitute an alternative family of neural network architectures that replaces convolutional operations with attention mechanisms, enabling content-adaptive aggregation of information across an input. Originally developed for sequence modeling [272], transformers have been successfully transferred to vision tasks, where they serve as the backbone of both discriminative models, such as vision transformers for segmentation [41, 74, 97, 137, 220, 257, 320], and generative models used for image synthesis [11, 69, 196, 203, 210, 265, 297]. In this dissertation, transformer-based models are primarily relevant as enabling components for synthetic image generation and preprocessing steps, rather than as the main backbone for downstream industrial inspection models.

2.2.3.1 Self-Attention Mechanism

The core building block of transformer-based models is the self-attention mechanism, which enables the model to dynamically weight and aggregate information across different parts of the input. Unlike convolutional layers, which operate on fixed local neighborhoods, self-attention allows each element of the input to directly interact with all others, providing a global receptive field from the first layer.

Let an input sequence be represented by a matrix

$$\mathbf{X} \in \mathbb{R}^{N \times d},$$

where each row of $\mathbf{X}$ corresponds to a sequence element.

Self-attention linearly projects the input into three intermediate representations corresponding to queries, keys, and values:

$$\mathbf{H}^Q = \mathbf{X}\mathbf{W}^Q, \qquad \mathbf{H}^K = \mathbf{X}\mathbf{W}^K, \qquad \mathbf{H}^V = \mathbf{X}\mathbf{W}^V, \tag{2.20}$$

where $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}^V \in \mathbb{R}^{d \times d_v}$ are learnable projection matrices. This yields $\mathbf{H}^Q, \mathbf{H}^K \in \mathbb{R}^{N \times d_k}$ and $\mathbf{H}^V \in \mathbb{R}^{N \times d_v}$.

The attention weights are computed using scaled dot-product attention,

$$\text{Attention}(\mathbf{X}) = \text{softmax}\left(\frac{\mathbf{H}^Q(\mathbf{H}^K)^\top}{\sqrt{d_k}}\right)\mathbf{H}^V, \tag{2.21}$$

where the softmax is applied row-wise over the N tokens and the scaling factor $\sqrt{d_k}$ improves numerical stability for large feature dimensions. The resulting output is a weighted combination of value vectors in $\mathbb{R}^{N \times d_v}$, where the weights depend on the similarity between query and key representations. This formulation allows the model to adaptively emphasize relevant tokens based on their content. The self-attention mechanism is illustrated in Fig. 2.3.

In practice, transformers employ multi-head attention [272], which applies multiple attention operations in parallel using different learned projections. The outputs of these attention heads are then concatenated and linearly transformed. Multi-head attention enables the model to capture different types of relationships simultaneously, such as local and global dependencies, without altering the fundamental attention formulation.

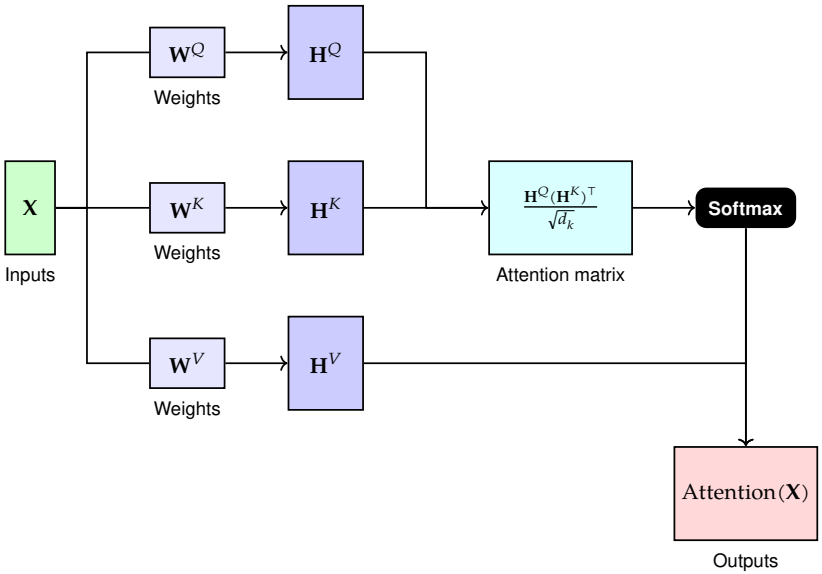


Figure 2.3 Illustration of the self-attention mechanism with query, key, and value projections and scaled dot-product attention.

The self-attention mechanism provides a flexible and expressive means of modeling long-range interactions. However, its computational complexity scales quadratically with the number of tokens, which can be challenging for high-resolution visual data [39, 59, 66, 176, 268]. This characteristic distinguishes transformer-based models from convolutional neural networks, whose locality constraints yield more favorable scaling properties for many industrial inspection tasks.

2.2.3.2 Vision Transformers and Practical Applications

To extend transformer architectures to visual data, images are represented as sequences of tokens, enabling the application of self-attention mechanisms originally developed for sequential inputs. In vision transformers (ViT) [66], an input image is first partitioned into a set of N non-overlapping patches of fixed spatial size. Each patch is flattened and

linearly projected into a d-dimensional embedding space, resulting in a sequence of visual tokens

$$\mathbf{X} \in \mathbb{R}^{N \times d},$$

where each row of $\mathbf{X}$ corresponds to the embedding of one image patch.

Since this patch-based representation does not inherently encode spatial relationships, positional encodings are added to the token embeddings to preserve information about the relative or absolute positions of image regions. The embedded token sequence is then processed by a stack of transformer encoder layers. Each encoder layer consists of a self-attention block followed by a position-wise feed-forward network, with residual connections and normalization applied between these components. The self-attention mechanism enables each token to aggregate information from all other tokens in the image, allowing global context to be modeled at every layer. The feed-forward networks further transform token representations independently, increasing the expressive capacity of the model. Depending on the task formulation, the output of the transformer encoder can be used in different ways. For image-level tasks such as classification, the token representations are typically aggregated into a single global descriptor that serves as the input to a prediction head [37, 66, 101, 161, 275, 323]. For dense prediction tasks, such as segmentation, the sequence of token embeddings can be reshaped or projected back into a spatial arrangement, forming feature maps that retain correspondence to image regions [41, 74, 137, 220, 257]. In this way, vision transformers provide a flexible framework that supports both global and spatially resolved visual representations.

Vision transformers have been adopted in a variety of architectural variants that address practical challenges in visual modeling. The original ViT architecture [66] applies global self-attention over all image patches, while hierarchical transformer variants, such as Swin Transformer [161], introduce a multi-scale structure by restricting attention to local windows and progressively merging tokens across layers. This design enables efficient modeling of high-resolution images while preserving the ability to capture long-range dependencies. Data-efficient transformer variants, such as DeiT [268], focus on improving training efficiency through architectural and training strategy modifications, allowing transformer models to be trained with reduced data requirements.

In this dissertation, transformer-based vision models are applied primarily in data generation and preprocessing stages rather than as downstream inspection backbones. One representative example is the Segment Anything Model (SAM) [137, 220], which is built upon a vision transformer backbone and encodes image features using self-attention. These features are combined with prompt-based conditioning to generate segmentation masks for arbitrary objects. In this dissertation, SAM is used as a preprocessing tool to automatically extract individual granules from large scene images, enabling subsequent processing at the object level without manual annotation (*cf.* Section 3.2). Transformers also play a key role in modern generative vision models. Large language vision models (LLVMs) and diffusion-based image generators frequently rely on transformer architectures or transformer-based attention modules to integrate visual representations with semantic conditioning [11, 196, 203, 210, 212, 265, 311]. In the context of this dissertation, such models are used for the synthesis of crack images, where transformer-based backbones enable flexible image generation conditioned on textual descriptions or latent representations (*cf.* Section 3.4).

2.3 Synthetic Data Generation for Industrial Visual Inspection

Synthetic image generation has become an important tool for industrial visual inspection, particularly in scenarios where collecting and annotating large amounts of real defect data is impractical. This section reviews common approaches for generating synthetic visual data, covering both modeling-based methods that rely on explicit scene construction and learning-based methods that infer image distributions from data. The strengths and limitations of these approaches are discussed to provide context for the synthetic data generation strategies adopted in this dissertation.

2.3.1 Modeling-Based Image Generation

Modeling-based image generation refers to approaches that synthesize images by explicitly constructing a virtual representation of the inspec-

tion scene and rendering it using graphics- or physics-based pipelines. In this paradigm, objects, backgrounds, illumination conditions, and camera parameters are defined explicitly, enabling controlled and repeatable image generation.

In industrial visual inspection, modeling-based generation is commonly realized using CAD tools, simulation environments, or 3D rendering engines [32, 70, 182, 183, 207, 209, 234, 240, 246, 294, 325]. These tools allow the geometry, material properties, and spatial arrangement of objects to be specified in detail, while rendering parameters such as lighting, viewpoint, and camera characteristics can be systematically varied. As a result, synthetic images can be generated under a wide range of controlled conditions that would be difficult or costly to reproduce in real-world acquisition.

One advantage of modeling-based image generation is the potential availability of precise ground-truth annotations. In commonly used modeling and rendering tools, such as Blender [54] or similar simulation environments, the explicit scene representation can be leveraged to export auxiliary information alongside rendered images. Depending on the rendering setup, labels such as object masks, defect locations, depth maps, or geometric measurements can be obtained directly from the underlying scene model without manual annotation. This capability is beneficial for tasks that require pixel-level supervision or detailed spatial information, provided that the modeling and rendering pipeline is designed to support such outputs. Furthermore, modeling-based generation offers a high degree of interpretability and controllability. The influence of individual scene parameters on the rendered appearance is transparent, which is beneficial in industrial inspection scenarios where defects often have well-defined geometric or physical characteristics [10, 144, 182, 242, 266]. This explicit control also enables targeted data generation, for example by systematically varying defect size, shape, or position.

However, modeling-based approaches also exhibit inherent limitations. Accurately reproducing complex surface textures, fine-grained appearance variations, or subtle imaging artifacts can be challenging and time-consuming. Consequently, images generated purely through modeling may differ from real inspection data in terms of low-level appearance statistics, leading to a domain gap that can reduce the effectiveness of

synthetic data when used directly for training data-driven models. Despite these limitations, modeling-based image generation provides a structured and controllable foundation for synthetic data creation. It is therefore frequently used as an initial step in synthetic-to-real pipelines, where explicitly generated images are subsequently refined or adapted to better match real-world data distributions, as discussed in Section 2.4.

2.3.2 Learning-Based Image Generation

Learning-based image generation refers to approaches that synthesize images by learning the underlying data distribution from examples rather than explicitly modeling the scene geometry and image formation process. In contrast to modeling-based generation, these methods rely on data-driven training to capture appearance characteristics such as texture, illumination, and noise patterns that are difficult to parameterize analytically.

Generative adversarial networks (GANs) [8, 82, 91, 123, 130, 146, 180, 217, 322] constitute a widely studied class of learning-based generative models. A GAN consists of two neural networks trained simultaneously: a generator $F^{(G)}$ parametrized by $\theta^{(G)}$, which maps a random latent vector $\mathbf{z} \sim p(\mathbf{z})$ to a synthetic image $F^{(G)}(\mathbf{z})$, and a discriminator $F^{(D)}$ parametrized by $\theta^{(D)}$, which aims to distinguish between real images and those produced by the generator. The two networks are optimized in an adversarial manner following the minimax objective

$$\min_{\theta^{(G)}} \max_{\theta^{(D)}} \; \mathbb{E}_{\mathbf{X}\sim p_{\text{data}}(\mathbf{X})}\left[\log F^{(D)}(\mathbf{X})\right] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log\left(1 - F^{(D)}\left(F^{(G)}(\mathbf{z})\right)\right)\right], \tag{2.22}$$

where $p_{\text{data}}(\mathbf{X})$ denotes the real data distribution and $p(\mathbf{z})$ is a predefined latent prior distribution (*e.g.*, a Gaussian or uniform distribution). In practice, the discriminator and generator are optimized via gradient-based minimization. Therefore, the discriminator objective is commonly rewritten as a minimization problem by taking the negative of the maximization term, yielding the discriminator loss

$$\mathcal{L}^{(D)} = -\mathbb{E}_{\mathbf{X}\sim p_{\text{data}}(\mathbf{X})}\left[\log F^{(D)}(\mathbf{X})\right] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log\left(1 - F^{(D)}\left(F^{(G)}(\mathbf{z})\right)\right)\right], \tag{2.23}$$

which is minimized with respect to $\theta^{(D)}$ while keeping $\theta^{(G)}$ fixed. For the generator, the optimization goal is to minimize the term

$$\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\Big[\log\Big(1 - F^{(D)}\Big(F^{(G)}(\mathbf{z})\Big)\Big)\Big] \,.$$

However, the formulation of this minimization term suffers from vanishing gradients when the discriminator becomes strong. In practice, the generator is therefore commonly trained using the non-saturating alternative

$$\mathcal{L}^{(G)} = -\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\Big[\log F^{(D)}\Big(F^{(G)}(\mathbf{z})\Big)\Big] \,, \tag{2.24}$$

which provides stronger gradients during training and improves convergence behavior. Training thus proceeds by alternating minimization steps of $\mathcal{L}^{(D)}$ with respect to $\theta^{(D)}$ and $\mathcal{L}^{(G)}$ with respect to $\theta^{(G)}$, reflecting the adversarial interplay between the two networks. The interaction between generator and discriminator during adversarial training is illustrated in Fig. 2.4. During training, the discriminator learns to better separate real and generated samples, while the generator is encouraged to produce images that are increasingly indistinguishable from real data. Beyond unconditional image synthesis, GANs can be extended to conditional and image-to-image translation settings, where the generator learns a mapping between two image domains [180, 195, 202, 282, 300, 322]. In such configurations, GANs are not only used for generating new images but also for reducing the domain gap. This capability has made GAN-based methods attractive for industrial visual inspection, both for synthesizing realistic object or defect appearances and for adapting synthetic images to real-world visual characteristics.

More recently, diffusion models [63, 107, 131, 226, 252–254] and flow-matching models [45, 79, 136, 154, 160, 199] have emerged as alternatives for learning-based image generation. Diffusion models formulate image synthesis as a two-stage stochastic process consisting of a fixed forward noising process and a learned reverse denoising process. In the forward diffusion process, a data sample $\mathbf{X}_0$ is gradually corrupted by Gaussian noise over T steps according to

$$p_{fwd}(\mathbf{X}_t \mid \mathbf{X}_{t-1}) = \mathcal{N}\Big(\mathbf{X}_t \mid \sqrt{1-\sigma_t}\,\mathbf{X}_{t-1},\, \sigma_t \mathbf{I}\Big) \,, \tag{2.25}$$

where p_{fwd} denotes the predefined forward transition distribution and $\{\sigma_t\}_{t=1}^{T}$ defines a variance schedule. After sufficiently many steps, the

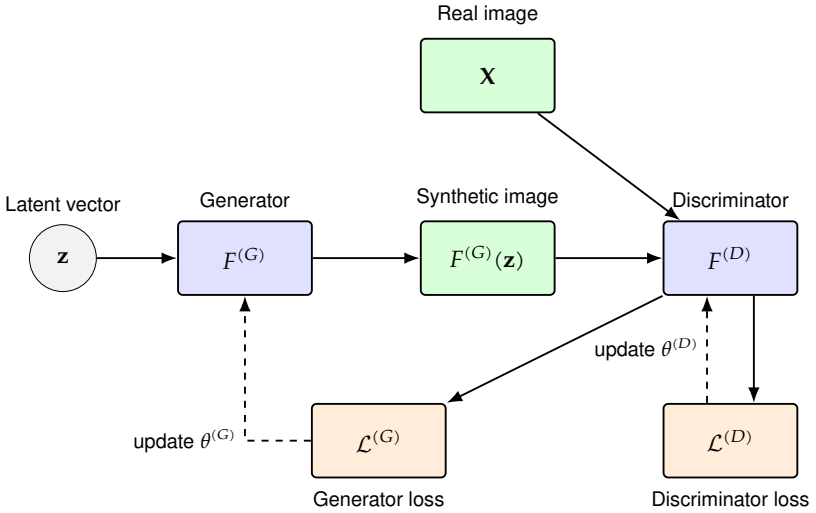


Figure 2.4 GAN training mechanism with forward signal flow shown by solid arrows and backpropagation of loss signals shown by dashed arrows for updating the generator and discriminator parameters.

distribution of $\mathbf{X}_T$ approaches a standard normal distribution. The generative model is then trained to approximate the reverse process by learning a parameterized conditional distribution $p_{rev,\theta}$, which models the reverse transition from noisy samples toward cleaner ones. During inference, new images are generated by starting from random noise and iteratively applying the learned reverse transitions,

$$p_{rev,\theta}(\mathbf{X}_{t-1} \mid \mathbf{X}_t)\,, \tag{2.26}$$

until a sample in the data space is obtained. The forward noising and learned reverse denoising processes are illustrated in Fig. 2.5.

Flow-matching models adopt a related but continuous-time formulation, where generation is described as a deterministic transformation from a noise distribution to the data distribution. Instead of discrete denoising steps, flow-matching methods learn a time-dependent vector

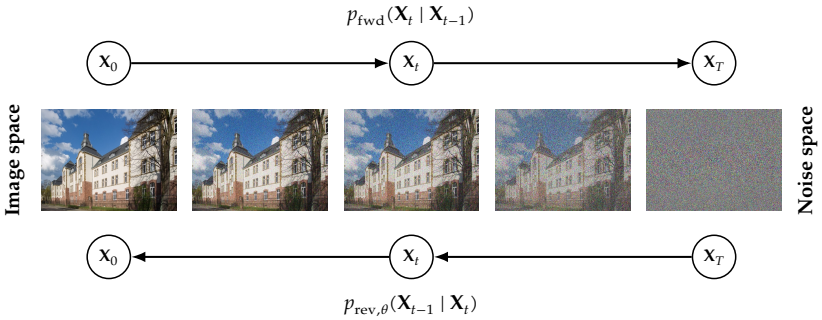


Figure 2.5 Mechanism of diffusion models. The forward transition stochastically perturbs data toward a Gaussian distribution, and the learned reverse transitions generate images by iterative denoising.

field $\mathbf{v}_\theta(\mathbf{X}, t)$ parameterized by θ that specifies how samples evolve over continuous time according to the ordinary differential equation

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{v}_\theta(\mathbf{X}(t), t)\,, \qquad t \in [0, 1]\,. \tag{2.27}$$

The model is trained to match this vector field to target probability flows that transport samples from an initial noise distribution to the data distribution. Image synthesis is then performed by numerically integrating the learned dynamics from $t = 0$ to $t = 1$, yielding a data sample.

From a unifying perspective, diffusion models can be interpreted as a particular instantiation of flow-based generative modeling [154, 254, 255]. Specifically, the reverse diffusion process corresponds to a stochastic probability flow that transports samples from a simple noise distribution to the data distribution. Flow-matching models generalize this idea by directly learning a continuous-time probability flow through a deterministic vector field, avoiding the need for explicit stochastic sampling. In this sense, diffusion models can be viewed as a discrete and stochastic special case within the broader class of flow-matching approaches.

Compared to modeling-based image generation, learning-based approaches typically require access to representative training data and substantial computational resources, as the underlying data distribution must be learned implicitly from examples [8, 29, 107, 146, 254, 297].

This reliance on data can limit their applicability in industrial scenarios where defect samples are scarce or highly imbalanced. In addition, learning-based methods generally offer less explicit control over geometric structure or physical consistency than modeling-based approaches, since such constraints are not directly encoded in the generation process. Despite these limitations, diffusion models and flow-matching models offer several important advantages. Their learning objectives avoid adversarial optimization, which generally yields more stable training behavior than adversarial optimization [63, 107, 154, 297]. Moreover, these models are capable of producing high-fidelity image samples that can closely match real data distributions, particularly with respect to fine-grained appearance statistics [63, 111, 226]. Their probabilistic and continuous formulations further support flexible conditioning mechanisms, such as class labels, spatial constraints, or textual prompts, making them well suited for controllable image generation. These benefits typically come at the cost of increased computational complexity during sampling, as image synthesis requires iterative denoising or numerical integration.

LLVMs build upon diffusion or flow-matching backbones and extend them with language-conditioned generation capabilities [11, 42, 155, 193, 218, 233, 265]. By leveraging models that have been pre-trained on large-scale image–text pairs, such approaches exhibit strong generalization and few-shot generation capabilities. This property is particularly relevant for industrial inspection scenarios, where only a limited number of real defect samples may be available. In the context of this dissertation, these models enable the direct synthesis of images (*cf.* Section 3.4) by specifying high-level semantic attributes through natural language prompts, reducing the need for extensive task-specific training data while still producing visually diverse and realistic samples.

2.4 Synthetic-to-Real Domain Gap and Image Refinement

Synthetic image refinement aims to reduce the discrepancy between synthetic and real images by adapting appearance characteristics while preserving task-relevant content. In industrial visual inspection, such refinement is often necessary to mitigate the domain gap that limits the

direct transferability of models trained on synthetic data. This section reviews representative refinement paradigms and introduces key criteria used to characterize and evaluate synthetic-to-real alignment, providing the conceptual background for the refinement strategies investigated in this dissertation.

2.4.1 Domain Gap and Refinement Criteria

A central challenge in leveraging synthetic data for industrial visual inspection is the presence of a domain gap between synthetic and real images. This gap arises from discrepancies in appearance characteristics that are not directly related to task semantics, such as texture statistics, illumination effects, noise patterns, and sensor-specific artifacts [27, 116, 182, 204, 242, 243, 248, 267, 294, 324, 326]. As a result, models trained on synthetic data may learn representations that do not transfer reliably to real inspection data, leading to degraded performance in downstream tasks.

Let $p_{\text{syn}}(\mathbf{X})$ and $p_{\text{real}}(\mathbf{X})$ denote the data distributions of synthetic and real images over the image space $\mathcal{X}$, respectively. The domain gap can be formalized as a discrepancy between these two distributions, either directly in image space or in a learned feature space induced by a feature extractor F. Synthetic image refinement aims to reduce this discrepancy by transforming samples drawn from $p_{\text{syn}}(\mathbf{X})$ such that the resulting distribution becomes closer to $p_{\text{real}}(\mathbf{X})$.

To characterize and evaluate the effectiveness of synthetic image refinement, this dissertation focuses on two complementary criteria: fidelity and variety.

Fidelity describes how closely refined synthetic images resemble real images in terms of visual appearance. A commonly used distribution-level metric for fidelity is the Fréchet Inception Distance (FID) [105], which compares the statistics of feature embeddings extracted from a pretrained network. Let $\mu_{\text{syn}}, \mu_{\text{real}}$ and $\Sigma_{\text{syn}}, \Sigma_{\text{real}}$ denote the empirical means and covariance matrices of synthetic and real features from a feature extractor F. The FID score is defined as

$$\text{FID} = \|\mu_{\text{syn}} - \mu_{\text{real}}\|_2^2 + \text{Tr}\Big(\Sigma_{\text{syn}} + \Sigma_{\text{real}} - 2\left(\Sigma_{\text{syn}}\Sigma_{\text{real}}\right)^{1/2}\Big) . \quad (2.28)$$

Lower FID values indicate closer alignment between the refined and real data distributions in feature space.

Another distribution-level criterion for assessing fidelity is the maximum mean discrepancy (MMD) [87], which measures the distance between two distributions based on kernel mean embeddings in a reproducing kernel Hilbert space. Given two sets of samples $\{\mathbf{X}^i_{\text{syn}}\}_{i=1}^N$ and $\{\mathbf{X}^j_{\text{real}}\}_{j=1}^M$ drawn from $p_{\text{syn}}(\mathbf{X})$ and $p_{\text{real}}(\mathbf{X})$, respectively, the squared MMD is defined as

$$\begin{aligned}\text{MMD}^2 = &\frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k\left(\mathbf{X}^i_{\text{syn}}, \mathbf{X}^{i'}_{\text{syn}}\right) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k\left(\mathbf{X}^j_{\text{real}}, \mathbf{X}^{j'}_{\text{real}}\right) \\ &- \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k\left(\mathbf{X}^i_{\text{syn}}, \mathbf{X}^j_{\text{real}}\right),\end{aligned} \tag{2.29}$$

where $k(\cdot,\cdot)$ denotes a positive-definite kernel function, commonly chosen as a Gaussian kernel. A lower MMD value indicates a smaller discrepancy between the two distributions. Unlike FID, which relies on second-order statistics of feature embeddings, MMD provides a non-parametric measure that can capture higher-order differences depending on the choice of kernel [87, 125].

In contrast to these distribution-level metrics, image-wise similarity measures such as mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [284], or learned perceptual image patch similarity (LPIPS) [312] operate on paired image comparisons. These criteria are suitable when a direct correspondence between synthetic and real images exists, for example in paired refinement scenarios. However, they are generally not applicable when synthetic and real datasets are unpaired, which is often the case in industrial inspection tasks. Consequently, such metrics are used selectively in this dissertation, depending on the availability of paired reference data.

Variety refers to the extent to which the refined synthetic dataset captures the range of appearance variations present in real data. Unlike fidelity, there is no universally accepted metric for measuring variety across two unpaired image sets. In practice, variety is often assessed by analyzing the spread of feature representations within a single dataset, typically through pairwise distances between feature embeddings or

coverage-based criteria that evaluate how comprehensively the feature space neighborhoods represent the real data manifold [145, 187, 195, 235, 236]. In this dissertation, variety is quantified using an intra-set perceptual distance measure, which evaluates the average pairwise distance between images within the refined dataset in a learned feature space (more details in Section 4.2.1). This criterion reflects how broadly the refined data spans the appearance space and provides a practical proxy for dataset diversity. Such intra-set measures are particularly useful when simple assumptions (*e.g.*, a uniformly dense embedding space [235]) about the dataset do not hold or when paired correspondence is unavailable.

Fidelity and variety are inherently related and often exhibit a trade-off. Refinement strategies that strongly enforce alignment with real data appearance may improve fidelity while reducing variety [29, 130, 187], whereas methods that encourage diverse outputs may introduce samples that deviate from task-relevant characteristics [27, 248]. For this reason, effective synthetic image refinement requires balancing these two criteria rather than optimizing either in isolation.

2.4.2 GAN-Based Image Refinement

GAN-based image refinement addresses the synthetic-to-real domain gap by learning a mapping between two image domains rather than generating images from random noise. In this setting, synthetic images are treated as inputs to the generator, and the objective is to transform their appearance such that the resulting images resemble real data while preserving task-relevant content. This formulation naturally casts refinement as an image-to-image translation problem.

Let $\mathbf{X}_{\text{syn}} \sim p_{\text{syn}}(\mathbf{X})$ denote a synthetic image and $\mathbf{X}_{\text{real}} \sim p_{\text{real}}(\mathbf{X})$ a real image. GAN-based refinement aims to learn a transformation function $F^{(G)}$, parameterized by $\theta^{(G)}$, such that the distribution induced by $F^{(G)}(\mathbf{X}_{\text{syn}})$ approximates the real image distribution $p_{\text{real}}(\mathbf{X})$. Unlike unconditional GANs, the generator is conditioned on the input image, and the discriminator is trained to distinguish refined images from real ones. The adversarial objective encourages the refined outputs to match the appearance statistics of real data.

A central challenge in GAN-based refinement is the preservation of semantic and structural content. Since paired synthetic–real image correspondences are typically unavailable, many refinement methods rely on additional constraints to enforce consistency. Zhu *et al*. [322] introduce an inverse mapping between domains and penalize reconstruction error after a forward–backward transformation, thereby discouraging excessive content distortion. Park *et al*. [202] use contrastive patch-level objectives to constrain local feature representationsso that they remain consistent between the input and refined images without requiring explicit inverse mappings.

By learning complex, non-linear transformations, GAN-based refinement methods can effectively reduce discrepancies in texture, color distribution, and illumination between synthetic and real data [8, 91, 121, 130, 202, 208, 217, 248, 322]. As a result, refined images often achieve high fidelity with respect to real-world appearance metrics. However, these methods also exhibit characteristic limitations. Adversarial training can be sensitive to hyperparameter choices and training stability, and refinement models may suffer from mode collapse or reduced output diversity [8, 63, 91]. Furthermore, strong global appearance adaptation may inadvertently suppress subtle but task-relevant structures, particularly when these structures are small or localized [112, 177, 182, 243, 285, 287]. These factors make the balance between fidelity and variety a critical consideration when applying GAN-based refinement in industrial inspection scenarios.

Overall, GAN-based image refinement provides a powerful framework for bridging the synthetic-to-real domain gap through data-driven appearance adaptation. Its strengths and limitations motivate the exploration of alternative refinement paradigms that offer different trade-offs between realism, diversity, and structural preservation.

2.4.3 Neural Style Transfer for Image Refinement

Neural style transfer (NST) [75, 76] was originally proposed as a method for modifying the artistic appearance of an image while preserving its semantic content. Unlike generative models that synthesize images from noise or learn explicit domain mappings, NST operates by directly optimizing an output image to match predefined feature statistics extracted

from a pretrained convolutional neural network. This property makes NST particularly attractive for image refinement scenarios where structural consistency must be preserved. The fundamental idea of NST is to decouple image representations into content and style components in a learned feature space. Given a content image $\mathbf{X}_{\text{con}}$ and a style image $\mathbf{X}_{\text{sty}}$, feature activations at selected layers of a pretrained convolutional network are used to define content and style representations. Let $\mathbf{h}^{(l)}$, $\mathbf{h}_{\text{con}}^{(l)}$, and $\mathbf{h}_{\text{sty}}^{(l)}$ denote the feature maps at layer l corresponding to the optimized image, the content image, and the style image, respectively. The content discrepancy is defined as

$$\mathcal{L}_{\text{con}} = \left\| \mathbf{h}^{(l)} - \mathbf{h}_{\text{con}}^{(l)} \right\|_F^2 . \tag{2.30}$$

The style representation is defined through correlations between feature channels, captured by Gram matrices. For $\mathbf{h}^{(l)} \in \mathbb{R}^{C_l \times H_l \times W_l}$, the spatial dimensions are flattened to obtain a matrix in $\mathbb{R}^{C_l \times (H_l W_l)}$, and the Gram matrix is computed as

$$\mathbf{G}^{(l)} = \mathbf{h}^{(l)} \mathbf{h}^{(l)\top} . \tag{2.31}$$

The corresponding style discrepancy is formulated as

$$\mathcal{L}_{\text{sty}} = \sum_{l \in \mathcal{S}} \left\| \mathbf{G}^{(l)} - \mathbf{G}_{\text{sty}}^{(l)} \right\|_F^2 , \tag{2.32}$$

where $\mathbf{G}_{\text{sty}}^{(l)}$ denotes the Gram matrix of the style image at layer l, and $\mathcal{S}$ denotes the set of layers used to represent style.

The overall NST objective combines content and style discrepancies into a single optimization problem,

$$\min_{\mathbf{X}} \ \mathcal{L}_{\text{NST}} = \alpha \mathcal{L}_{\text{con}} + \beta \mathcal{L}_{\text{sty}} , \tag{2.33}$$

where α and β control the relative importance of content preservation and style adaptation. The output image is obtained by directly optimizing $\mathbf{X}$ using gradient-based methods, starting from an initialization such as the content image itself. The overall optimization procedure of NST is illustrated in Fig. 2.6.

From a refinement perspective, NST can be interpreted as an appearance adaptation process that modifies low-level statistics while explicitly

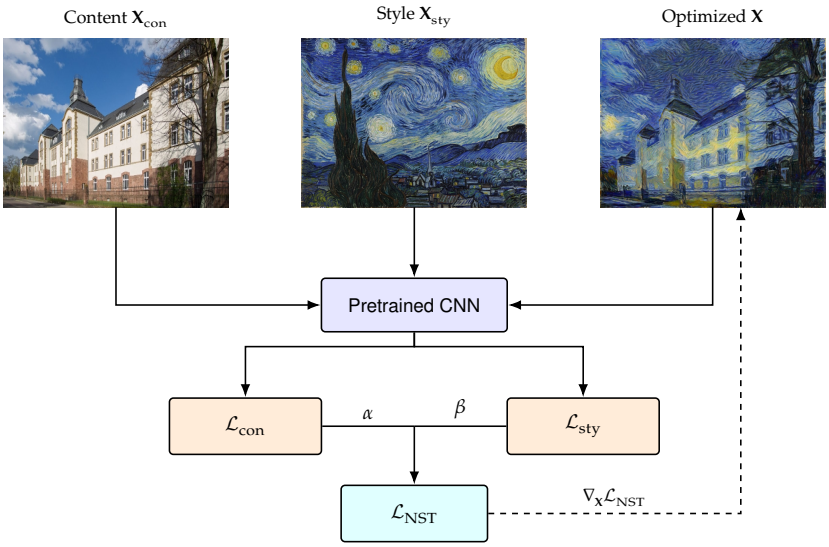


Figure 2.6 Illustration of the neural style transfer framework. A fixed pretrained convolutional network extracts content and style representations from the respective images, and the resulting losses are combined to iteratively optimize the output image.

constraining high-level feature structure. Since the content loss anchors the optimization to the original synthetic image, task-relevant structures such as object geometry or defect morphology are intended to be preserved by construction. At the same time, the style loss encourages the refined image to adopt appearance characteristics of real data, including texture, contrast, and noise patterns.

NST has been widely applied in artistic image stylization [38, 53, 75–77, 139, 200, 230] and has also been explored for reducing domain gaps in applications such as autonomous driving and simulation-to-real transfer [22, 143, 156, 185]. In industrial visual inspection, however, its application has been comparatively limited, partly due to the computational cost of optimization-based NST and the need for careful parameter tuning. Despite these challenges, the explicit separation of content and style makes NST particularly suitable for refinement scenarios where preserving structural consistency is critical. An important advantage of

NST-based refinement lies in its interpretability and controllability. The influence of content and style is explicitly defined through loss terms and weighting factors, allowing refinement behavior to be adjusted in a transparent manner. Moreover, because each refined image is optimized independently, NST does not rely on learning a single global generator, which can help avoid mode collapse and preserve diversity across the refined dataset.

Overall, neural style transfer provides a conceptually distinct refinement paradigm compared to GAN-based methods. By operating directly in feature space and enforcing content preservation through explicit constraints, NST offers a flexible and interpretable approach to appearance adaptation that is well suited for scenarios requiring a careful balance between fidelity and variety.

2.4.4 Diffusion-Based Image Refinement

Diffusion-based image refinement extends diffusion generative models to conditional or guided settings, where an existing image serves as an input rather than starting from pure noise. In this context, refinement can be formulated as a process that progressively adapts a synthetic image toward the real data distribution while maintaining a degree of correspondence to the original input. This is commonly achieved by conditioning the denoising process on the synthetic image. Meng *et al*. [178] inject synthetic images as an initial state and add noise at an intermediate diffusion step. Ho and Salimans [108] use the synthetic image as a conditioning signal throughout the reverse process.

Formally, given a synthetic image $\mathbf{x}_{\text{syn}}$, refinement can be realized by first perturbing it with noise according to a forward diffusion process and then applying a learned reverse process conditioned on $\mathbf{x}_{\text{syn}}$.

Compared to unconditional generation, such conditional diffusion formulations offer greater control over the output and have the potential to balance fidelity and diversity more effectively than adversarial approaches. Despite these advantages, diffusion-based refinement methods are still an emerging research direction, particularly in industrial visual inspection scenarios. Most existing diffusion models are trained for unconditional generation [52, 107, 132, 167, 226, 253, 254] or text-conditioned synthesis [78, 134, 153, 188, 311, 314], and adapting them

for image-to-image refinement requires large-scale training data and substantial computational resources. In addition, the iterative nature of the denoising process leads to high computational cost during inference, which can be a limiting factor in practical inspection pipelines.

In the context of this dissertation, diffusion-based refinement is not explored further. The available datasets are relatively limited in size, and the refinement task requires strict preservation of fine-grained, task-relevant structures. Under these constraints, GAN-based and NST–based refinement methods provide more practical and controllable solutions, allowing systematic analysis of fidelity and variety without the overhead associated with training and deploying diffusion-based refinement models.

Nevertheless, diffusion-based refinement represents a promising direction for future research. Advances in conditional diffusion, flow-matching formulations, and efficiency-oriented model designs may enable their effective application to industrial refinement tasks as data availability and computational resources continue to improve.

2.5 Training Stability and Model Selection under Limited Data

Training stability and model selection are critical challenges in industrial visual inspection, particularly in settings where labeled validation data is unavailable or severely limited. This section reviews factors that contribute to unstable training dynamics and overfitting in deep learning models, as well as commonly used regularization and early stopping strategies. Special attention is given to scenarios such as anomaly detection and one-class learning, where conventional validation-based model selection is not directly applicable, thereby motivating alternative approaches for training control.

2.5.1 Overfitting and Training Instability in Deep Models

Overfitting and training instability are long-standing challenges in machine learning and become particularly pronounced in deep neural networks. In classical statistical learning theory, overfitting is understood as

the phenomenon where a model with high capacity adapts excessively to the training data, capturing noise or incidental patterns that do not generalize to unseen samples. Bishop and Nasrabadi [20] as well as Hastie *et al*. [96] formalize this behavior through the bias–variance trade-off, showing that minimizing empirical risk alone is insufficient to guarantee good generalization.

In the context of deep learning, this issue is compounded by the representational capacity of deep neural networks. Goodfellow [81] emphasizes that deep models can represent highly complex functions, making them susceptible to memorization even when trained on relatively small datasets. Zhang *et al*. [309] further demonstrate that deep networks are capable of fitting random labels with zero training error, highlighting that conventional explanations of overfitting based solely on model capacity are insufficient to fully characterize generalization behavior in deep learning.

Formally, consider a model F_θ trained on a dataset $\mathcal{D}_{\text{train}} = \{(\mathbf{X}^i, \mathbf{y}^i)\}_{i=1}^N$ by minimizing the empirical risk as defined in (2.16). Overfitting occurs when continued reduction of the training objective $\mathcal{L}(\theta)$ leads to increasingly specialized parameter configurations that fail to generalize to the underlying data-generating distribution. In practice, this manifests as a divergence between training performance and generalization ability, even though the optimization objective appears to improve.

Training instability, while related to overfitting, arises from difficulties in the optimization process itself. Glorot and Bengio [80] identify poor gradient propagation as a key obstacle in deep networks, showing that inappropriate weight initialization can lead to vanishing or exploding activations and gradients. This analysis motivated initialization schemes designed to stabilize signal variance across layers, later extended by He *et al*. [99] for rectified nonlinearities. Beyond initialization, instability can also result from the non-convex and highly structured loss landscapes characteristic of deep models. Keskar *et al*. [133] link optimization dynamics to generalization behavior by showing that certain training configurations converge to sharp minima that are sensitive to perturbations, leading to unstable or brittle solutions. From an optimization standpoint, instability may be reflected by large fluctuations in parameter updates

or by rapidly changing internal representations despite monotonically decreasing training loss.

Additional sources of instability stem from the depth of modern architectures and the stochastic nature of gradient-based optimization. Early analyses by Bengio *et al.* [16] and Hochreiter [109] show that gradient-based learning can struggle to propagate informative signals across many layers, leading to slow convergence or unstable dynamics.

In industrial visual inspection, these challenges are further amplified by limited training data, strong class imbalance, and the use of synthetic data [14, 28, 33, 194, 231, 267, 318]. While such data can enrich training sets, they may also introduce appearance variations or distributional shifts that interact unfavorably with the learning dynamics. As a result, deep models may exhibit unstable feature representations and heightened sensitivity to training hyperparameters, even when the optimization objective appears well behaved. Taken together, overfitting and training instability reflect fundamental limitations of empirical risk minimization and gradient-based optimization in deep learning. Understanding these phenomena is essential for motivating training control strategies that go beyond loss minimization and form the basis for the model selection challenges discussed in the following sections.

2.5.2 Early Stopping and Regularization Strategies

To mitigate overfitting and stabilize optimization in deep neural networks, a variety of regularization strategies have been developed. These methods aim to control model complexity, smooth the learning dynamics, or reduce sensitivity to noise in the training data, thereby improving generalization performance.

Early stopping is one of the most widely used regularization techniques in practice. The core idea is to terminate training before full convergence to the minimum of the empirical risk, thereby preventing the model from excessively adapting to the training data. Bishop [19] and Goodfellow [81] describe early stopping as an implicit form of regularization, showing that limiting the number of gradient updates effectively constrains the complexity of the learned solution. From a theoretical perspective, Yao *et al.* [299] demonstrate that, under certain conditions, early stopping in gradient-based optimization is equivalent to explicit regularization

methods, such as norm penalties, in terms of their effect on generalization.

In practice, early stopping is commonly implemented by monitoring model performance on a validation set and selecting the checkpoint that optimizes a chosen validation criterion. Prechelt [215] analyzes this procedure empirically and shows that its effectiveness depends strongly on the quality and representativeness of the validation signal, as well as on the noise characteristics of the monitored metric. These findings highlight that early stopping, while conceptually simple, can be sensitive to training conditions and hyperparameter choices.

Beyond early stopping, several explicit regularization techniques are routinely applied in deep learning. Krogh and Hertz [141] introduce weight decay as a means of penalizing large parameter values, thereby discouraging overly complex solutions. Srivastava *et al.* [256] propose dropout, which randomly deactivates a subset of activations during training to reduce co-adaptation between neurons and improve robustness. Data augmentation represents another widely used regularization strategy, where label-preserving transformations are applied to input samples to increase the effective size and diversity of the training set [56, 81, 140, 247, 276].

Normalization methods also play an important role in stabilizing training and implicitly regularizing deep models. Ioffe and Szegedy [122] introduce batch normalization, showing that normalizing intermediate activations reduces sensitivity to initialization and learning rates. Subsequent analyses by Santurkar *et al.* [239] demonstrate that batch normalization smooths the optimization landscape, leading to more stable and predictable training dynamics.

While these regularization strategies are effective in many supervised learning settings, their performance is closely tied to appropriate hyperparameter selection and training configuration. Moreover, early stopping methods that rely on validation performance assume the availability of reliable validation data. These dependencies motivate further discussion of model selection strategies in scenarios where such assumptions do not hold, as addressed in the following section.

2.5.3 Model Selection without Validation Data

Model selection is a fundamental component of supervised learning, where hyperparameters and training duration are typically chosen based on performance on a labeled validation set. However, this becomes problematic in learning scenarios where validation data are unavailable, unreliable, or unrepresentative of deployment conditions. In practice, such situations are common in unsupervised learning, one-class classification, and anomaly detection in industrial visual inspection. In one-class learning, the training objective is defined exclusively on samples from a single class, typically corresponding to normal operation. Schölkopf *et al*. [245] formalize this setting as support estimation in high-dimensional spaces, where the goal is to characterize the region occupied by normal data without access to examples from outside the support. Tax and Duin [264] further assume that no representative anomalous samples are available during training or validation, making conventional model selection strategies inapplicable. These assumptions extend naturally to modern anomaly detection benchmarks. Bergmann *et al*. [18] define an industrial inspection protocol in which models are trained solely on normal samples, and no labeled validation anomalies are assumed. Performance is evaluated only at test time using a held-out set containing both normal and defective samples. Ruff *et al*. [231] adopt a similar setting in deep one-class classification, highlighting that the absence of validation anomalies is not a limitation of dataset design but rather a reflection of realistic industrial conditions.

In the absence of validation data, model selection is often performed using heuristic or proxy-based criteria. Akcay *et al*. [4] and Perera *et al*. [206] report selecting checkpoints based on reconstruction error statistics, training loss behavior, or qualitative inspection of generated samples. While such approaches can yield reasonable performance in controlled experiments, they lack robustness and reproducibility, as they rely on implicit assumptions about training dynamics or data distributions that may not hold across different datasets or model architectures.

More generally, Hastie *et al*. [96] argue that model selection without validation data requires alternative principles, such as complexity control, information-theoretic criteria, or intrinsic properties of the learned representations. In deep learning, Bengio *et al*. [17] show that unsuper-

vised and self-supervised training regimes often rely on training-internal signals rather than explicit validation metrics, shifting the focus from performance monitoring to analysis of learning dynamics.

Recent self-supervised learning methods further illustrate this trend. Chen *et al*. [46] and Grill *et al*. [88] demonstrate that meaningful representations can be learned without labeled validation data, but they also highlight that assessing convergence and selecting training checkpoints remain nontrivial. These works implicitly motivate the use of stability, consistency, or convergence-related signals derived from the training process itself as indicators of model quality.

Overall, existing literature shows that validation-free model selection is an inherent challenge in anomaly detection and related learning paradigms. While heuristic strategies are commonly employed in practice, they provide limited guarantees regarding robustness and deployability. This motivates the exploration of training-intrinsic criteria for model selection.

3 Image Synthesis for Materials and Defects

The limited availability of annotated visual data remains a central challenge for training deep learning models in industrial inspection. To mitigate this issue, synthetic images are generated for two representative industrial inspection tasks: plastic granule classification and metal crack detection. For plastic granule classification, photorealistic rendering is performed using 3D models in Blender [54], simulating both normal and defective granules. These are tailored to match two real-world data batches captured under different lighting and camera setups. For metal crack detection, prompt-guided generation is carried out via large language vision models (LLVMs), producing high-resolution images of metallic components with surface cracks. Although ground-truth annotations are unavailable for the corresponding real images, the synthetic data provides a controlled basis for training and evaluation in downstream tasks. The methods introduced here establish the data basis for the refinement techniques in Chapter 4 and the performance analyses in Chapter 6.

3.1 Introduction

Synthetic data generation has become increasingly important in industrial visual inspection, particularly when real defective samples are costly to obtain. Previous research has employed procedural pipelines and physically based rendering to simulate defect-bearing surfaces in industrial components, reporting improved downstream model performance in some inspection settings [23, 50, 116, 242, 258]. Alternatively, GAN-based frameworks like Defect-GAN [310] have synthesized high-fidelity defects by modeling defacement and restoration processes, showing competitive

performance in anomaly detection tasks. More recently, inpainting-based diffusion models have been developed for realistic defect insertion, with some works reporting state-of-the-art results on specific visual inspection benchmarks [24, 150, 251]. Meanwhile, prompt-guided image generation using LLVMs and advanced prompt engineering has emerged as a flexible tool for defect synthesis without explicit geometric priors [277]. In this chapter, synthetic datasets are generated for two distinct inspection tasks. First, for plastic granule classification, 3D models are built from scratch and rendered in Blender using physically plausible materials and domain-specific lighting and camera parameters. This setup emulates two real acquisition conditions with varying optics and scene configurations. Second, for metal crack detection, high-resolution images are generated through prompt-guided synthesis using LLVMs, allowing control over crack appearance via natural language instructions. Guided by prompt engineering techniques, this approach can produce realistic and varied crack geometries without manual annotation or pre-existing geometric defect designs.

Compared with many prior approaches, the proposed 3D modeling and rendering pipeline is fully automated and can be adapted to different lighting and camera settings with limited manual modification. In addition, LLVM-based generation is applied directly to unstructured metal surfaces with crack defects, relying on prompt engineering rather than conventional generative networks or inpainting models.

The contributions of this chapter are as follows:

- A Blender-based 3D rendering pipeline for generating synthetic plastic-granule images under two distinct acquisition settings, with fully automated object generation, scene setup, rendering, and labeling, while material properties are specified manually.
- Prompt-based generation of synthetic crack images on metallic parts using LLVMs, enabling flexible and realistic defect synthesis.
- Synthetic datasets for both tasks, with task-appropriate annotations including image-level labels and pixel-level masks where required, to support controlled training and evaluation in downstream visual inspection methods.

3.2 Real Datasets

This section describes the real datasets used as visual references and as downstream evaluation datasets related to the synthetic image generation process. It includes two batches of granule images acquired under different industrial setups.

The first batch contains 513 images captured under a reflected-light illumination setup during the free fall of plastic granules. The granules appear densely packed with frequent overlap and depth variation, often resulting in occlusions.

The second batch contains 1324 images captured under a transmitted-light illumination setup, where granules are transported on a slide and imaged using a line-scan camera. This configuration leads to sparse, well-separated granules with sharp edge highlights and minimal occlusion.

The defect type of granules in both batches is burn marks, which normally appear in the form of small black specks. The burn marks are formed due to overheating during the injection process of plastics and can negatively affect subsequent product quality due to physical and chemical differences from normal granules.

To enable model training and evaluation, both sets of raw images are processed using the Segment Anything Model (SAM) to extract individual granules. The segmentation is performed using the ViT [66]-based model variant, with parameters chosen to favor precise mask generation, including a predicted IoU threshold of 0.86, stability score threshold of 0.92, and a minimum region size of 80 pixels. Each segmented granule is cropped into a separate image and manually labeled. The cropped classification datasets are referred to as $\mathcal{D}^{\text{rl}}_{\text{real}}$ and $\mathcal{D}^{\text{tl}}_{\text{real}}$ in the following.

The resulting datasets are as follows:

- $\mathcal{D}^{\text{rl}}_{\text{real}}$: 2000 crops of normal granules and 400 defective samples.
- $\mathcal{D}^{\text{tl}}_{\text{real}}$: 745 crops of normal granules and 745 defective samples.

Examples from both raw sets of granule images are shown in Fig. 3.1, and illustrate the visual characteristics used to configure the rendering pipeline described in the next section.

In addition to granule inspection, a real dataset of metal parts $\mathcal{D}^{\text{crack}}_{\text{real}}$ is used for studying crack detection. This dataset consists of 195 normal



(a) Example from the first batch under reflected-light illumination.



(b) Example from the second batch under transmitted-light illumination.

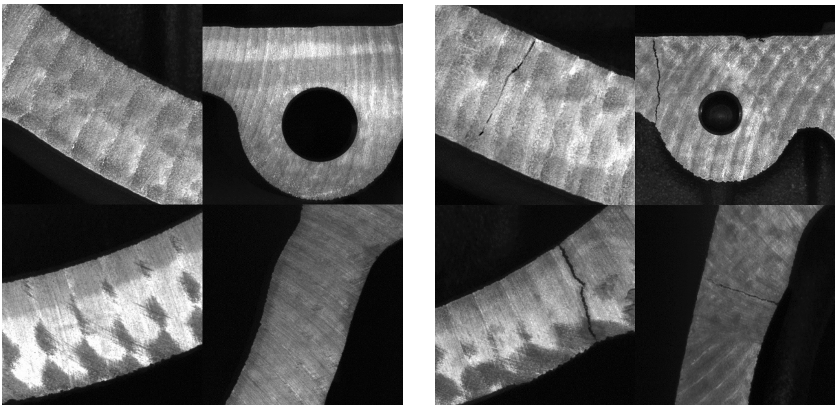
Figure 3.1 Examples from two batches of real granule images used as visual references for synthetic data generation. Each example shows one normal granule marked by a green box and one defective granule with a burn mark marked by a red box. The granules in Fig. 3.1(b) are zoomed out for better visualization.

metal part images and 15 images containing surface cracks. The parts are imaged under consistent lighting and background conditions, resulting in relatively uniform contrast and limited spatial variation across the dataset. The cracks vary in size and morphology, with some appearing as narrow surface fissures and others as more irregular damage patterns. These real examples are used as references for prompt-guided synthetic image generation, as discussed in Section 3.4. Examples of normal metal parts and metal parts with crack are shown in Fig. 3.2.

3.3 Scene Rendering Based on 3D Modeling

To generate synthetic images of plastic granules used for deep learning, a controlled 3D rendering pipeline is developed using Blender. The goal is to simulate granule samples in a visually plausible yet parametrically controllable environment, facilitating the generation of diverse datasets that reflect domain-specific variability in illumination, geometry, and object placement.

The pipeline builds upon the characteristics observed in the real datasets shown in Fig. 3.1, aiming to replicate the optical and spatial conditions of both raw granule image batches through domain-adapted rendering setups. Each synthetic scene is constructed using a modular process with four key components:



(a) Normal metal parts. **(b)** Metal parts with crack.

Figure 3.2 Example images from the metal part dataset $\mathcal{D}_{\text{real}}^{\text{crack}}$.

- Geometry modeling, which defines the 3D shape and structural variability of individual granules.
- Environment modeling, which configures camera placement, lighting setup, and background surfaces to match the target acquisition domain.
- Material modeling, which assigns optical properties to granules, defects, and backgrounds for physically plausible rendering.
- Scene composition and annotation, which arranges objects in space and generates structured labels for downstream tasks.

Each component is implemented programmatically to support scalable and repeatable generation of annotated synthetic data under diverse conditions. The following subsections provide a detailed overview of each module.

3.3.1 Geometry Modeling

The synthetic granules are modeled using procedural geometry operations in Blender. To reflect the typical shape of the granules considered

here, each object is initialized as a high-resolution cylinder, approximating the rounded and elongated geometry observed in real samples. Further refinement is applied by manipulating the internal mesh structure, including localized insets and vertex connections, which serve to increase the granule's surface complexity and improve its apparent physical plausibility during rendering.

To achieve a diverse yet realistic set of granules, each instance is generated from a cylindrical base mesh whose end faces are subdivided and whose edges are rounded using beveling. The resulting geometry is then subjected to several parameterized local deformations. Localized dents are produced by displacing selected vertices and neighboring faces using proportional editing with smooth or spherical falloff. Torsional variations are introduced by rotating predefined edge regions around the longitudinal axis, while local scaling and rotations of the upper and lower surface regions produce tapering, asymmetry, and subtle warping. Additional compression-like deformation is generated by translating two selected edge regions in opposite lateral directions.

The strength of these transformations is randomized using bounded normal or uniform distributions. The sampled values are clipped to predefined ranges to prevent unrealistic shapes, such that small deviations from the nominal geometry occur more frequently than severe distortions. After local deformation, each granule is first subjected to a small isotropic scale variation and subsequently to anisotropic scaling along the three spatial axes. The latter modifies both the overall size and the aspect ratio while constraining the dimensions to a predefined range.

In addition to the granules, separate synthetic defect elements are generated from spherical meshes. Several local regions of each sphere are independently displaced using proportional editing, resulting in irregular and asymmetric particles rather than ideal spherical shapes. Their final size is randomly sampled within a narrow range and kept substantially smaller than that of the granules. During scene generation, these elements are associated with a subset of granules and assigned contrasting material properties to imitate localized burn marks in the form of small dark specks.

3.3.2 Material Assignment

To realistically simulate the visual appearance of granules and defects under different illumination settings, physically inspired material shaders are implemented using Blender's node-based system. These shaders govern the interaction of granule surfaces with light, including reflection, transmission, transparency, and edge scattering. Separate materials are assigned to granules, defects, and background surfaces to support independent control of their visual properties during rendering.

Granules are modeled as semi-transparent plastic materials, reflecting their real-world physical properties. The material combines a glass BSDF and a transparent BSDF, blended through a fresnel module and light path–based mixing strategy. This configuration allows both surface reflection and light transmission, with reflectivity varying depending on the viewing angle and ray type. A mild roughness parameter is introduced to diffuse the surface reflections, and a geometry-dependent shading modulation is used to produce spatial variation across the granule body. This results in synthetic granules that exhibit light scattering and edge glow resembling those observed in the target domain, especially under reflected-light conditions.

The burn marks are assigned a separate material with reduced transparency and increased absorption. Visually, they appear darker and more opaque, closely resembling the small black specks observed in the real datasets. The defect material uses a simplified version of the granule shader but with much lower transmission and a more compact geometry, making it visually distinguishable when rendered in combination with a granule. This material contrast defines a granule as defective in the rendered image.

Background surfaces are also modeled differently depending on the target domain. For synthesizing the reflected-light granule images in the first batch, the background is assigned a neutral gray diffuse material, whereas for synthesizing the transmitted-light granule images in the second batch, an emissive shader is used and the background in this case is uniformly lit using a constant emission color, producing high-contrast granule silhouettes. This setup is intended to replicate the consistent background illumination observed in the real dataset.

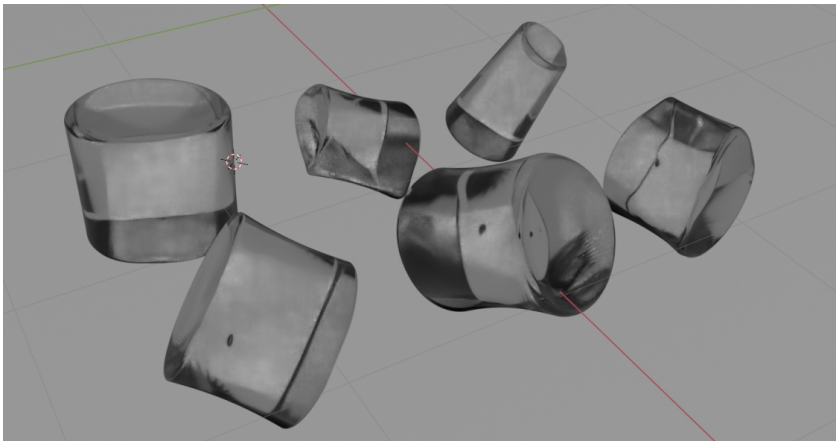


Figure 3.3 Examples of modeled normal granules and defective granules with burn marks in Blender.

Representative examples of granules and defects after geometry modeling and material assignment are shown in Fig. 3.3.

3.3.3 Environment Modeling

The environment setup defines the optical and spatial context in which synthetic granules are rendered. To replicate the visual characteristics of both batches of real granule images, two distinct illumination and background configurations are implemented.

For synthesizing the first batch of real granule images, which was acquired under reflected-light illumination, the background is implemented as a passive gray surface without any emissive properties. Instead, the scene is illuminated by three point light sources placed on the same plane of the camera, all positioned above the granules. This lighting arrangement not only approximates diffuse frontal illumination but also creates an intentionally uneven brightness distribution across the background, which mimics the non-uniform lighting observed in the real images. The camera is configured for top-down capture at a resolution matched to the original dataset, preserving visual fidelity in terms of granule scale and field of view.

For synthesizing the second batch of real granule images, corresponding to a transmitted-light imaging setup, the background plane itself is used as the primary light source. This plane is positioned below the granules and assigned an emissive material that simulates the illuminating slide used in the real acquisition. The camera remains in a top-down position, configured with a higher resolution, reflecting the sharper detail and lower granule density seen in the corresponding real images.

Both configurations are instantiated procedurally, allowing automated switching between lighting strategies and camera parameters. This controlled design ensures that the synthetic scenes accurately reflect the illumination geometry and visual structure of the respective target domains.

3.3.4 Scene Composition and Annotation

The final step in the data generation pipeline involves spatially arranging the modeled granules and defects within the virtual scene and generating corresponding annotations for downstream use. Since both batches of real granule images differ significantly in spatial configuration and acquisition context, two separate composition strategies are employed to mirror their respective characteristics. Before this scene-level placement, however, defective granules must first be composed by embedding small defect meshes, *i.e.*, the burn marks, into the granule bodies in a physically consistent manner. For both datasets, the combination follows the same controlled embedding procedure. First, a maximum inscribed cube is identified within the volumetric boundaries of a given granule mesh. This cube serves as the placement volume into which a defect mesh can be safely inserted without protruding from the granule surface. Second, the minimum enclosing cube of the defect mesh is computed. This ensures that the entire defect geometry can be rotated and translated into the granule without geometric intersection or exposure. A random 3D rotation is then applied to the defect mesh to introduce variability in appearance and orientation. Finally, the defect is placed within the granule by aligning the two cubes and translating the defect to a random position inside the granule's interior volume. Once placed, the defect and granule are grouped and treated as a single defective granule for subsequent scene composition and rendering.

Following this embedding stage, granules are placed into scenes based on the target dataset's spatial characteristics. In the case of synthesizing the batch of real granule images captured under a reflected-light setup, the goal is to replicate the uncontrolled and dense distribution of granules captured during free fall. To simulate this effect, granules are randomly placed within a defined volumetric region above the background, without enforcing collision constraints. The placement allows for variation in depth and overlapping projections, resulting in occlusion patterns and visual density similar to those seen in the real images. Rotational variation is introduced through random 3D orientation of each granule, further enhancing the geometric diversity in projection. This arrangement is intended to reflect inspection systems where granules are not individually isolated but appear in clusters or bulk material flow.

For synthesizing the batch of real granule images captured under a transmitted-light setup, a more structured and sparse layout is required to reflect the controlled placement of granules on a slide under the transmitted-light illumination. To achieve this, a customized object placement algorithm based on Poisson disc sampling is implemented in the 2D image plane. This algorithm generates candidate positions by enforcing a minimum separation radius between any two sample points, ensuring that granules are uniformly distributed without overlap. The output coordinates are mapped onto a fixed horizontal plane in 3D space, with granules subsequently positioned at those locations and given randomized in-plane rotations. This process guarantees a non-overlapping layout that corresponds to the real inspection conditions.

Once the scene is composed, annotations are automatically generated based on object geometry and material assignment. Granules with attached defect meshes or granules assigned a defect-specific material are labeled as defective, while all remaining granules are labeled as normal. During rendering, each object is assigned a unique index, which enables identification of individual granules at the object level. The annotation pipeline produces object-level class-labeled bounding boxes, which are computed by projecting the 3D vertex coordinates of each granule mesh into the camera plane using geometric transformation utilities provided by the rendering engine. This process yields tight axis-aligned bounding boxes that enclose the visible extent of each granule instance. To

ensure compatibility with standard deep learning pipelines, all annotations, including category labels, image dimensions, and bounding box coordinates, are stored in a structured format alongside the rendered images.

By aligning scene composition with real-world acquisition logic and producing complete annotations, this stage prepares the synthetic data for downstream use in tasks such as classification, object detection, and anomaly detection in industrial visual inspection.

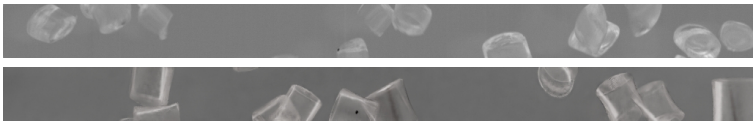
3.3.5 Results

Based on the modeling pipeline described in the previous sections, synthetic datasets were generated to match the characteristics of the real granule datasets introduced in Section 3.2. Two sets of synthetic images were created, each corresponding to the real granule images captured under a reflected-light or a transmitted-light setup.

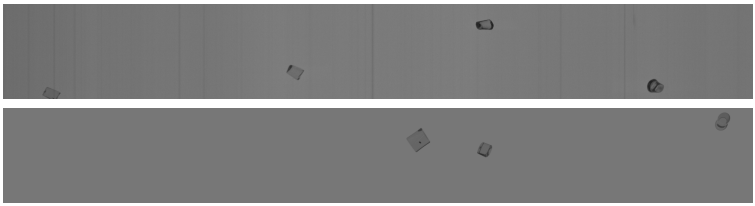
For the reflected-light configuration, a total of 400 raw synthetic images were rendered. These images are designed to reproduce the dense, overlapping distribution of granules as well as the defect patterns observed in the real images. Using the automated annotation procedure described in Section 3.3.4, individual granule crops were extracted and labeled to form the cropped dataset $\mathcal{D}_{\text{syn}}^{\text{rl}}$, containing 1000 normal and 200 defective granule samples.

For the transmitted-light configuration, 500 raw synthetic images were rendered. These images simulate sparse, well-separated granules on an illuminating slide. Following annotation, individual granule crops were extracted to form the cropped dataset $\mathcal{D}_{\text{syn}}^{\text{tl}}$, containing 520 normal and 520 defective granule samples.

The synthetic cropped datasets are intentionally sized to match the real training splits used in the downstream experiments rather than the full real cropped datasets. This design avoids using real validation or test images as style references during subsequent refinement, thereby reducing the risk of information leakage into the downstream evaluation. Consequently, $\mathcal{D}_{\text{syn}}^{\text{rl}}$ is limited to 1000 normal and 200 defective samples, matching the reflected-light real training split, while $\mathcal{D}_{\text{syn}}^{\text{tl}}$ is limited to 520 normal and 520 defective samples, matching the transmitted-light



(a) Real (top) and synthetic (bottom) raw granule images under a reflected-light setup



(b) Real (top) and synthetic (bottom) raw granule images under a transmitted-light setup

Figure 3.4 Comparison of real and synthetic raw images under reflected-light and transmitted-light setups. Each pair shows qualitative similarity in granule appearance, density, and illumination.

real training split. The remaining real crops are reserved for validation and testing in the downstream tasks described in Chapter 6.

Figure 3.4 presents visual comparisons between real and synthetic raw images under both reflected-light and transmitted-light illumination setups. The examples suggest that the rendered images capture key aspects of the structure, illumination, and defect patterns of the original datasets, while offering full control over annotation and class balance.

3.4 Prompt-Guided Image Generation via LLVMs

Generating synthetic defect images for metal surfaces presents specific challenges due to variations in crack shape, irregular crack boundaries, and heterogeneous surface texture. Instead of constructing explicit 3D models or relying on traditional generative frameworks, a prompt-guided image generation strategy based on LLVMs is adopted. In this setting, the task focuses not on synthesizing entire metal part images from scratch, but on modifying otherwise intact surfaces by synthesizing realistic crack structures through instruction-driven prompting.

The pipeline begins with a qualitative comparison of several LLVMs, each tested with a consistent set of prompts targeting localized crack synthesis. Once a suitable model is chosen, batch generation is performed and followed by post-processing to ensure spatial alignment and resolution consistency with the original images. To support downstream segmentation tasks, pixel-level crack annotations are produced using a semi-automatic labeling tool ISAT-SAM [127], followed by manual refinement to ensure accuracy and completeness.

This section outlines the full generation process, from model selection and prompt design to image processing and annotation, and presents the resulting synthetic dataset for use in crack detection and segmentation tasks.

3.4.1 Model Benchmarking and Selection

To identify a suitable model for prompt-guided crack synthesis, four LLVMs are qualitatively compared under identical conditions. A base prompt as follows is applied to each model:

"Add a tiny crack onto the metallic part in the reference image. Make sure to satisfy the following points when adding the crack: 1. the crack trajectory always starts from the edge of the metallic part; 2. the crack trajectory should look realistic and natural, and it should be relatively thin; 3. the other part of the image should remain unchanged."

This base prompt is designed to balance specificity and flexibility. Points 1 and 2 constrain the crack's geometric plausibility by enforcing its origin at the metal edge and requiring a realistic, thin trajectory of limited length. Point 3 ensures preservation of crack-free regions, which is critical for retaining the original surface texture and avoiding unnecessary alterations.

A set of reference images with varying surface textures and geometries is used to evaluate model robustness. Figure 3.5 presents the generated results. Qualitatively, GPT-image-1 produces the most photorealistic cracks among the tested models, with fine edges, realistic trajectories, and relatively consistent integration into the underlying surface texture while leaving the rest of the image largely unchanged. OmniGen2 generates plausible cracks in certain cases, but often introduces visible artifacts or unnatural texture distortions in surrounding areas. Grok struggles

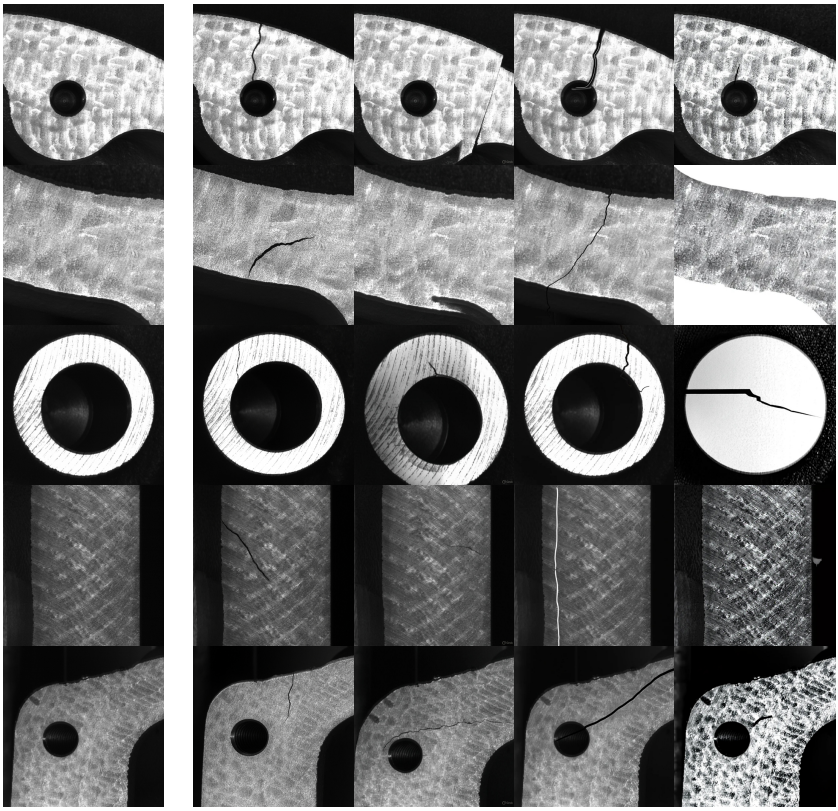


Figure 3.5 Qualitative comparison of different LLVMs on the same prompt and reference images. Columns: original image, GPT-image-1, Grok, Gemini, and OmniGen2. GPT-image-1 consistently produces realistic cracks while preserving the original metal surface appearance.

with shape preservation, sometimes altering the geometry of the metallic component itself in addition to adding a crack. Gemini is more consistent in maintaining the original image structure, but the cracks it produces tend to be overly simple and lack the fine irregularities characteristic of real defects, making them appear less realistic.

Based on this qualitative comparison, GPT-image-1 offers the best balance between defect realism, crack shape naturalness, and preservation

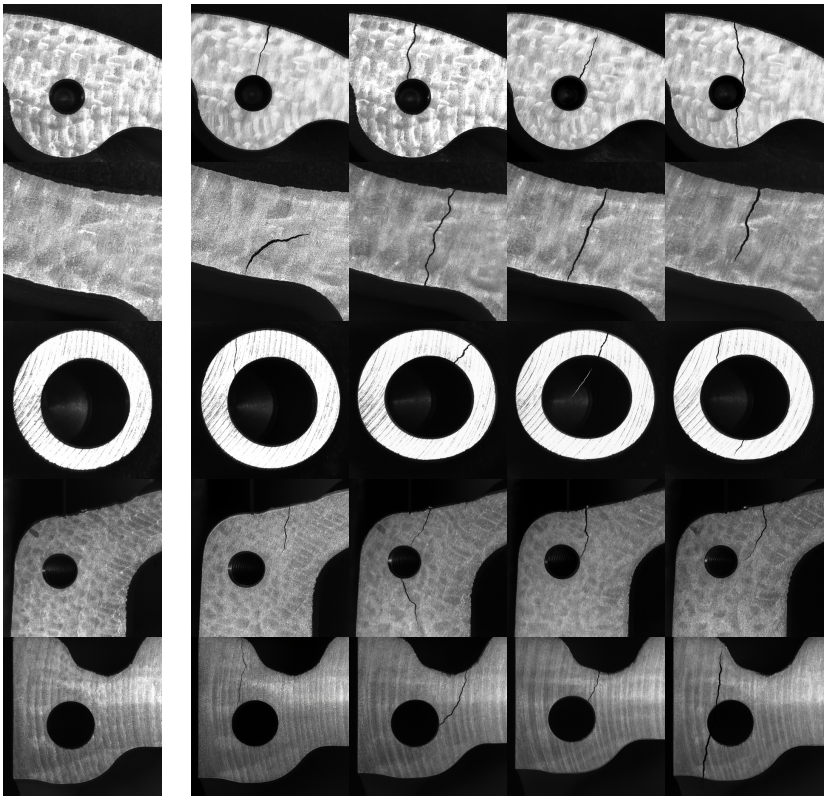


Figure 3.6 Stability assessment of GPT-image-1 across four independent generations for the same input image and prompt. Columns from left to right: original reference image followed by four independent generations. While general realism is preserved, variations in crack shape, crack location and fine structure reflect the stochastic nature of the model.

of the crack-free regions in the reference image, and is therefore selected for subsequent batch generation.

After choosing the GPT-image-1 model for the image synthesis, a stability experiment is conducted by repeatedly generating cracked versions of the same reference images using the identical prompt mentioned before in this section. The results, shown in Fig. 3.6, indicate that GPT-image-1 generally produces cracks with consistent starting points along the

metal edge and maintains realistic local texture blending. Meanwhile, noticeable variability exists in crack length, curvature, and branching for the same input and prompt, suggesting that stochastic elements in the generation process influence fine-scale morphology. This variation, while reducing repeatability, can be advantageous for dataset diversity by providing varied defect instances that remain visually plausible.

To further investigate the role of language prompts in guiding crack synthesis, we evaluate the effect of varying prompt complexity on generation quality. While the previous sections used a single standard prompt, here we test five different formulations—ranging from minimal instructions to detailed multi-constraint descriptions—on the same five reference images. This experiment allows a qualitative assessment of how prompt specificity affects controllability and visual quality, and whether more elaborate language improves visual realism or introduces unintended artifacts.

The five tested prompts are summarized below and referenced using short codes in the figure for clarity:

- **Prompt-1 (standard)**: *"Add a tiny crack onto the metallic part in the reference image. Make sure to satisfy the following points when adding the crack: 1. the crack trajectory always starts from the edge of the metallic part; 2. the crack trajectory should look realistic and natural, and it should be relative thin; 3. the other part of the image should remain unchanged."*
- **Prompt-2 (simplified)**: *"Add a thin and short crack to the metal part in the uploaded image. The crack should start from the edge. The rest of the image should not be changed."*
- **Prompt-3 (simplest)**: *"Put a small crack on the metal surface."*
- **Prompt-4 (focus-geometry)**: *"Generate an image by inserting a narrow, natural-looking crack into the uploaded image of a metal surface. The crack should originate from one edge of the metal part. It must be thin, and shaped realistically—no straight or artificial lines. All other regions of the image should remain visually identical to the original."*

- **Prompt-5 (focus-subtlety)**: *"Using the uploaded image of a metal part, insert a subtle and highly localized crack that begins at the edge. The added crack must be very narrow, matching the surrounding lighting and surface texture. Do not alter any other part of the image. "*

Figure 3.7 shows the qualitative results across all prompts. Each row corresponds to a reference image, and each column represents the result under a specific prompt from Prompt-1 to Prompt-5.

A qualitative analysis reveals several trends. Minimal prompts, *i.e.*, Prompt-2 and Prompt-3 often result in vague crack geometry or inconsistent placement. Some cracks appear disconnected from metal part edges, while others exhibit excessive thickness or visual noise. Across the shown examples, Prompt-1 produces realistic, edge-anchored cracks with relatively well-integrated textures and limited disturbance to crack-free areas. Detailed prompts with specific focuses, *i.e.* Prompt-4 and Prompt-5, improve crack realism and subtlety in some cases, but may also introduce exaggerated morphology or low-level blending artifacts due to over-specification. Overall, the comparison suggests that prompt specificity improves control over crack placement and appearance up to a certain point. Very simple prompts often lead to weak control over geometry and placement, whereas highly detailed prompts can improve crack realism but may also increase the risk of unintended artifacts or reduced output diversity. In the examined examples, the standard Prompt-1 provides the most balanced behavior and is therefore used for subsequent batch generation.

3.4.2 Batch Generation and Annotation

Following the model selection and prompt refinement, a dedicated pipeline was implemented to generate, align, and annotate synthetic images of metallic parts with surface cracks. The objective was to construct a dataset with high-resolution, visually realistic images and pixel-wise segmentation masks suitable for downstream training and evaluation tasks in crack detection.

Crack synthesis was performed in batch using the selected GPT-image-1 model via the OpenAI API. Each input consisted of a reference image of

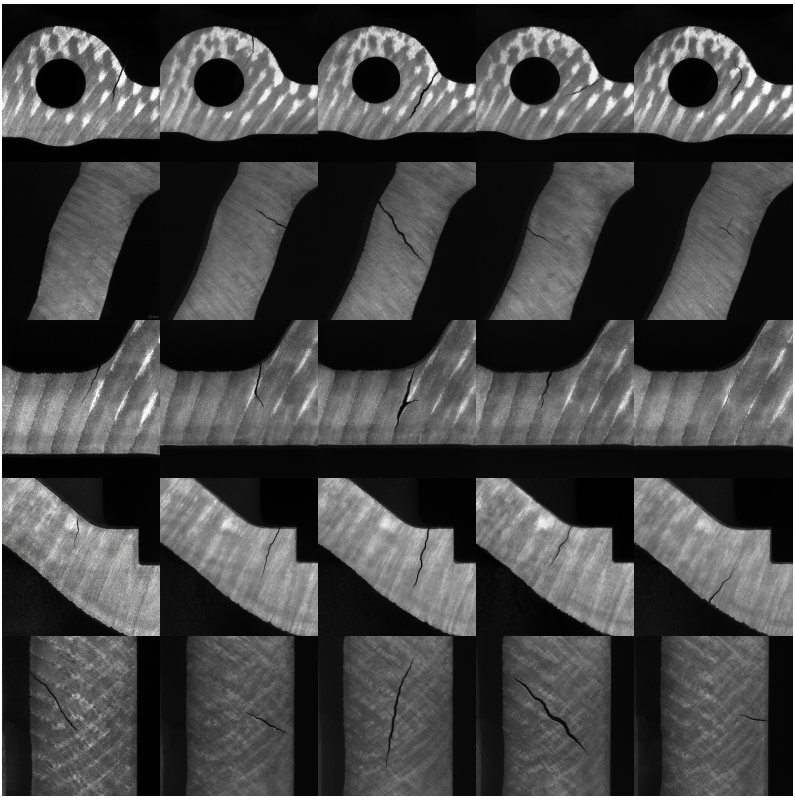


Figure 3.7 Effect of prompt formulation on crack generation. Each row corresponds to one reference image, and each column shows results under different prompt variants.

a normal metal part without a crack, together with the standard Prompt-1 in Section 3.4.1 specifying the shape and location of the added crack. The model produced outputs with a fixed resolution of 1024×1024 pixels.

Despite explicit prompt instructions to retain the non-defective areas, the generative process introduced spatial artifacts in some samples, including slight shifts, local warping, and padding effects. These artifacts led to geometric misalignment between the generated image and the original input, as illustrated in Fig. 3.8.

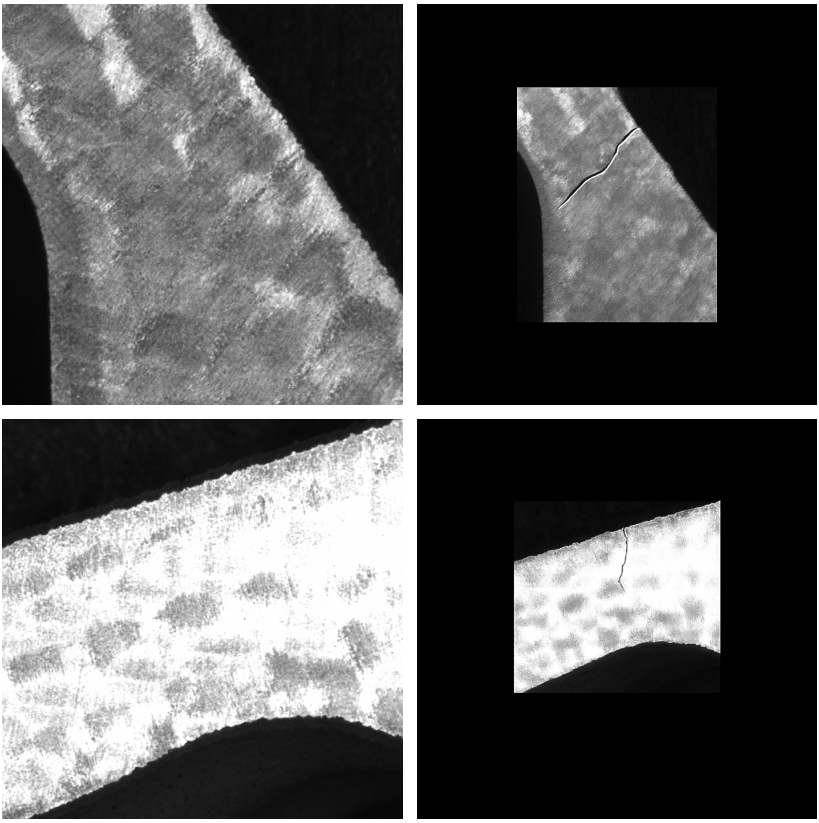


Figure 3.8 Illustration of generation-induced alignment artifacts. The reference images are shown in the left column, while the corresponding generated images from GPT-image-1 are shown in the right column, showing scaling, padding, geometric shifts at the edges, and textural changes.

To align each generated image with its corresponding reference, template matching was initially tested using the respective reference image as the template. However, it proved insufficient due to minor geometric and textural variations in the synthesized outputs. Thus, a more sophisticated registration strategy was developed. The alignment process begins by extracting adaptive masks of the metal part from both the reference and generated images, using the Canny edge detector followed by connected

component analysis to isolate the relevant region. These masks guide a multi-scale template matching stage, which operates on masked intensity profiles to estimate the initial spatial correspondence. When residual misalignments remain, the transformation is further refined through keypoint-based matching using Oriented FAST and rotated BRIEF (ORB) descriptors [229], followed by affine alignment. This combined procedure is intended to keep the synthetic crack correctly localized while restoring the spatial structure of the metal part as closely as possible to the original image.

The aligned synthetic images were then annotated using a semi-automatic segmentation framework based on the ISAT-SAM [127]. The annotation process leveraged the SAM-ViT-base backbone for interactive region proposals, enabling efficient segmentation of fine crack structures via click-and-hover operations. Nonetheless, each segmentation mask was visually reviewed and manually refined to ensure that each crack boundary is accurately captured, particularly in low-contrast or fine-grained regions.

3.4.3 Results

The resulting synthetic dataset $\mathcal{D}_{\text{syn}}^{\text{crack}}$ comprises 195 images of metallic parts with visually realistic synthetic cracks, each aligned to its corresponding real counterpart and accompanied by a manually validated pixel-wise segmentation mask. Analogous to the granule synthesis setup, crack synthesis uses only the 195 real crack-free images as references and excludes the 15 real defective crack images from the generation process to avoid information leakage into downstream evaluation. The resolution of the annotated images matches that of the original samples, ensuring compatibility for quantitative benchmarking. This dataset provides a labeled source for training and evaluating crack detection and segmentation models under controlled visual and geometric constraints that approximate the target domain. Representative examples of the synthesized images together with their real counterparts and pixel-level annotations are shown in Fig. 3.9.

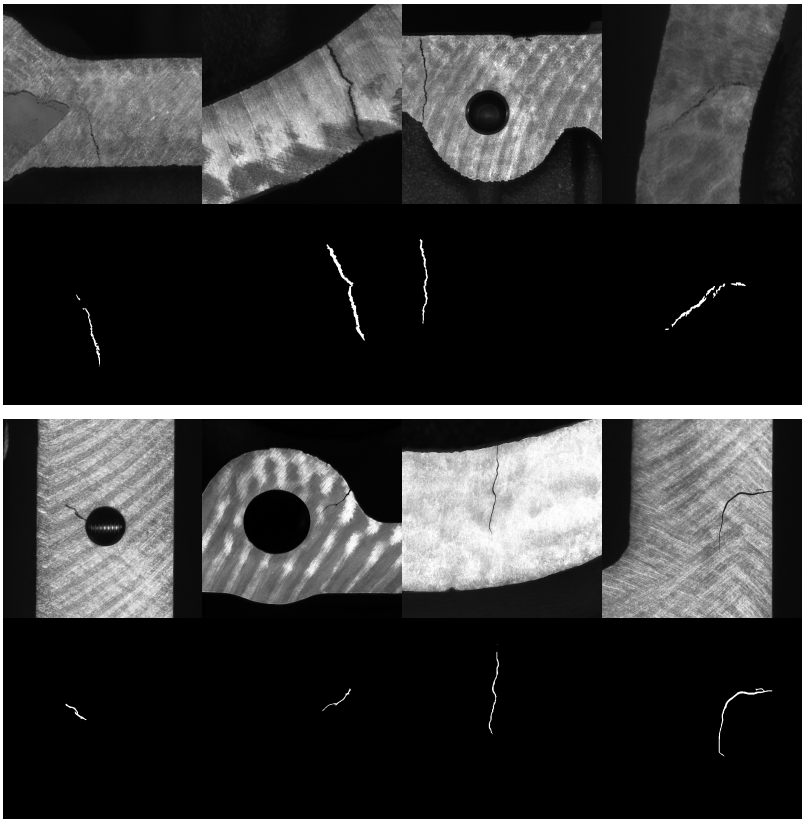


Figure 3.9 Examples from the crack datasets. The upper rows show examples of real crack images and their corresponding ground-truth masks, and the lower rows show examples of synthetic crack images and their corresponding ground-truth masks.

4 Feature-Consistent Synthetic Image Refinement

While synthetic images offer flexibility and scalability for dataset generation and augmentation, their visual fidelity often falls short of real-world imagery. Both the synthetic granule images and metal part images with synthetic cracks generated in Chapter 3 still exhibit visually recognizable shifts relative to the corresponding real data, as shown in Figs. 3.4 and 3.9. Previous studies have attempted to increase the fidelity of synthetic images across various vision tasks, but their effectiveness is often limited by a persistent domain gap. For example, Zhao *et al*. [319] showed that real-world clutter and novel structures, absent in synthetic training data, lead to poor generalization even with advanced image translation methods. Similarly, Peng *et al*. [205] demonstrated significant performance degradation in open-set classification and detection, highlighting the difficulty of adapting synthetic features to real distributions. More recently, Wang *et al*. [280] decomposed the synthetic-to-real domain gap into appearance, content, and quality gaps, and showed that synthetic data can remain substantially less effective than real data for downstream recognition despite high visual fidelity. Their analysis further identified insufficient content coverage as a major contributor to this performance gap. In this chapter, these limitations are addressed by applying style transfer–based refinement to both the synthetic granule dataset and the crack dataset, specifically aiming to enhance visual realism while maintaining feature consistency, which refers to the preservation of spatial position, geometry, and overall structure of discriminative features during refinement. Conventional generative models such as StyleGAN [130] are generally unsuitable for this objective due to their high stochasticity and lack of direct spatial alignment between input and output. The proposed methodology in this chapter emphasizes the careful selection of style-content image pairs to produce fine-grained refinements that

aim to preserve the diversity of the input data, in contrast to the more globally uniform translations often observed with the GAN-based baselines considered here. By constraining the refinement process in this way, realism can be enhanced while reducing the risk of altering object details needed for classification or defect detection, *e.g.*, improving illumination consistency, surface texture fidelity, and background integration. The refinement methodology is applied to both synthetic granule and synthetic crack datasets. The following sections present the selected refinement approaches, the experimental setup for their evaluation, and a comparative analysis of their effectiveness.

4.1 Method

As outlined in Chapter 2, neural style transfer (NST) was originally introduced by Gatys *et al*. [75] for transferring artistic styles between images. The method formulates image generation as an optimization problem in which the output image is encouraged to preserve the structural content of an input image while matching the style of a reference image through deep feature statistics represented by Gram matrices. Although initially developed for artistic synthesis, NST has also been applied to enhance the visual fidelity of synthetic images. For example, Dundar *et al*. [68] and Zaech *et al*. [305] modified the visual appearance of synthetic urban scenes to reduce texture bias and improve semantic segmentation on the Cityscapes dataset [55], either by transferring realistic styles or by diversifying textures through randomization. Ikeda *et al*. [120] extended this approach to instance-level style transfer, adapting rendered objects to real-world materials and lighting for improved 6D pose estimation. However, such methodologies do not transfer straightforwardly to industrial visual inspection, where refinement requires not only improved realism but also preservation of fine-grained and consistent object features. In this work, the baseline NST framework serves as the foundation for domain-specific refinements tailored to the granule and crack datasets.

4.1.1 Refinement of Synthetic Granule Images

The synthetic granule datasets $\mathcal{D}_{\text{syn}}^{\text{rl}}$ and $\mathcal{D}_{\text{syn}}^{\text{tl}}$, as described in Section 3.3.5, provide controlled training samples, but their visual fidelity remains limited compared to real-world images. A straightforward application of NST to entire synthetic scenes is suboptimal, since applying style transformations to large images containing multiple granules often distorts local details and introduces artifacts across object boundaries. Moreover, unlike the synthetic crack dataset $\mathcal{D}_{\text{syn}}^{\text{crack}}$ (cf. Section 3.4.3), the synthetic granules are modeled independently in Blender withouts correspondence to specific real granules. Consequently, direct pairwise matching between synthetic and real images is not practical.

To address these problems, an NST-based refinement framework is designed to enhance realism while ensuring that subtle but critical features, such as burn marks or surface irregularities, remain localized and intact. The refinement is performed on individual synthetic granule crops, which serve as content images for NST. Style images are sampled independently from the real granule dataset to enrich diversity while maintaining domain consistency.

To improve the initial compatibility between content and style pairs, a feature-based pairing strategy is introduced. For each synthetic granule crop, the most visually similar real granule image is identified by minimizing a combined discrepancy between their VGG19 [249] feature representations and Gram matrices. The feature similarity score uses the same relative weighting as the subsequent NST objective, with the style term weighted 10^6 times more strongly than the content term. This encourages the selection of a style image with comparable structure and texture to the synthetic target, providing a more coherent initialization for refinement. Assuming similar numbers of synthetic and real images, the time complexity of this similarity-based pairing strategy is $O(n^2)$. Fig. 4.1 shows two examples of feature-based pairing for selecting style images in NST.

Considering the quadratic time complexity of feature-based pairing, a simpler augmentation-based pairing strategy is developed. For each content image, the style image is first randomly sampled from the real domain and then geometrically aligned by rotation and flipping to better

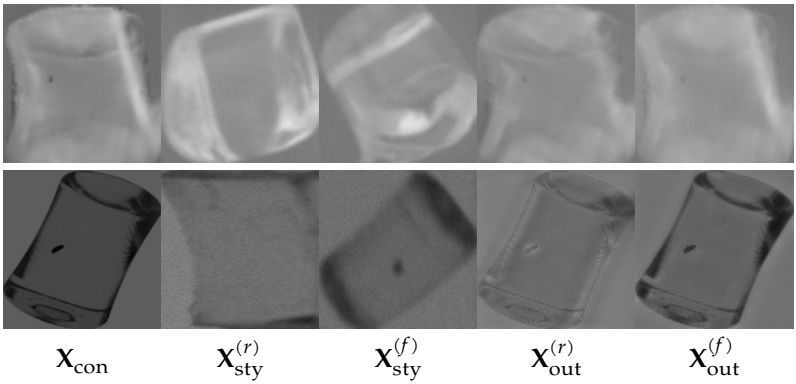


Figure 4.1 Effect of feature-based pairing before transfer. The two rows show examples from the granule datasets under reflected-light and transmitted-light illumination, respectively (*cf.* Section 3.2). The first column shows the synthetic content images. The second and third columns show randomly selected real style images and the most similar real style images selected by the initial content and style difference. The fourth and fifth columns show the corresponding transfer outputs using $\mathbf{X}_{\text{sty}}^{(r)}$ and $\mathbf{X}_{\text{sty}}^{(f)}$.

match the dominant edge flow and orientation of the content granule. These transformations preserve the aspect ratio and photometric properties while improving spatial alignment, thereby reducing artifacts such as ghost contours or repeated edge structures. Fig. 4.2 shows two examples of augmentation-based pairing. This feature-based pairing strategy reduces the time complexity from $O(n^2)$ to $O(n)$.

The complete procedure of augmentation-based pairing is summarized in Algorithm 1.

Both strategies aim to improve the compatibility between content and style images by reducing mismatches in appearance and orientation before refinement. In principle, they can be combined sequentially by first selecting a style image that is similar in feature space and then applying geometric alignment through rotation and flipping. However, such a combination does not necessarily improve the refinement result compared with augmentation-based alignment alone. One reason is that the feature-based pairing is performed on the original, unaligned images and may therefore favor a style image whose spatial layout is not well matched to the content image. In contrast, augmentation-based pairing

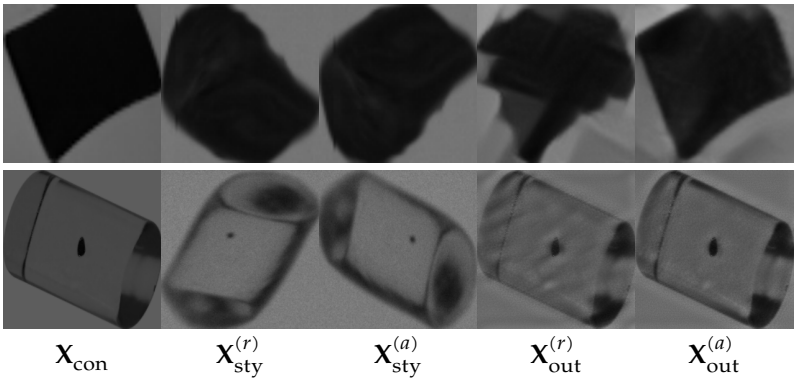


Figure 4.2 Effect of augmentation-based pairing before transfer. The two rows show examples from the granule datasets under reflected-light and transmitted-light illumination, respectively (*cf.* Section 3.2). The first column shows the synthetic content images. The second and third columns show randomly selected real style images and their optimally aligned versions based on the initial content loss. The fourth and fifth columns show the corresponding transfer outputs using $\mathbf{X}_{\text{sty}}^{(r)}$ and $\mathbf{X}_{\text{sty}}^{(a)}$.

explicitly searches over a small set of rotated and flipped variants and may therefore identify a better-aligned style image even when the initial image was selected at random. As a result, feature-based similarity and geometric alignment capture different aspects of image compatibility, and their combination does not guarantee a consistent advantage over using augmentation-based alignment alone.

4.1.2 Refinement of Synthetic Crack Images

Compared with synthetic granule images, the crack images differ in two crucial ways. First, each synthetic image is obtained by prompting the GPT-image-1 model to add a crack to a real reference image. In practice, the generated image often shows texture drift on the metal surface and small contour changes along worn edges, even when the prompt requires leaving all other pixels unchanged. Fig. 4.3 shows a few examples of these changes brought by the crack synthesis. Second, each generated crack image retains a direct correspondence to its crack-free reference image, which allows pairwise refinement and full-reference evaluation.

Algorithm 1 NST with augmentation-based pairing strategy

Input: content image $\mathbf{X}_{\text{con}}$, style image $\mathbf{X}_{\text{sty}}^{(r)}$, optimization steps T
Output: style-transferred output image $\mathbf{X}_{\text{out}}^{(a)}$

$\mathcal{S}_{\text{aug}} \leftarrow \{\}$ ▷ Set of augmented style candidates
for $d \in \{0°, 90°, 180°, 270°\}$ **do**
  for $f \in \{\text{True}, \text{False}\}$ **do**
    $\mathbf{X}_{\text{aug}} \leftarrow \text{RotateAndFlip}(\mathbf{X}_{\text{sty}}^{(r)}, d, f)$
    $\mathcal{S}_{\text{aug}} \leftarrow \mathcal{S}_{\text{aug}} \cup \{\mathbf{X}_{\text{aug}}\}$
  end for
end for
$\mathbf{X}_{\text{sty}}^{(a)} \leftarrow \text{None}$
$\mathcal{L}_{\text{con}}^{\min} \leftarrow \infty$
for $\mathbf{X}_{\text{aug}} \in \mathcal{S}_{\text{aug}}$ **do**
  $\mathcal{L}_{\text{con}} \leftarrow \mathcal{L}_{\text{con}}\left(\mathbf{X}_{\text{con}}, \mathbf{X}_{\text{aug}}\right)$ ▷ Initial content discrepancy
  if $\mathcal{L}_{\text{con}} < \mathcal{L}_{\text{con}}^{\min}$ **then**
    $\mathcal{L}_{\text{con}}^{\min} \leftarrow \mathcal{L}_{\text{con}}$
    $\mathbf{X}_{\text{sty}}^{(a)} \leftarrow \mathbf{X}_{\text{aug}}$
  end if
end for
Initialize $\mathbf{X}_{\text{out}}^{(a)} \leftarrow \mathbf{X}_{\text{con}}$
for $t = 1 \ldots T$ **do**
  $\mathbf{X}_{\text{out}}^{(a)} \leftarrow \text{NST}(\mathbf{X}_{\text{out}}^{(a)}, \mathbf{X}_{\text{sty}}^{(a)})$
end for
return $\mathbf{X}_{\text{out}}^{(a)}$

Applying standard neural style transfer to LLVM-generated crack images can improve the overall surface texture alignment with the original images. However, shortcomings arise in regions occupied by the synthetic cracks. Since the cracks cover only a small fraction of the image, the global style loss is dominated by the large crack-free regions, and optimization primarily focuses on aligning textures there. Consequently, the crack pixels are often corrupted, displaying unrealistic colors or blurred structures. Fig. 4.4 shows two examples of the artifacts introduced after applying standard NST for refinement.

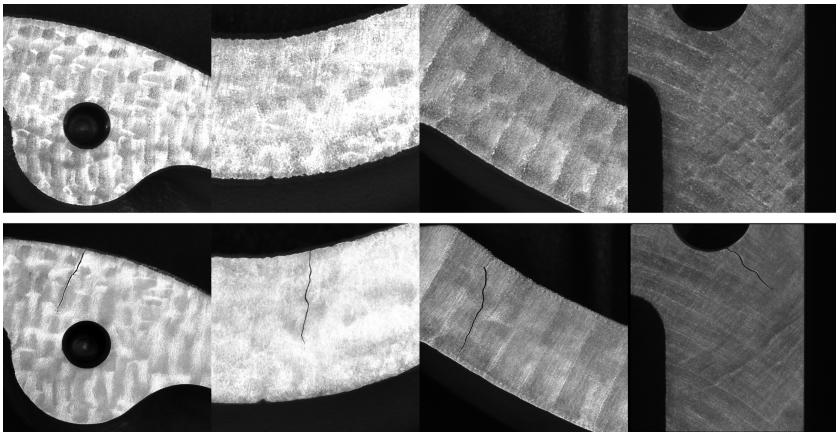


Figure 4.3 Examples of shifted texture and changed metallic part contours introduced by LLVM-based crack synthesis. The reference images are shown in the top row, while the LLVM-synthesized crack images are shown in the bottom row.

Attempts to alleviate this problem by freezing the crack pixels during backpropagation frequently destabilize the optimization and result in exploding loss values. Such instability is consistent with reports in the literature [65, 157, 181, 302] that freezing substantial portions of the model or input can disrupt gradient flow and lead to unstable optimization under gradient-based methods. Fig. 4.5 demonstrates an example of failed refinement.

To address these issues, a feature-consistent NST variant is developed, which preserves the crack region while preventing loss explosion. Let $\mathbf{X}_{\text{con}} \in \mathbb{R}^{H \times W \times 3}$ denote the LLVM-synthesized image containing a synthetic crack, used as the content image. Let $\mathbf{X}_{\text{sty}} \in \mathbb{R}^{H \times W \times 3}$ denote the

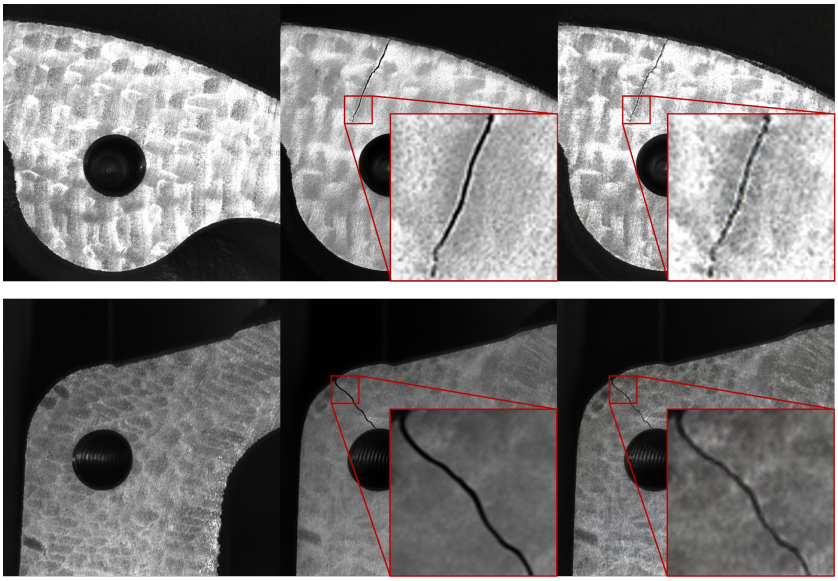


Figure 4.4 Examples of artifacts in crack regions of LLVM-synthesized crack image after applying standard NST. The first column shows the reference images. The LLVM-synthesized crack images, and their NST-refined versions are shown in the second and third columns, respectively, with details of their cracks.

crack-free reference image, and let $\mathbf{M} \in \{0,1\}^{H \times W}$ denote the corresponding binary crack mask. The objective function is defined as

$$\begin{aligned} \mathcal{L}_{\text{NST}}(\mathbf{X}_{\text{out}}) = \lambda_{\text{sty}} \sum_{l=1}^{L_s} \Big\| & \mathbf{G}^{(l)}(\mathbf{X}_{\text{out}} \odot (\mathbf{1} - \mathbf{M})) \\ & - \mathbf{G}^{(l)}(\mathbf{X}_{\text{sty}} \odot (\mathbf{1} - \mathbf{M})) \Big\|_F^2 \\ & + \lambda_{\text{con}} \sum_{l=1}^{L_c} \| F^{(l)}(\mathbf{X}_{\text{out}}) - F^{(l)}(\mathbf{X}_{\text{con}}) \|_2^2 \,. \end{aligned} \tag{4.1}$$

The refined image is obtained by

$$\mathbf{X}_{\text{out}}^{(c)} = \arg\min_{\mathbf{X}_{\text{out}}} \mathcal{L}_{\text{NST}}(\mathbf{X}_{\text{out}}) \,. \tag{4.2}$$

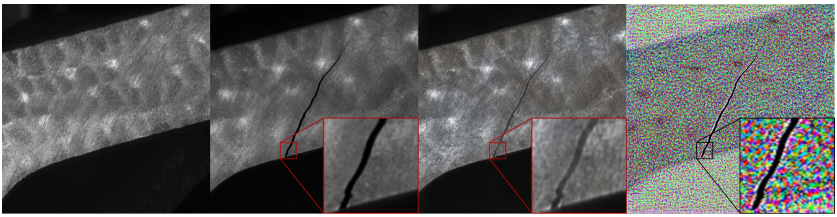


Figure 4.5 An example of failed refinement caused by the loss explosion during the NST optimization due to forcibly freezing pixels in crack region. From left to right: I. the reference image, II. LLVM-synthesized image with crack, III. refined synthetic image using standard NST, and IV. failed NST output due to loss explosion. Note that despite the high-frequency noise, the crack region is well-preserved.

To mitigate instability caused by region freezing during optimization, a soft-freeze mechanism is employed. A feathered mask $\widetilde{\mathbf{M}}$ is constructed, and both the feature projection and gradient update are performed using $\widetilde{\mathbf{M}}$ (*cf.* Algorithm 2). This relaxation reduces the hard discontinuity at the crack boundary, which otherwise can lead to sharp gradients and unstable updates. In practice, the soft mode with a feather radius of $r = 3\,\mathrm{px}$ was found to provide the most stable optimization while preserving crack morphology. The L-BFGS optimizer is further combined with a Wolfe line search and gradient clipping. The Wolfe conditions encourage each step to reduce the objective function while also satisfying curvature constraints, which prevents overly aggressive updates and enforces smoother convergence trajectories. Gradient clipping complements this mechanism by truncating large gradient values that may arise in early iterations, thereby reducing the risk of runaway steps that destabilize the optimization. Together, these techniques help reduce oscillations and the risk of divergence.

Meanwhile, the weighting of the loss terms is adjusted to emphasize structural preservation. The content weight λ_{con} is chosen to be comparatively larger than in granule refinement, since geometry and shading continuity near the crack are critical for downstream inspection tasks. This weighting discourages any geometric drift of the crack region while still permitting moderate textural adaptation of the surrounding surface. Overall, the combination of Soft-Freeze, carefully balanced loss terms, and stabilized optimization is intended to create a narrow transition

Algorithm 2 Feature-consistent NST with Soft-Freeze

Input: content image $\mathbf{X}_{\text{con}}$, style image $\mathbf{X}_{\text{sty}}$, crack mask $\mathbf{M}$, feather radius r, optimization steps T

Output: style-transferred output image $\mathbf{X}_{\text{out}}^{(c)}$

Initialize $\mathbf{X}_{\text{out}}^{(c)} \leftarrow \mathbf{X}_{\text{con}}$
$\widetilde{\mathbf{M}} \leftarrow \text{blur}(\text{dilate}(\mathbf{M}, r))$
for $t = 1 \dots T$ **do**
    $\mathbf{X}_{\text{out}}^{(c)} \leftarrow \widetilde{\mathbf{M}} \odot \mathbf{X}_{\text{con}} + (\mathbf{1} - \widetilde{\mathbf{M}}) \odot \mathbf{X}_{\text{out}}^{(c)}$
    Compute $\mathcal{L}_{\text{NST}}(\mathbf{X}_{\text{out}}^{(c)})$ based on Eq. (4.1)
    L-BFGS optimization step with Wolfe line search and clip gradients
end for
return $\mathbf{X}_{\text{out}}^{(c)}$

band around the crack, improve optimization smoothness, and enhance visual continuity. The methodology is summarized in Algorithm 2.

For comparison, a naive alternative is adopted. First, standard NST is applied on $\mathbf{X}_{\text{con}}$ to obtain a refined image $\mathbf{X}_{\text{out}}$, then the crack region from $\mathbf{X}_{\text{con}}$ is copied onto $\mathbf{X}_{\text{out}}$ with smooth alpha blending. For a fair comparison, a feather with $r = 3$ px is applied for the crack region. This process requires no modification of the NST optimization and only the binary mask $\mathbf{M}$.

$$\mathbf{X}_{\text{out}}^{(\text{p})} = \widetilde{\mathbf{M}} \odot \mathbf{X}_{\text{con}} + (\mathbf{1} - \widetilde{\mathbf{M}}) \odot \mathbf{X}_{\text{out}}\,. \tag{4.3}$$

The image $\mathbf{X}_{\text{out}}^{(\text{p})}$ denotes the standard-NST-refined result with the crack region copy-pasted from the original LLVM-synthesized image.

This strategy preserves the original crack pixels inside the masked region and reduces color leakage inside the crack while still transferring texture, illumination, *etc*. in the crack-free regions. Fig. 4.6 visualizes the output images refined with NST and its variants.

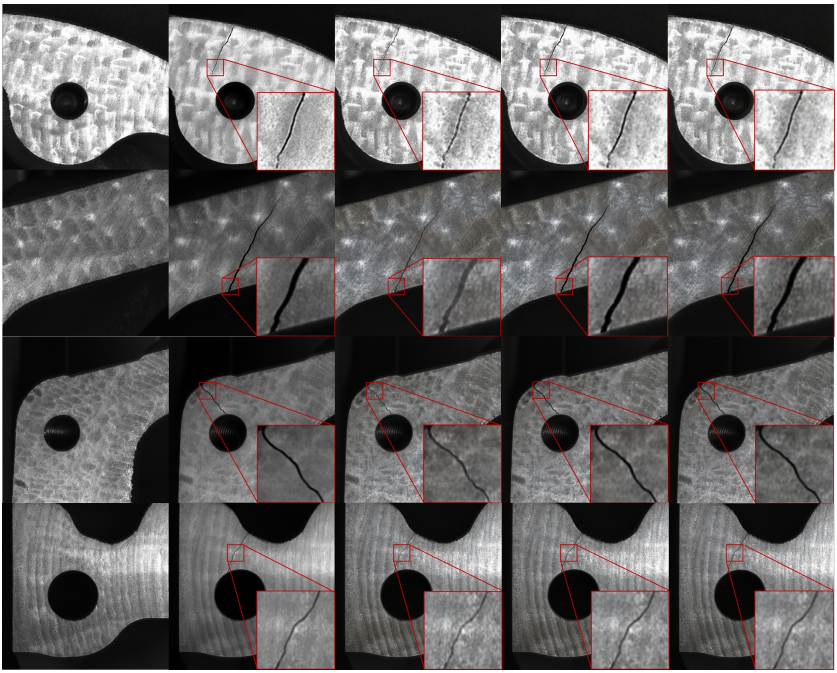


Figure 4.6 Examples showing the comparison of different NST-based refinement methodologies. From left to right: I. reference image, II. synthesized crack image, III. standard NST-refined image, IV. standard NST result with the original crack region from II copied and blended back into the image, and V. feature-consistent NST with Soft-Freeze.

4.2 Evaluation of Refined Images

The goal of this section is to evaluate the effectiveness of the proposed refinement strategies for both synthetic granules and metallic crack images. The evaluation considers two complementary perspectives. For granule datasets, which lack one-to-one correspondence between synthetic and real samples, distribution-level metrics are employed to measure the alignment of refined images with real data. For metallic cracks, where each synthetic sample is generated directly from a real counterpart, pairwise similarity measures are used to quantify the preservation of structural and textural fidelity. To provide a fair comparison, two image-

to-image translation methods, CycleGAN and Contrastive Unpaired Translation (CUT), are included as baselines alongside the proposed refinement. For the granule datasets, all synthetic and refined images are evaluated against matched real subsets reserved for downstream training, while for the crack dataset, all 195 synthetic and refined crack images are evaluated against their corresponding crack-free real reference images. Results are analyzed both quantitatively and qualitatively.

4.2.1 Evaluation Setup

The evaluation is performed on the synthetic datasets described in Sections 3.3.5 and 3.4.3. For the granule datasets, the full synthetic sets are used, containing 1200 cropped images for $\mathcal{D}^{\text{rl}}_{\text{syn}}$ and 1040 cropped images for $\mathcal{D}^{\text{tl}}_{\text{syn}}$. To compute distribution-level metrics with matched sample sizes, corresponding real subsets are drawn from the portions of $\mathcal{D}^{\text{rl}}_{\text{real}}$ and $\mathcal{D}^{\text{tl}}_{\text{real}}$ reserved for downstream training, containing 1200 and 1040 cropped images, respectively. This keeps the refinement evaluation consistent with the downstream protocol and avoids using real validation or test images as references during refinement. The corresponding refined datasets have the same number of images as their synthetic source datasets. For the metallic part images, each image set contains 195 samples, covering the original real images, the LLM-synthesized crack images, and the corresponding refined crack images produced by different methods.

Granule images are resized to 256 × 256 pixels, while the metallic part images are resized to 512 × 512 pixels prior to evaluation. For granule images, comparison is made between the real and synthetic datasets as distributions. Following standard practice, Fréchet Inception Distance (FID) is computed on 2048-dimensional pool3 features of InceptionV3 model [105, 260], and maximum mean discrepancy (MMD) is evaluated using radial basis function (RBF) kernels on flattened image tensors. Both metrics are unitless numerical discrepancy measures, which capture the fidelity and distributional similarity between real and refined granules without requiring one-to-one correspondences. More details about FID and MMD are provided in Appendix A.1.

In addition to distribution-level fidelity measures between image sets, an *intra-set pairwise perceptual loss* is defined to quantify the internal variety within an image set by jointly considering both content and style discrepancies.

Let $\mathcal{D} = \{\mathbf{X}^i\}_{i=1}^N$ denote a set of images. The expected value and standard deviation of the pairwise perceptual loss over all unordered image pairs are defined as

$$\begin{aligned} \mu_{\text{ppl}}(\mathcal{D}) &= \mathbb{E}_{i<j}\left\{\mathcal{L}_{\text{ppl}}(\mathbf{X}^i, \mathbf{X}^j)\right\}, \\ \sigma_{\text{ppl}}(\mathcal{D}) &= \sqrt{\text{Var}_{i<j}\left\{\mathcal{L}_{\text{ppl}}(\mathbf{X}^i, \mathbf{X}^j)\right\}}. \end{aligned} \tag{4.4}$$

where $\mathcal{L}_{\text{ppl}}(\mathbf{X}^i, \mathbf{X}^j)$ denotes a weighted perceptual loss between two images:

$$\mathcal{L}_{\text{ppl}}(\mathbf{X}^i, \mathbf{X}^j) = \lambda_{\text{con}}\, \mathcal{L}_{\text{con}}(\mathbf{X}^i, \mathbf{X}^j) + \lambda_{\text{sty}}\, \mathcal{L}_{\text{sty}}(\mathbf{X}^i, \mathbf{X}^j). \tag{4.5}$$

Here, $\mathcal{L}_{\text{con}}(\mathbf{X}^i, \mathbf{X}^j)$ and $\mathcal{L}_{\text{sty}}(\mathbf{X}^i, \mathbf{X}^j)$ denote the content and style discrepancies, respectively, computed using VGG-based feature representations as described in Section 2.4.3.

The pairwise perceptual loss $\mathcal{L}_{\text{ppl}}(\mathbf{X}^i, \mathbf{X}^j)$ thus captures both the stylistic and structural discrepancies between image pairs. The weights of the content and style terms are identical to those used in NST, maintaining sensitivity to stylistic diversity while ensuring structural coherence. The mean and standard deviation are then computed over all unordered pairs ($i < j$) within $\mathcal{D}$, providing a robust estimate of the intra-set variation. Gram-based representations capture global style statistics such as color tone, illumination, and texture co-occurrence while being largely agnostic to spatial layout. By combining them with a lightly weighted content term, the proposed metric reflects both stylistic dispersion and coarse structural consistency within a set, thereby offering a perceptually balanced measure of internal diversity rather than cross-domain discrepancy as in FID or MMD. The intra-set expectation μ_{ppl} indicates the overall level of perceptual diversity, whereas σ_{ppl} quantifies the heterogeneity of that diversity across sample pairs. This formulation aligns with the NST objective by combining content and style discrepancies over image pairs, enabling systematic evaluation of image sets while maintaining perceptual relevance.

In contrast, for the crack images, a strict correspondence between the original real image and its LLVM-generated crack variant with or without refinement is preserved. Accordingly, similarity is assessed on a per-image basis. Each refined crack image is compared against its corresponding original using mean squared error (MSE), structural similarity index measure (SSIM), peak signal-to-noise ratio (PSNR), and learned perceptual image patch similarity (LPIPS). All metrics are averaged across the dataset. More details about these metrics are provided in Appendix A.2. The variety measure using the defined intra-set pairwise perceptual loss is not applied for the metallic crack data, since all refined subsets are derived from the same original images.

Two established image-to-image translation frameworks, CycleGAN [322] and Contrastive Unpaired Translation [202], are also included in the evaluation for comparison. Both models are trained separately for each domain pair using the official implementations. For granules, both batches captured under different illumination setups are trained independently using both frameworks. For crack images, training is performed on the 195 LLVM-generated crack images and their corresponding original crack-free counterparts. Although these images are strictly paired, the models are trained in unpaired mode to prevent trivial overfitting. Training is carried out for 120 epochs, using the default optimizer and loss settings. After training, inference is run on the full synthetic datasets at 512×512 pixels resolution, producing refined images that are then evaluated with the same metrics as the proposed refinement approach. This setup enables a direct comparison between feature-consistent style transfer and adversarial domain translation in terms of both fidelity and structural preservation.

4.2.2 Evaluation of Fidelity

High-fidelity synthetic data can improve the generalization of models trained on these images to real inspection scenarios, as discrepancies in texture, illumination, or shading can lead to domain-specific overfitting. To this end, the synthetic images are quantitatively evaluated against their respective real references using perceptual and distributional metrics tailored to each dataset. The following subsections present separate analyses for $\mathcal{D}_{\text{syn}}^{\text{rl}}$, $\mathcal{D}_{\text{syn}}^{\text{tl}}$, and for the LLVM-based crack images.

Table 4.1 Quantitative evaluation of fidelity for synthetic granule images in $\mathcal{D}_{\text{syn}}^{\text{rl}}$.

Method	FID↓	MMD↓
Original synthetic (no refinement)	87.68	0.3571
NST	87.67	0.3207
CycleGAN [322]	90.03	0.3424
CUT [202]	86.93	0.3101
NST (feat)	83.49	0.3163
NST (aug)	**82.33**	**0.3098**
NST (feat + aug)	82.65	0.3110

4.2.2.1 Synthetic Granule Images in $\mathcal{D}_{\text{syn}}^{\text{rl}}$

As mentioned in Section 4.2.1, the fidelity of the refined synthetic images in $\mathcal{D}_{\text{syn}}^{\text{rl}}$ is quantitatively assessed using the FID and MMD between real and synthetic image distributions. Lower values of both metrics indicate higher similarity to the real data in terms of feature statistics and pixel-wise distribution, respectively. Table 4.1 summarizes the results across all compared methods, including the original unrefined synthetic images, the proposed feature-consistent refinement, and the GAN-based baselines. The feature-based and augmentation-based pairing strategies are introduced in Section 4.1.1.

The original synthetic set shows the largest MMD and also has a relatively high FID, although CycleGAN yields an even higher FID in this setting. All refinement methods reduce MMD relative to the unrefined synthetic data, indicating improved distributional alignment in pixel space. Among the compared methods, NST with augmentation-based pairing yields the lowest values for both FID and MMD, although the differences to NST with feature-based pairing and to the combined variant remain small. These results therefore suggest that augmentation-based alignment is a useful component in this setting, but they do not indicate a clear separation among the best-performing NST variants. In contrast, standard NST and CycleGAN do not improve FID relative to the original synthetic data, whereas CUT achieves a moderate reduction. Overall, the results are consistent with the discussion in Section 4.1.1, *i.e.*, combining

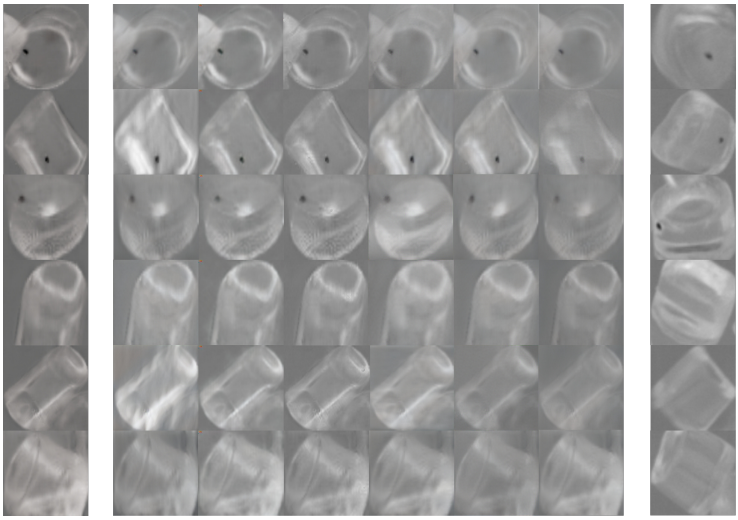


Figure 4.7 Qualitative results of refinement for images in $\mathcal{D}^{\mathrm{rl}}_{\mathrm{syn}}$. The real granule images in the rightmost column are manually selected from $\mathcal{D}^{\mathrm{rl}}_{\mathrm{real}}$, which are visually similar to the corresponding synthetic granule images in the same rows. These real images are used only for comparison, and are not the style images used for NST.

feature-based pairing with augmentation-based pairing does not provide a systematic advantage over augmentation-based pairing alone.

The qualitative comparison in Fig. 4.7 provides further insight into the differences among refinement strategies. The original synthetic granules exhibit the most pronounced fine details due to their high-resolution rendering, yet these details appear overly sharp compared to the real granules, whose surface texture is naturally smoother. In addition, the illumination and overall color tone of the synthetic images differ noticeably from the real samples. The standard NST effectively adjusts the color tone and illumination, aligning them more closely with real data. However, this benefit comes at the cost of visible artifacts in certain cases, such as vague edges and local texture inconsistencies. Both CycleGAN and CUT generate visually coherent and natural results, with improved tonal balance and surface shading, though the retained fine details remain

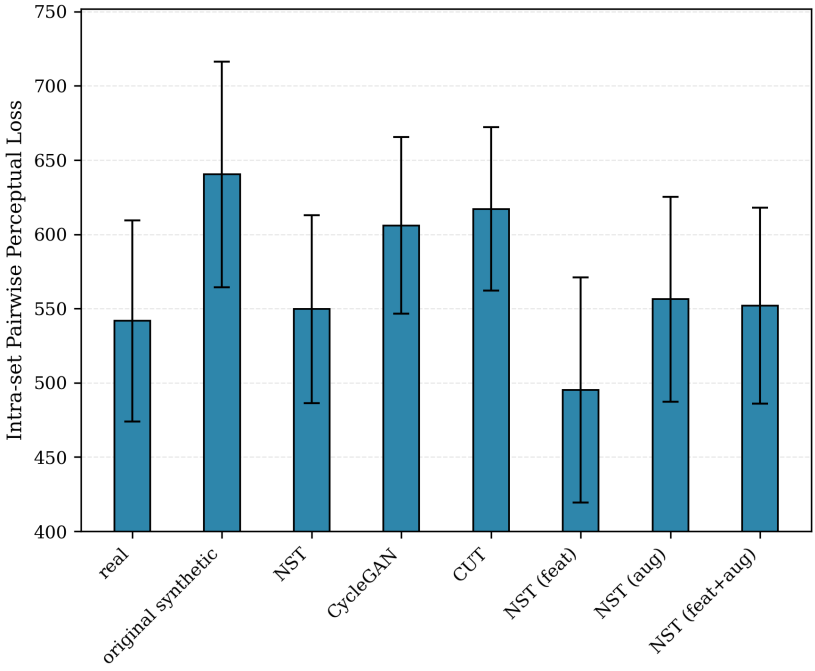


Figure 4.8 Intra-set pairwise perceptual loss illustrating the internal content and style diversity of $\mathcal{D}^{\text{rl}}_{\text{real}}$, original $\mathcal{D}^{\text{rl}}_{\text{syn}}$ and refined $\mathcal{D}^{\text{rl}}_{\text{syn}}$ using different methods.

slightly exaggerated relative to the real samples. Among the proposed NST variants, the visual outcomes differ subtly: the similarity-based pairing produces realistic textures but with occasional uneven contrast, while the combined similarity-and-augmentation pairing enhances consistency but marginally reduces sharpness. The NST with augmentation only appears to yield the most balanced appearance among the shown examples, closely resembling real granules in color tone and texture while avoiding structural distortions or artifacts.

The intra-set pairwise perceptual loss in Fig. 4.8 is used here as a proxy for the internal diversity of each image set, combining style-related differences with a weaker sensitivity to structural variation. Under this measure, the real images show a moderate level of dispersion, whereas

Table 4.2 Quantitative evaluation of fidelity for synthetic granule images in $\mathcal{D}_{\text{syn}}^{\text{tl}}$.

Method	FID↓	MMD↓
Original synthetic (no refinement)	140.69	0.4062
NST	129.19	0.3805
CycleGAN [322]	**70.04**	**0.2453**
CUT [202]	110.78	0.3543
NST (feat)	108.94	0.3523
NST (aug)	113.01	0.3594
NST (feat + aug)	112.77	0.3504

the original synthetic images exhibit substantially larger variation. This indicates that the unrefined synthetic set is more heterogeneous than the real data under the chosen perceptual metric. All refinement methods reduce this dispersion, suggesting that refinement makes the synthetic images more internally consistent. Among the refined variants, standard NST and NST with feature-and-augmentation-based pairing remain closer to the real set than the original synthetic data while still preserving a non-negligible level of internal variation. In contrast, NST with feature-based pairing alone yields the lowest loss values overall, which may indicate a stronger reduction of diversity. Taken together, these results suggest that the NST-based variants differ not only in fidelity to the real domain but also in how strongly they compress the internal variability of the synthetic set.

4.2.2.2 Synthetic Granule Images in $\mathcal{D}_{\text{syn}}^{\text{tl}}$

The fidelity of the synthetic granule images in $\mathcal{D}_{\text{syn}}^{\text{tl}}$ is evaluated in the same way as for $\mathcal{D}_{\text{syn}}^{\text{rl}}$. Table 4.2 summarizes the obtained FID and MMD values across all refinement and baseline methods.

Compared with the results in Table 4.1, the differences among methods are more pronounced for $\mathcal{D}_{\text{syn}}^{\text{tl}}$, especially in terms of FID. The original synthetic images again show the largest discrepancy from the real distribution in both metrics. All refinement methods reduce FID and MMD relative to the unrefined synthetic set, indicating improved distribution-

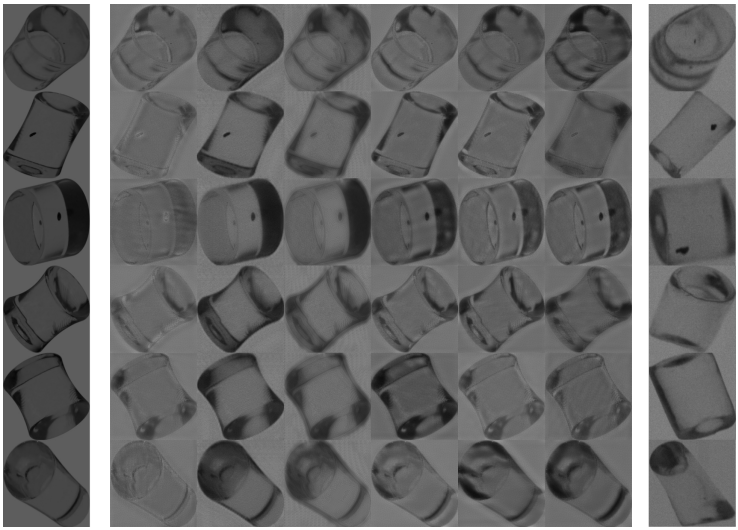


Figure 4.9 Qualitative results of refinement for images in $\mathcal{D}_{\text{syn}}^{\text{tl}}$. The real granule images in the rightmost column are manually selected from the real domain, which are visually similar to the corresponding synthetic granule images in the same rows. These real images are used only for comparison, and are not the style images used in NST.

level alignment. Among the evaluated methods, CycleGAN achieves the lowest values for both metrics by a clear margin. The remaining methods, including standard NST, CUT, and the three NST variants with feature-based and/or augmentation-based pairing, also improve over the original synthetic images but remain noticeably behind CycleGAN in these numerical measures. At the same time, these metrics reflect only distributional similarity and do not by themselves determine whether the refined images appear more realistic or preserve the relevant granule characteristics more faithfully. For this reason, the quantitative results are interpreted together with the qualitative comparison discussed below.

The qualitative comparison in Fig. 4.9 illustrates the visual differences among the refined images under transmitted-light setup.

The original synthetic images display a homogeneous appearance with highly similar illumination and contrast across samples, resulting from

the fixed rendering parameters used in their generation. After refinement, the standard NST introduces a more realistic tonal variation and slightly softens the excessive contrast, but in some cases the surface texture becomes overly smooth. CycleGAN and CUT further diversify the global tone and brightness while maintaining consistent geometry, producing visually coherent results with balanced shading. However, their outputs exhibit a tendency toward a common color palette and moderate local contrast, leading to a perceptually uniform style across samples. In comparison, the NST variants based on feature-, augmentation-, and combined pairing strategies achieve a broader range of illumination and reflectance conditions while preserving the granules' structural integrity. The feature-based pairing preserves clean surface details while maintaining stable geometry, whereas the augmentation-based variant introduces modest variations in surface appearance, particularly in local intensity and light transmission, thereby better reflecting the variability observed in real data. Overall, the refinement methods introduce a broader range of visual variation than the original synthetic set, with the NST-based approaches offering a balanced compromise between realistic variability and structural consistency.

The intra-set pairwise perceptual loss in Fig. 4.10 is overall higher than in Fig. 4.8, indicating that the granule set under transmitted-light illumination exhibits greater internal variation under this metric. The original synthetic images again show the largest dispersion among the compared sets. After refinement, both CycleGAN and CUT yield comparatively low average losses, which is consistent with the more visually uniform appearance observed in their refined outputs. In contrast, NST, NST with augmentation-based pairing, and NST with feature-and-augmentation-based pairing remain closer to the real set in terms of the mean perceptual loss. Under this measure, these variants therefore appear to preserve a level of intra-set variation that is more similar to that of the real data, whereas the adversarial baselines compress the distribution more strongly.

4.2.2.3 Synthetic Crack Images

The fidelity of different synthetic crack image sets is evaluated using full-reference similarity measures computed between each refined image

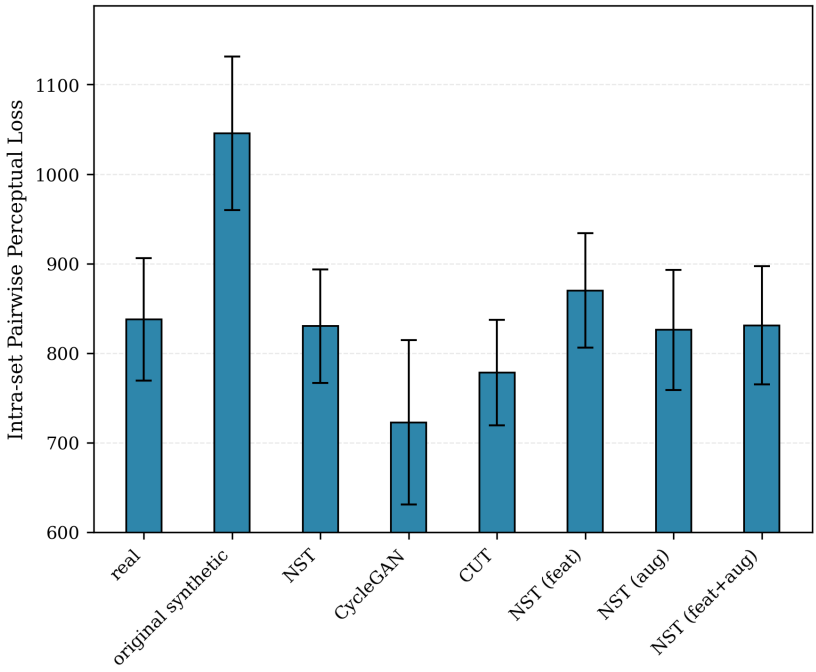


Figure 4.10 Intra-set pairwise perceptual loss illustrating the internal content and style diversity of $\mathcal{D}^{\text{tl}}_{\text{real}}$, original $\mathcal{D}^{\text{tl}}_{\text{syn}}$ and refined $\mathcal{D}^{\text{tl}}_{\text{syn}}$ using different methods.

and its corresponding original, as described in Section 4.2.1. All values are averaged across 195 images, and lower MSE and LPIPS together with higher PSNR and SSIM indicate higher perceptual similarity to the reference images.

The quantitative results in Table 4.3 show that the NST-based refinements consistently achieve higher similarity to the reference images than the original synthetic data and the GAN-based baselines. Standard NST attains the best results in terms of MSE, PSNR, and LPIPS, whereas the Copy–Paste and Soft-Freeze variants yield comparable values. The Soft-Freeze variant also achieves the highest SSIM. Taken together, these results indicate that all three NST-based variants preserve the overall appearance of the reference images more effectively than CycleGAN and

Table 4.3 Quantitative evaluation of fidelity for synthetic crack images.

Method	MSE↓	PSNR↑	SSIM↑	LPIPS↓
Original synthetic (no refinement)	0.0125	19.7386	0.4010	0.3018
NST	**0.0122**	**19.9745**	0.4323	**0.2182**
CycleGAN [322]	0.0147	19.3635	0.4282	0.2807
CUT [202]	0.0144	19.4387	0.3918	0.3074
NST + Copy–Paste	0.0124	19.9083	0.4321	0.2240
NST + Soft-Freeze	0.0123	19.9330	**0.4326**	0.2243

CUT under the selected full-reference metrics. At the same time, the numerical differences among the NST-based methods remain small, so these metrics alone do not allow a definitive distinction between the variants in terms of their suitability for downstream use. In particular, visual inspection shows that standard NST can still introduce local artifacts in the crack regions, such as blurred or discolored pixels (*cf.* Figs. 4.4 and 4.6), even when its global similarity scores are the most favorable. By contrast, the Copy–Paste and Soft-Freeze variants suppress such artifacts more effectively and therefore provide a more reliable preservation of crack structure in qualitative terms.

The qualitative comparison in Fig. 4.11 provides a detailed view of the visual characteristics of the refined crack images produced by the different methods. The original synthetic images contain the crack structures based on the reference images, but their surface texture deviates notably from the original metallic appearance. Applying the standard NST transfers the reference texture onto the synthetic images and thereby improves the surface realism, yet it introduces artifacts within the crack regions, including colored noise and local blurring that distort the defect boundaries. CycleGAN and CUT produce smoother and visually coherent outputs but substantially alter the material texture, leading to surfaces that appear more uniform and less detailed than in the original. This behavior can be attributed to the global nature of adversarial learning, where the generator learns an averaged texture representation across the entire training set rather than a localized one-to-one correspondence as in NST. In contrast, the Copy-Paste and Soft-Freeze variants preserve the correct crack geometry and effectively suppress artifacts.

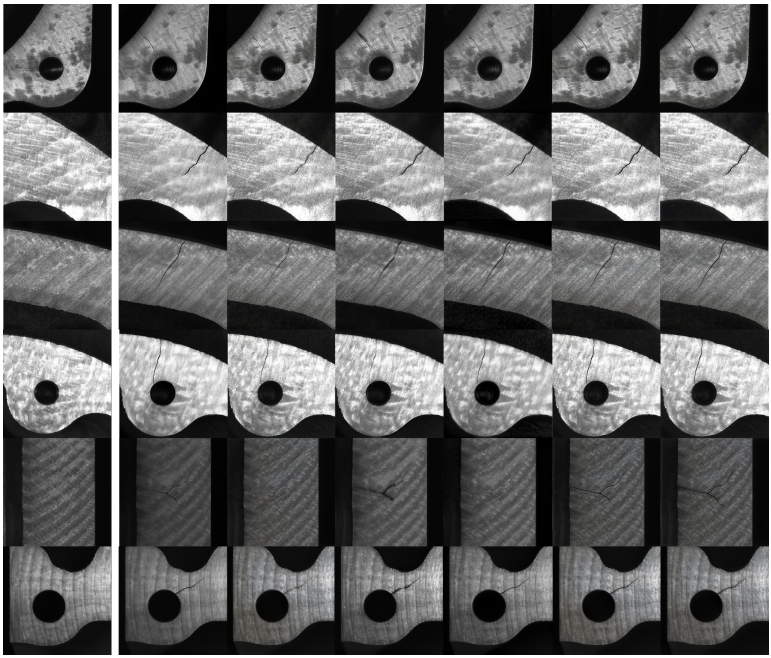


Figure 4.11 Qualitative results of refinement for crack images. The original reference images without crack are shown in the leftmost column.

The Copy-Paste variant tends to preserve the crack region sharply but sometimes introduces minor zigzag patterns along the boundaries, while the Soft-Freeze variant appears to yield the most consistent and realistic results among the shown examples, maintaining both smooth defect contours and surface texture continuity. Overall, the modified NST variants achieve a balance between accurate crack synthesis and faithful texture reproduction, avoiding the uniform appearance and texture loss observed in the GAN-based refinements.

4.3 Conclusion

This chapter presented a feature-consistent refinement framework for synthetic images based on neural style transfer. Feature- and augmentation-based pairing improved granule refinement by aligning structure and orientation, while the Copy–Paste and Soft-Freeze variants for crack images protected defect geometry and removed artifacts. Quantitative and qualitative evaluations showed that the proposed refinements improve fidelity relative to the original synthetic data and provide competitive alternatives to the GAN-based baselines, particularly when structural consistency is prioritized. Although the proposed NST-based refinements require iterative optimization, they offer interpretable and visually consistent results, providing refined datasets for the downstream classification, segmentation, and anomaly-detection experiments in Chapter 6.

5 Training Stability and Early Stopping for Industrial Visual Anomaly Detection Models

This chapter introduces a stability-guided early stopping strategy for industrial visual anomaly detection models. The focus is on detecting convergence without labeled validation data by monitoring the temporal stability of learned feature embeddings during training. The method is evaluated primarily within anomaly detection pipelines, yet its formulation is general and applicable to any encoder-based model trained in a validation-free setting. The developed strategy provides a unified experimental protocol for fair and reproducible model selection, which is used in this chapter for anomaly detection benchmarks and later applied to the downstream tasks evaluated in Chapter 6, including models trained on the refined synthetic images introduced in Chapter 4.

5.1 Introduction

The previous chapters established a complete pipeline for generating and refining synthetic data tailored to industrial inspection scenarios. While these synthetic datasets provide visually plausible and structure-preserving samples, the reliable training of downstream visual models remains a separate and equally critical challenge, particularly in anomaly detection. Methods such as GLASS [44] and SimpleNet [162] operate without labeled validation data, and their training losses exhibit non-monotonic behavior with frequent fluctuations that prevent straightforward convergence monitoring. As a consequence, existing approaches often rely on fixed training lengths or use test data as a surrogate for

checkpoint selection, which introduces information leakage and compromises the comparability of results.

This chapter addresses the problem of validation-free model selection by introducing an early stopping strategy based on the temporal stability of learned feature embeddings. Instead of relying on loss values or performance indicators, the proposed criterion measures the consistency of image embeddings across consecutive epochs and halts training once these representations stabilize. The resulting strategy is fully unsupervised, computationally lightweight, and compatible with a broad range of encoder-based architectures.

The remainder of this chapter formalizes the problem setting, introduces the proposed embedding-stability measure, describes its integration into existing anomaly detection frameworks, and evaluates its effectiveness on standard benchmarks.

5.2 Limitations of State-of-the-Art Methods

Training industrial visual anomaly detection models is typically performed without labeled validation data. This constraint originates from both the structure of widely used benchmark datasets and the characteristics of real industrial inspection processes. Public datasets such as MVTec AD [18] and VisA [327] provide a separation between training samples, which contain only normal images, and test samples, which include normal and defective cases, but they do not allocate a validation split. This separation reflects common industrial conditions, where collecting large volumes of normal data is straightforward due to the continuous operation of production lines, whereas defective samples are rare, heterogeneous in appearance, and often discovered only after deployment. Even when defective products are archived, their distribution may not represent all future defect modes, making them insufficient for reliable validation. Consequently, a validation set for monitoring generalization and preventing overfitting is often unavailable in practice, and traditional validation-driven checkpoint selection cannot be applied.

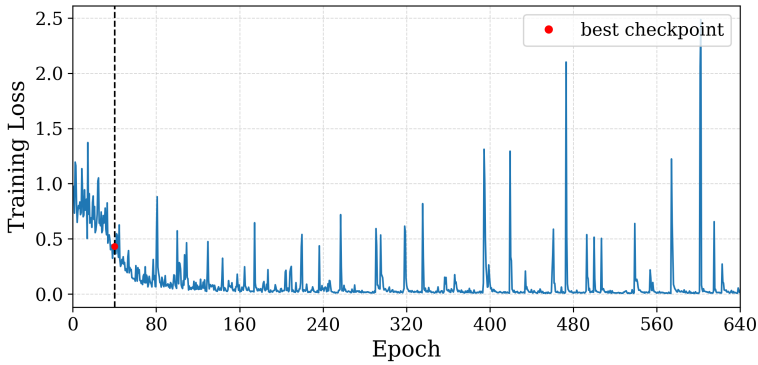
State-of-the-art frameworks on these benchmark datasets, such as GLASS (Global and local anomaly synthesis strategy) [44] and SimpleNet [162], operate in a fully unsupervised setting and optimize re-

construction or embedding objectives using normal training samples. While these objectives support the learning of compact representations of normal appearances, they do not provide a reliable indication of when the model has converged. The absence of validation data leads to a fundamental checkpoint-selection problem that directly affects the stability, fairness, and reproducibility of experimental results. In their default evaluation protocols, these frameworks select the checkpoint that performs best on the test set. However, this practice introduces information leakage from test data, artificially inflates performance scores, and prevents consistent comparison across different datasets or training conditions.

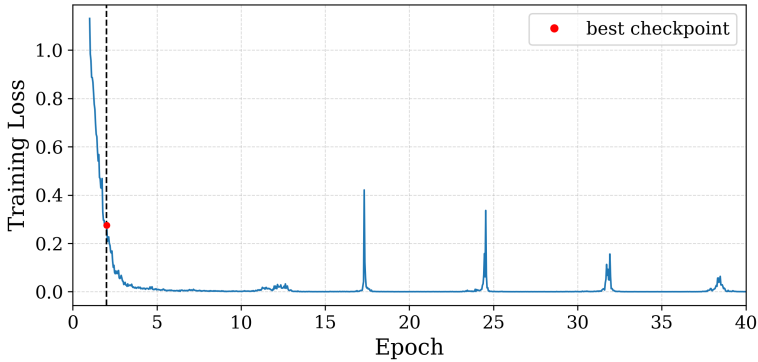
The challenge is further illustrated by examining the training-loss trajectories of GLASS [44] and SimpleNet [162]. Fig. 5.1 shows the training loss curves of both methods when trained on the transistor class of the MVTec AD dataset using their default training setups. Both loss curves exhibit irregular, non-monotonic behavior with local minima and sudden fluctuations across epochs. These patterns highlight that training loss alone cannot serve as a meaningful convergence indicator, as neither its absolute value nor its gradient correlates reliably with the quality of the learned representations.

An alternative solution is to train the model with a fixed number of epochs or steps. However, such a training schedule may suffer from underfitting or overfitting, as the optimal stopping point could vary with dataset complexity, network initialization, and optimization dynamics. Although the overall loss curves in Fig. 5.1 generally approach a steady level toward the final epoch for both frameworks, as shown later in Section 5.6, the models at that point have already overfit to the training data, as their performance on the test set deteriorates noticeably, confirming that loss convergence does not necessarily imply generalization.

Hence, a validation-free criterion that can detect whether the learned embeddings themselves have stabilized is needed for these anomaly detection frameworks, independent of the magnitude or smoothness of the optimization loss.



(a) Training loss of GLASS [44].



(b) Training loss of SimpleNet [162].

Figure 5.1 Training loss curves of GLASS [44] and SimpleNet [162] on the transistor class of the MVTec AD dataset. For GLASS, the horizontal axis is resolved directly by training epoch, whereas for SimpleNet the loss is recorded at training-step resolution and rescaled to the epoch axis for comparison. The red marker indicates the best checkpoint selected according to test performance.

5.3 Embedding-Stability-Guided Early Stopping

The proposed early stopping strategy monitors the temporal consistency of learned feature embeddings during training and halts optimization

once the change in these representations falls below an adaptive threshold. Unlike loss-based or test-driven checkpoint selection, the method relies only on the encoder outputs computed from normal training images and is therefore fully compatible with the validation-free setting described in Section 5.2.

5.3.1 Embedding Extraction

Following the common practice in encoder-based anomaly detection methods, image embeddings are constructed by aggregating multi-level feature maps extracted from a selected network layer. For an input image $\mathbf{X}^i$, the selected feature map is first represented as a set of local patch descriptors $\{\mathbf{p}_{j,t}^i\}_{j=1}^M$ at epoch t. These descriptors are then spatially averaged and vectorized to form an image-level embedding

$$\mathbf{e}_t^i = Vec\left(\frac{1}{M}\sum_{j=1}^{M}\mathbf{p}_{j,t}^i\right) \in \mathbb{R}^d, \tag{5.1}$$

where M denotes the number of patches, $Vec(\bullet)$ denotes the vectorization operator, which reshapes the aggregated patch feature into a column vector by stacking all its elements. The resulting embedding dimensionality is d. Note that spatial averaging reduces the influence of local fluctuations in individual patch embeddings and yields a stable estimate of the global feature shift between epochs, which is used here as the relevant signal for identifying convergence of the overall representation.

Collecting the N embeddings for the normal training set yields the embedding matrix

$$\mathbf{E}_t = \begin{bmatrix} (\mathbf{e}_t^1)^\top \\ (\mathbf{e}_t^2)^\top \\ \vdots \\ (\mathbf{e}_t^N)^\top \end{bmatrix} \in \mathbb{R}^{N\times d}, \tag{5.2}$$

where d denotes the embedding dimensionality. The sequence $\{\mathbf{E}_t\}_{t=0}^T$ forms the basis for analyzing the temporal evolution of the encoder's feature space.

5.3.2 Stability Divergence of Embeddings

Given the embedding matrices at two consecutive epochs, $\mathbf{E}_{t-1}$ and $\mathbf{E}_t$, the raw stability divergence is defined as

$$S_t = 1 - \frac{1}{N} \sum_{i=1}^{N} \cos(\mathbf{e}_{t-1}^i, \mathbf{e}_t^i), \tag{5.3}$$

where $\cos(\mathbf{e}_{t-1}^i, \mathbf{e}_t^i)$ denotes the cosine similarity between the feature embeddings of the same image at two consecutive epochs, computed as

$$\cos(\mathbf{e}_{t-1}^i, \mathbf{e}_t^i) = \frac{(\mathbf{e}_{t-1}^i)^\top \mathbf{e}_t^i}{\|\mathbf{e}_{t-1}^i\|_2 \, \|\mathbf{e}_t^i\|_2}. \tag{5.4}$$

Although the stability divergence is defined in terms of individual embedding vectors for clarity, in practice the computation is performed directly on the embedding matrices $\mathbf{E}_{t-1}$ and $\mathbf{E}_t$, allowing efficient parallel evaluation on GPUs via batched computations. A large value of S_t indicates substantial drift of the embedding space, whereas the value approaches zero as embeddings converge. Since S_t is defined between two consecutive epochs, the first observable stability divergence is S_1, which measures the embedding change between $\mathbf{E}_0$ and $\mathbf{E}_1$.

Because S_t may fluctuate locally across epochs, an exponential moving average (EMA) is applied:

$$S_t^{\mathrm{EMA}} = \alpha \, S_t + (1 - \alpha) \, S_{t-1}^{\mathrm{EMA}}, \qquad S_1^{\mathrm{EMA}} = S_1, \tag{5.5}$$

with smoothing coefficient $\alpha \in (0, 1)$.

The intuition behind the stability divergence is visualized conceptually in Fig. 5.2. Early in training, embeddings differ significantly across epochs. As the encoder converges, the embeddings from consecutive epochs become nearly identical, and the stability divergence shrinks accordingly.

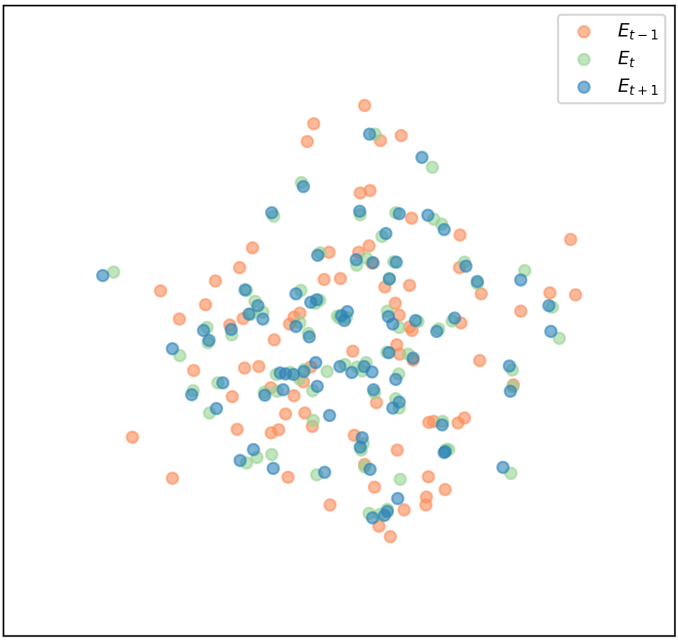


Figure 5.2 Conceptual illustration of embedding drift for three consecutive epochs under the simplifying assumption that the embeddings of data are 2D. The substantial deviation between $\mathbf{E}_{t-1}$ (orange) and $\mathbf{E}_t$ (green) illustrates that the encoder is still changing at epoch $t-1$, whereas the minor deviation from $\mathbf{E}_t$ (green) to $\mathbf{E}_{t+1}$ (blue) is consistent with a converged representation.

5.3.3 Adaptive Threshold for Convergence Detection

A fixed threshold for stability divergence is unsuitable across different datasets or architectures. Instead, an adaptive threshold is initialized from the first observed stability divergence:

$$\tau_{\text{ada}} = \beta \cdot S_1 \,, \tag{5.6}$$

where $\beta \in (0,1)$ controls the strictness of the criterion. Here, S_t denotes the stability signal used for stopping. It can refer either to the raw stability divergence or to its EMA-smoothed variant S_t^{EMA}, depending on the experimental setting. Using S_1 anchors the threshold to the model's initial

representation scale, ensuring that τ_{ada} is calibrated to the magnitude of feature variations specific to each training run. A smaller β enforces a stricter tolerance and tends to trigger early stopping later.

At each epoch, the stability divergence S_t is recorded. Training is terminated when S_t remains below the adaptive threshold for k consecutive epochs:

$$S_t < \tau_{\text{ada}} \qquad \text{for } t = t^*, t^*+1, \ldots, t^*+k-1\,. \tag{5.7}$$

The checkpoint from the first epoch of this stable interval, t^*, is retained as the early stopping checkpoint.

5.4 Integration into Anomaly Detection Frameworks

The proposed early stopping strategy can be incorporated into existing industrial anomaly detection frameworks, requiring no modification to losses, architectures, or objectives. Its integration requires only the extraction of image-level embeddings after each training epoch and the computation of the stability divergence described in Section 5.3. This design makes the method compatible with a broad range of encoder-based anomaly detection models, including reconstruction-driven approaches such as GLASS [44] as well as embedding-based methods like SimpleNet [162].

GLASS [44] constructs a normality manifold using real normal training samples and synthesized anomalies. For inference, it classifies test samples via reconstruction errors and feature-based comparisons. During training, GLASS [44] updates both the encoder and decoder parameters using a weighted combination of reconstruction and perceptual losses. However, these losses do not exhibit stable or monotonic behavior over epochs, and therefore cannot serve as reliable stopping indicators (*cf.* Fig. 5.1).

To integrate the proposed criterion, the encoder output of GLASS [44] is used to construct embeddings $\mathbf{E}_t$ as defined in (5.2). After each epoch, the stability divergence S_t is computed and compared with the adaptive threshold τ_{ada}. The checkpoint is stored once stability is consistently achieved.

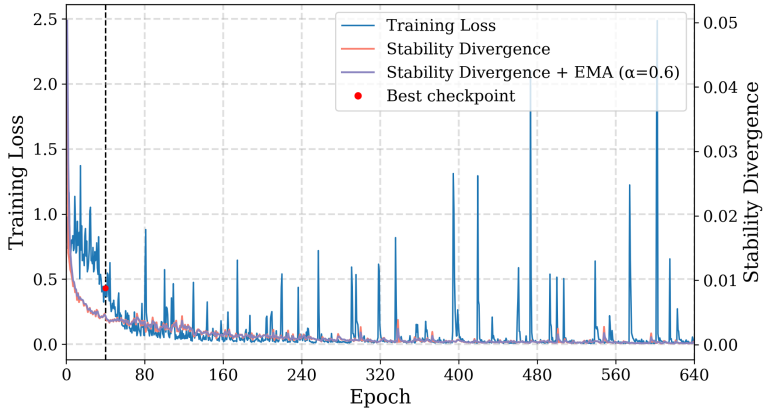


Figure 5.3 Stability divergence of GLASS [44] on the transistor class, compared with the corresponding training loss.

Figure 5.3 compares the stability divergence with the training loss for the transistor class of the MVTec AD dataset. While both quantities decrease substantially during the early stages of training and remain low thereafter, the training loss exhibits pronounced fluctuations and occasional spikes, whereas the stability divergence follows a considerably smoother trend toward a stable level.

SimpleNet [162] operates on extracted features and trains a lightweight discriminator to distinguish normal features from synthetically perturbed ones. Unlike GLASS [44], the feature extractor of SimpleNet [162] is frozen during the training process. Therefore, for SimpleNet [162], the image-level embedding $\mathbf{E}_t$ is taken from the second-to-last layer of its discriminator at epoch t, excluding the final classification layer that performs pixel-wise anomaly scoring. The rest of the integration procedure parallels that of GLASS [44].

Figure 5.4 compares the stability divergence of SimpleNet [162] with the corresponding training loss on the same dataset. In contrast to GLASS [44](*cf.* Fig. 5.3), the fluctuations of both quantities are strongly correlated, with most pronounced peaks occurring at similar stages of training. This behavior is related to the architecture of SimpleNet [162].

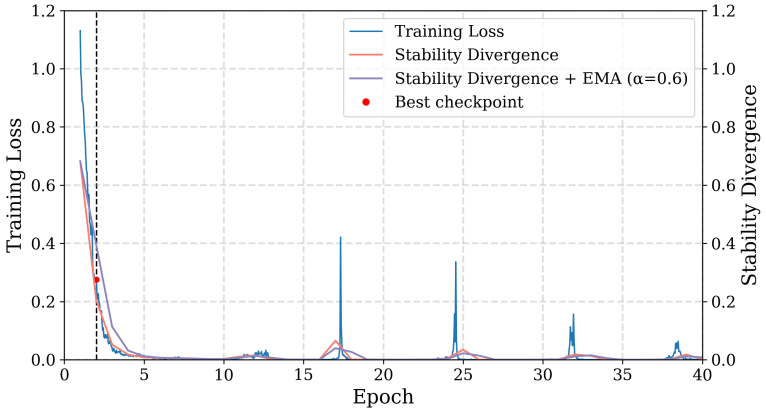


Figure 5.4 Stability divergence of SimpleNet [162] on the transistor class, compared with the corresponding training loss.

The feature extractor is frozen, and the stability divergence is therefore computed from the second-to-last layer of the discriminator, which is closely coupled to the network output used to calculate the training loss. The stability divergence exhibits substantially smaller fluctuations than the training loss, providing a smoother signal for validation-free early stopping.

The proposed integration introduces only limited computational overhead. Embedding extraction requires a single forward pass over normal training images, and the stability divergence computation is a lightweight cosine-similarity operation. No additional gradient updates or auxiliary networks are required, and the anomaly detection pipeline remains unchanged.

5.5 Experimental Setup

This section describes the datasets, models, training protocols, baseline strategies, and evaluation metrics used to assess the effectiveness of the proposed embedding-stability-guided early stopping. All experiments follow the validation-free setting of industrial anomaly detection as de-

fined in Section 5.2, using only normal samples for training unless stated otherwise.

5.5.1 Datasets

Experiments are conducted on two widely used industrial anomaly detection benchmarks:

- MVTec AD [18], consisting of 15 diverse object and texture categories with high-resolution industrial images. Each category provides normal images for training and both normal and anomalous images for testing.
- VisA [327], containing real-world product inspection data across multiple food and industry-relevant categories with significant appearance variation.

Neither dataset includes a validation split, reflecting the practical industrial setting in which only normal samples are reliably available during training.

5.5.2 Frameworks and Training Protocol

The proposed early stopping strategy is evaluated on both state-of-the-art anomaly detection frameworks GLASS [44] and SimpleNet [162]. Both frameworks follow their default training schedules, with GLASS [44] trained for a maximum length of 640 epochs and SimpleNet [162] for 40 epochs. For both frameworks, except for the used feature extractors, all trainable components are trained from scratch.

To better evaluate the performance of the proposed early stopping strategy, three checkpoint-selection protocols are included for comparison:

1. Best-test protocol: Following the default behavior in GLASS [44] and SimpleNet [162] implementations, the checkpoint that achieves the highest test-set anomaly detection performance is selected. Although this approach produces an upper bound on achievable performance, it leaks information from the test set and is therefore invalid for practical deployment.

2. Fixed-epoch protocol: Models are trained for a fixed number of epochs, *i.e.*, 640 for GLASS [44] and 40 for SimpleNet [162]. The final checkpoint is then selected.

3. Early stopping protocol: Training is monitored using the proposed stability divergence signal, which saves the checkpoint once the change in the training image embeddings remains below the adaptive threshold for several consecutive epochs.

5.5.3 Early Stopping Hyperparameters

Unless specified otherwise, the EMA-smoothed stability divergence S_t^{EMA} defined in (5.5) is applied. The EMA smoothing coefficient is set to

$$\alpha = 0.6$$

for both frameworks. Ablation experiments in Section 5.6.2.1 later show that using S_t^{EMA} with $\alpha \in [0.6, 0.8]$ or the unsmoothed signal S_t yields similar performance on the cable class of MVTec AD, which is used as the calibration class for the hyperparameter analysis.

For both GLASS [44] and SimpleNet [162], the adaptive threshold τ_{ada} is initialized according to (5.6), but with method-specific scaling factors:

$$\beta = 0.06 \quad \text{for GLASS}, \qquad \beta = 0.02 \quad \text{for SimpleNet}.$$

The adaptive threshold factor β is chosen via a grid search on the cable class of MVTec AD for each framework and then kept fixed for all other categories in both MVTec AD and VisA.

The required number of consecutive stable epochs, k, reflects the difference in their maximum training lengths:

$$k = 5 \quad \text{for GLASS}, \qquad k = 3 \quad \text{for SimpleNet}.$$

These values ensure that the stopping condition is sufficiently robust to short-term fluctuations while remaining effective for detecting convergence.

5.5.4 Evaluation Metrics

Following standard practice, results are quantitatively assessed using two complementary metrics: image-level AUROC (area under ROC curve) for overall defect detection and pixel-level AUROC for anomaly localization (*cf.* Appendix A.5). Results are reported separately for each category and aggregated via class-average scores.

5.6 Experimental Results

This section evaluates the effectiveness of the proposed embedding-stability-guided early stopping across industrial anomaly detection benchmarks. Quantitative comparisons are conducted against the commonly used best-test and fixed-epoch checkpoint-selection strategies on both MVTec AD and VisA, followed by qualitative analysis of inference on test data. An ablation study further examines the influence of key hyperparameters, including the EMA smoothing coefficient α and the threshold scaling factor β.

5.6.1 Anomaly Detection Performance on Industrial Benchmarks

This section reports the quantitative performance of the proposed embedding-stability-guided early stopping on MVTec AD and VisA. For each model and dataset, three checkpoint-selection protocols are compared: the best-test checkpoint determined using test data (upper bound, not admissible in practice), the fixed final checkpoint obtained after the default training schedule, and the proposed early stopping checkpoint selected using the stability divergence criterion. This comparison distinguishes between a test-set-based upper bound, and two deployable checkpointing strategies that must operate without access to labeled validation anomalies. All results are reported as image-level and pixel-level AUROC (%).

The following comparison focuses on whether early stopping can recover the performance loss caused by fixed-length training under a validation-free setting.

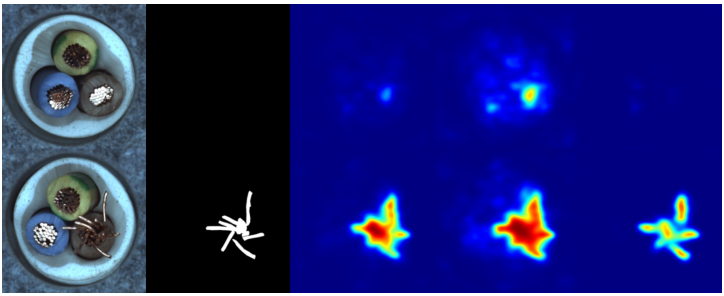
GLASS on MVTec AD

Table 5.1 GLASS [44] on MVTec AD: Image / Pixel AUROC (%).

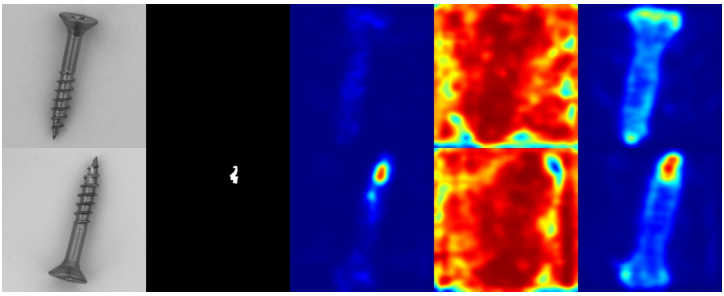
Class	Best-test	Fixed	ES
bottle	100.00 / 99.27	100.00 / 99.08	99.68 / 98.82
cable	99.23 / 98.63	89.11 / 81.88	99.25 / 98.39
capsule	99.96 / 99.26	99.24 / 99.23	98.24 / 99.21
carpet	99.84 / 99.37	97.99 / 99.36	97.99 / 99.36
grid	100.00 / 99.31	100.00 / 99.20	100.00 / 99.20
hazelnut	100.00 / 99.33	99.93 / 98.87	99.93 / 98.87
leather	100.00 / 99.83	100.00 / 99.55	100.00 / 99.55
metal_nut	100.00 / 99.34	97.90 / 94.80	97.90 / 94.80
pill	99.37 / 99.27	95.01 / 98.46	96.21 / 98.38
screw	99.10 / 99.49	77.00 / 95.96	96.70 / 99.31
tile	100.00 / 99.67	100.00 / 99.37	100.00 / 99.37
toothbrush	100.00 / 99.07	97.78 / 98.59	97.78 / 98.59
transistor	99.83 / 97.60	81.58 / 75.56	99.75 / 96.75
wood	100.00 / 98.53	99.56 / 97.94	99.56 / 97.94
zipper	99.90 / 99.65	99.05 / 99.51	99.05 / 99.51
Mean	**99.82 / 99.17**	**95.61 / 95.82**	**98.80 / 98.54**

Table 5.1 shows the results on MVTec AD for GLASS [44]. The fixed-epoch baseline suffers from overfitting in several categories, most notably in the cable, screw, and transistor categories, where image-level AUROC drops to 89.11, 77.00, and 81.58, respectively. The proposed early stopping strategy recovers most of this drop, raising the corresponding scores to 99.25, 96.70, and 99.75, and improves the mean image/pixel AUROC from 95.61/95.82 to 98.80/98.54.

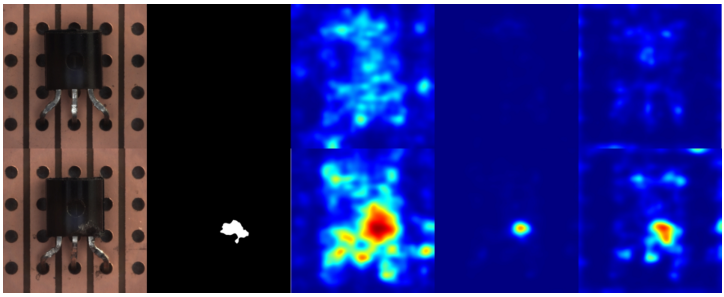
For several classes (*e.g.*, carpet, grid, hazelnut *etc*.), the early stopping and fixed checkpoints coincide exactly. In these cases, the criterion does not trigger before the final epoch, so the procedure falls back to the fixed final checkpoint. This behavior indicates that the stopping rule is conservative, while still correcting strong late-training degradation when it occurs.



(a) cable



(b) screw



(c) transistor

Figure 5.5 Qualitative comparison of anomaly maps for GLASS [44] on selected MVTec AD classes. For each class, the first row shows a normal test image and the second row a defective image, followed by the ground truth (GT) defect mask and anomaly maps from the best-test, fixed-epoch, and early stopping (ES) checkpoints.

Fig. 5.5 illustrates example anomaly maps for the cable, screw, and transistor classes, for which the developed early stopping mechanism yields the largest improvements for GLASS [44]. For each class, the first row shows a normal test image and the second row a defective sample, followed by the corresponding defect mask. The remaining columns depict the anomaly maps obtained from the best-test, fixed-epoch, and early stopping checkpoints.

For the cable class, the anomaly maps reveal clear differences among the three checkpoints. On the normal sample, the fixed-epoch model produces distinct false positives around the cable cores, whereas the best-test and early stopping checkpoints suppress these activations. On the defective sample, the best-test checkpoint produces a strong response on the protruding wire cluster but slightly over-extends the activated region by including the small normal areas between the wires. The early stopping checkpoint shows the complementary behavior. It keeps these inter-wire regions close to background level, yielding a more precise separation between normal and defective pixels, although its activation on the wires is marginally less intense. The fixed-epoch model performs noticeably worse, as it both weakens the response on the true defect and produces substantial false positives.

For the screw class, the fixed-epoch checkpoint performs poorly. Its heatmaps exhibit uniformly high activation on both normal and defective samples, suggesting that the model has converged to a poor solution that no longer separates normal and defective patterns meaningfully. For both the normal test image and the defective test image, the best-test checkpoint yields the cleanest response, with the screw surface only mildly highlighted and the background almost fully suppressed. In contrast, the early stopping checkpoint shows a slightly stronger response on the screw body, leading to a higher false positive rate than the best-test model.

For the transistor class, the fixed checkpoint underestimates the anomaly score. Although it produces near-zero activation on the normal sample, it also yields a noticeably smaller predicted region on the defective sample, missing substantial portions of the true anomaly and resulting in a high false-negative rate. The best-test checkpoint exhibits the opposite tendency, assigning elevated activation more broadly across

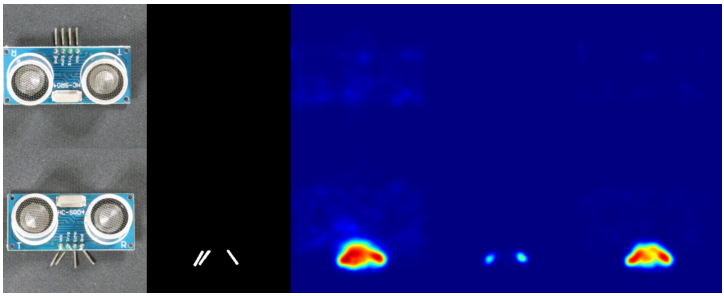
both samples. This reduces false negatives on the defect image but introduces a false-positive bias on normal regions. The early stopping checkpoint offers the most balanced behavior. It maintains low activation on the normal test image while localizing the defect compactly.

GLASS on VisA

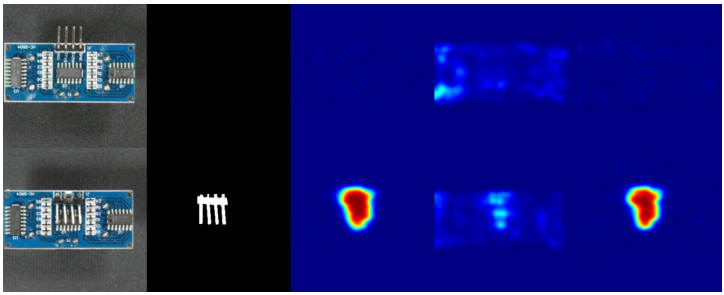
Table 5.2 GLASS [44] on VisA: Image / Pixel AUROC (%).

Class	Best-test	Fixed	ES
candle	98.57 / 99.30	96.59 / 99.12	97.07 / 99.27
capsules	96.50 / 99.11	92.65 / 98.77	92.65 / 98.77
cashew	98.92 / 98.86	96.08 / 97.72	97.08 / 99.02
chewinggum	99.84 / 98.94	99.02 / 98.79	99.78 / 98.77
fryum	97.68 / 93.86	95.74 / 88.73	94.38 / 93.32
macaroni1	99.70 / 99.63	98.55 / 99.60	98.55 / 99.60
macaroni2	91.82 / 99.42	90.20 / 99.32	90.20 / 99.32
pcb1	97.86 / 99.34	85.44 / 98.91	96.36 / 99.28
pcb2	99.30 / 98.12	84.46 / 91.97	95.14 / 97.75
pcb3	99.43 / 98.87	98.08 / 96.63	98.08 / 96.63
pcb4	99.16 / 97.89	98.76 / 95.19	98.43 / 95.72
pipe_fryum	99.74 / 99.26	92.78 / 95.59	99.66 / 99.17
Mean	**98.21 / 98.55**	**94.03 / 96.69**	**96.45 / 98.05**

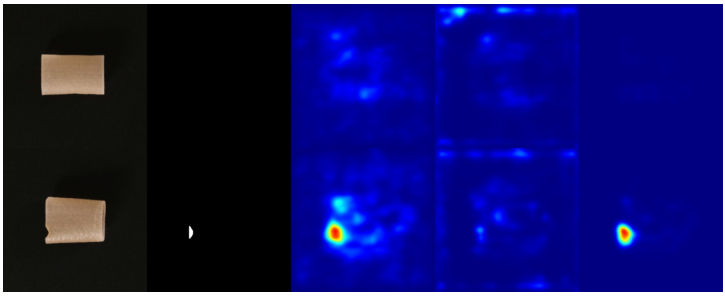
From Table 5.2, a similar pattern can be observed on VisA. The results assess whether the same pattern holds under a dataset with different object categories and defect characteristics. The fixed-epoch baseline underperforms significantly in several classes, including pcb1, pcb2 and pipe_fryum, where late training epochs degrade anomaly detection performance. Early stopping improves the mean image/pixel AUROC from 94.03/96.69 to 96.45/98.05, largely closing the gap to the upper bound represented by the best-test checkpoint.



(a) pcb1



(b) pcb2



(c) pipe_fryum

Figure 5.6 Qualitative comparison of anomaly maps for GLASS [44] on selected VisA classes. For each class, the first row shows a normal test image and the second row a defective image, followed by the ground truth (GT) defect mask and anomaly maps from the best-test, fixed-epoch, and early stopping (ES) checkpoints.

Fig. 5.6 presents representative qualitative results for the three VisA classes where early stopping provides the clearest improvement. On the normal test samples, both the best-test and early stopping checkpoints show consistently low activation values, whereas the fixed checkpoint occasionally assigns elevated anomaly scores to background regions (*e.g.*, in pcb2 and pipe_fryum), indicating instability when trained without a convergence-aware stopping rule.

Across all shown defect examples, the fixed checkpoint consistently underestimates the anomaly score. The predicted heatmaps exhibit only weak or fragmentary activation on the true defect regions. In contrast, both the best-test and early stopping checkpoints correctly highlight the defective area. Between them, the early stopping checkpoint produces the most accurate localization. Activations are tightly concentrated within the annotated defect region, while normal pixels remain almost uniformly suppressed.

For SimpleNet, the main question is whether the proposed criterion remains effective when the evolving representation is taken from the discriminator rather than the encoder.

SimpleNet on MVTec AD

For SimpleNet [162] on MVTec AD, as shown in Table 5.3, the fixed-epoch baseline again exhibits pronounced drops in several classes, particularly in the cable and toothbrush classes. Early stopping improves the mean performance from 96.55/94.36 to 98.89/97.69, almost matching the best-test mean of 99.49/97.68. As with GLASS [44], several classes exhibit identical fixed and early stopping scores, indicating that the early stopping criterion selects the final checkpoint in these cases.

Fig. 5.7 illustrates the qualitative performance of SimpleNet [162] on the cable and toothbrush classes of the MVTec AD dataset.

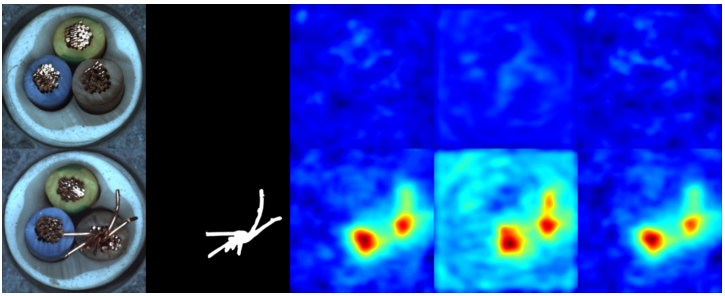
For the cable class, the fixed-epoch checkpoint produces noticeably elevated activations in regions surrounding the defect as well as in background areas of the defective sample. This is consistent with the quantitative degradation observed in Table 5.3, and suggests that the model trained for the full schedule over-amplifies irrelevant structures resulting in a substantial number of false positives. On the corresponding

Table 5.3 SimpleNet [162] on MVTec AD: Image / Pixel AUROC (%).

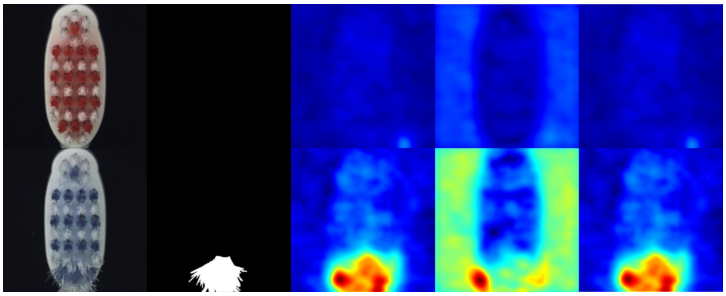
Class	Best-test	Fixed	ES
bottle	100.00 / 98.18	100.00 / 98.02	100.00 / 98.01
cable	99.70 / 97.28	81.32 / 88.57	98.91 / 97.27
capsule	98.09 / 98.98	96.61 / 98.94	96.93 / 98.97
carpet	99.80 / 98.33	99.28 / 98.31	99.04 / 97.15
grid	99.92 / 98.15	98.41 / 98.56	98.66 / 98.71
hazelnut	100.00 / 98.18	99.82 / 97.96	99.86 / 96.70
leather	100.00 / 99.12	100.00 / 98.85	100.00 / 99.05
metal_nut	99.95 / 98.25	99.56 / 98.26	99.76 / 98.61
pill	98.66 / 98.48	96.45 / 97.99	96.56 / 98.44
screw	97.32 / 99.11	94.43 / 99.20	96.99 / 99.04
tile	99.64 / 94.78	99.53 / 96.69	98.56 / 96.84
toothbrush	99.44 / 98.57	85.00 / 56.51	99.44 / 98.57
transistor	100.00 / 95.99	99.29 / 95.28	99.50 / 96.49
wood	99.91 / 92.81	98.77 / 93.47	99.47 / 92.54
zipper	99.89 / 98.97	99.79 / 98.84	99.74 / 98.95
Mean	**99.49 / 97.68**	**96.55 / 94.36**	**98.89 / 97.69**

normal sample, all three checkpoints produce similarly low activations, indicating that the defect-free case is inherently less sensitive to training instability. The best-test and early stopping checkpoints generate more concentrated anomaly responses on the true defect region.

For the toothbrush class, the fixed-epoch checkpoint produces an inverted activation profile: background regions receive the highest scores, followed by defective areas, while intact bristle regions exhibit the lowest responses. This ordering contradicts the expected anomaly structure, in which defective bristles should dominate the activation. A similar pattern appears on the normal sample, where background pixels are incorrectly emphasized. In contrast, the best-test and early stopping checkpoints suppress the background response and assign higher activation to the defective bristle region, which is consistent with the recovery in pixel-level AUROC reported in Table 5.3.



(a) cable



(b) toothbrush

Figure 5.7 Qualitative comparison of anomaly maps for SimpleNet [162] on selected MVTec AD classes. For each class, the first row shows a normal test image and the second row a defective image, followed by the ground truth (GT) defect mask and anomaly maps from the best-test, fixed-epoch, and early stopping (ES) checkpoints.

The results on VisA for SimpleNet provide a particularly demanding test case, as the gap between the theoretical upper bound and the performance of deployable checkpoint selection strategies is larger than on MVTec AD.

SimpleNet on VisA

As shown in Table 5.4, SimpleNet [162] shows the largest discrepancy between the best-test and fixed-epoch checkpoints on VisA, particularly for challenging categories such as macaroni2, pcb4, and pipe_fryum. The fixed baseline yields a mean image/pixel AUROC of 86.68/92.60,

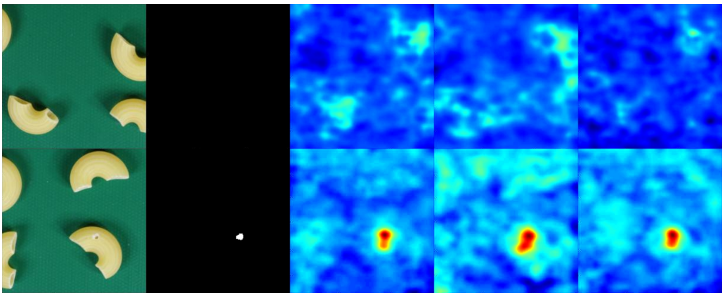
Table 5.4 SimpleNet [162] on VisA: Image / Pixel AUROC (%).

Class	Best-test	Fixed	ES
candle	97.66 / 97.75	93.47 / 95.80	96.37 / 97.97
capsules	90.42 / 98.60	79.05 / 94.92	90.42 / 98.60
cashew	96.62 / 98.47	93.44 / 98.10	89.54 / 97.69
chewinggum	99.86 / 98.21	99.74 / 97.65	99.72 / 97.89
fryum	98.26 / 86.80	97.82 / 89.35	97.80 / 88.43
macaroni1	99.57 / 99.39	95.77 / 98.48	98.68 / 99.27
macaroni2	82.01 / 94.10	62.73 / 91.73	74.87 / 96.71
pcb1	99.14 / 99.33	89.38 / 98.07	97.69 / 99.27
pcb2	98.51 / 97.60	97.06 / 95.97	97.41 / 97.61
pcb3	98.70 / 98.53	94.46 / 98.10	97.94 / 98.57
pcb4	99.40 / 96.65	53.76 / 60.24	98.54 / 95.85
pipe_fryum	99.94 / 98.62	83.52 / 92.85	99.08 / 98.28
Mean	**96.67 / 97.00**	**86.68 / 92.60**	**94.84 / 97.18**

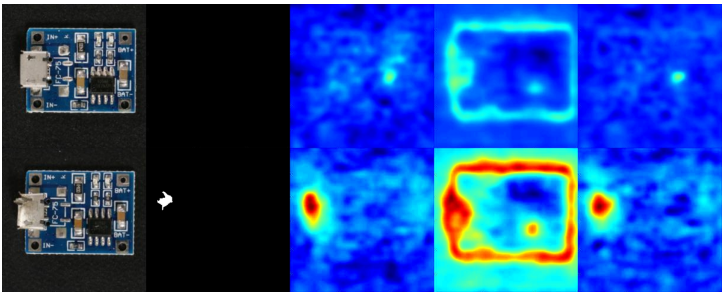
whereas early stopping raises this to 94.84/97.18, substantially narrowing the gap to the best-test mean of 96.67/97.00.

In both Table 5.3 and Table 5.4, the pixel-level AUROC of the early stopping checkpoint slightly exceeds that of the best-test checkpoint. This does not contradict the criterion for the best-test checkpoint, which considers the sum of image- and pixel-level AUROC rather than the pixel-level score alone. Instead, such a result indicates that the early stopping checkpoint improves pixel-level localization of defects compared with the test-based best-test checkpoint.

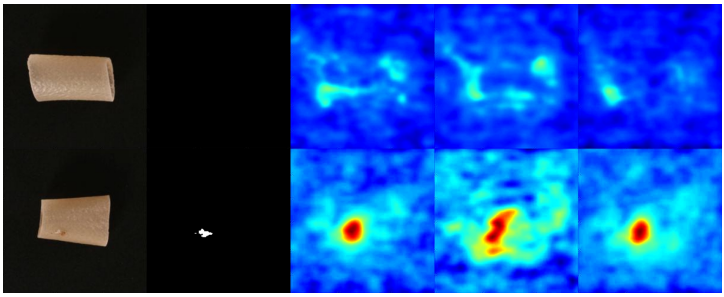
Fig. 5.8 shows the qualitative results of SimpleNet [162] for three VisA categories. For macaroni2, both the best-test and early stopping checkpoints correctly identify the small defect region while maintaining low activations on normal pixels. The best-test checkpoint shows a slightly more precise prediction of the defective area, whereas the early stopping checkpoint produces lower background activations, which reduces false alarms. In contrast, the fixed checkpoint shows elevated responses across broader regions of both the normal and defective images, indicating a higher false-positive rate.



(a) macaroni2



(b) pcb4



(c) pipe_fryum

Figure 5.8 Qualitative comparison of anomaly maps for SimpleNet [162] on selected VisA classes. For each class, the first row shows a normal test image and the second row a defective image, followed by the ground truth (GT) defect mask and anomaly maps from the best-test, fixed-epoch, and early stopping (ES) checkpoints.

For the pcb4 class, the fixed checkpoint consistently highlights the outer contour of the PCB for both normal and defective samples, which reflects a failure to learn meaningful anomaly-sensitive features, while both the best-test and early stopping checkpoints produce focused responses on the true defect in the defective sample.

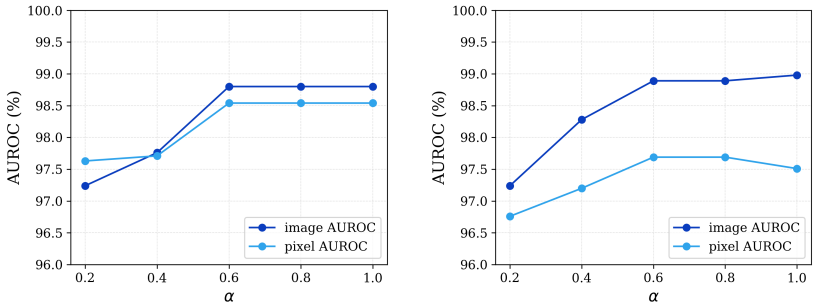
For the pipe_fryum class, the overall behavior mirrors that of macaroni2. The best-test and early stopping checkpoints both localize the defect well while keeping the normal sample anomaly-free. The fixed checkpoint, however, produces noticeably higher activations on both the normal and defective images, revealing a systematic false-positive bias.

5.6.2 Analysis of Hyperparameter Choices

This subsection examines how the key hyperparameters of the proposed early stopping strategy influence the resulting anomaly-detection performance. Since neither MVTec AD nor VisA provides a validation split, the analysis is restricted to the cable class of MVTec AD, which is used as a calibration class. The EMA coefficient α and the threshold scaling factor β are varied independently on this class. The resulting analysis supports the default values for α and β specified in Section 5.5.3, while the full benchmark results in Section 5.6.1 evaluate whether these fixed values transfer to the remaining categories and datasets. The number of required consecutive stable epochs, k, is chosen empirically based on the stability divergence curves in Figs. 5.3 and 5.4 and the maximum number of training epochs of each framework.

5.6.2.1 Smoothing Coefficient α

Figure 5.9 illustrates how the smoothing coefficient α influences the resulting early stopping performance for GLASS [44] and SimpleNet [162] on the cable class of MVTec AD. When α is set to 1, no temporal smoothing is applied and the stability signal follows the epoch-wise embedding drift directly. Across both methods, values in the range $\alpha \in [0.6, 1.0]$ yield consistently strong image-level and pixel-level AUROC, indicating that the early stopping criterion is relatively insensitive to the exact amount of smoothing within this interval. In contrast, overly small values $\alpha \leq 0.4$ result in a noticeable performance reduction. Strong smoothing



(a) GLASS [44]

(b) SimpleNet [162]

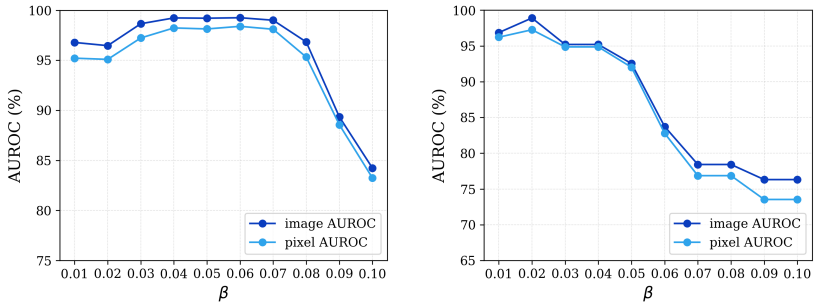
Figure 5.9 Effect of the smoothing coefficient α on image- and pixel-level AUROC for the cable class of the MVTec AD dataset.

introduces additional inertia into the stability signal, causing later divergence values to depend more strongly on earlier observations. This may delay the apparent stabilization of the representations and, consequently, the selected stopping point. Overall, the results on the calibration class suggest that any value within $\alpha \in [0.6, 1.0]$ provides a suitable choice for the subsequent benchmark evaluation.

5.6.2.2 Adaptive Threshold Factor β

Fig. 5.10 summarizes the grid search over the adaptive threshold factor β on the MVTec AD cable class. For GLASS [44], both image- and pixel-level AUROC are highest in the range $\beta \in [0.04, 0.07]$, with very similar performance across these values. Smaller values such as $\beta = 0.01$ or $\beta = 0.02$ lead to a slightly reduced AUROC, which indicates that an overly strict threshold delays the stopping point and lets training continue beyond the point where the embeddings have effectively stabilized. In contrast, larger values $\beta \geq 0.08$ cause a pronounced drop in both metrics, since the higher threshold is already satisfied at comparatively large stability divergences and the model is stopped before reaching a well-converged representation.

A similar situation is observed for SimpleNet [162]. The best performance is obtained around $\beta = 0.02$, and increasing β beyond this value



(a) GLASS [44] **(b)** SimpleNet [162]

Figure 5.10 Effect of the adaptive threshold factor β on image- and pixel-level AUROC for the cable class of the MVTec AD dataset.

quickly degrades the AUROC, which again suggests premature stopping. For values close to the optimum, such as $\beta = 0.03$ and $\beta = 0.04$, the results remain similar, indicating that the method is not overly sensitive within a narrow range around the chosen default. The identical scores observed for larger values such as $\beta = 0.09$ and $\beta = 0.10$ indicate that these thresholds select the same stopping epoch, but they correspond to substantially degraded performance and should therefore not be interpreted as robust choices. The occurrence of identical scores is expected because SimpleNet is trained for at most 40 epochs and the corresponding thresholds often select the same stopping epoch. Overall, the experiments indicate that the selected defaults

$$\tau_{\text{ada}}^{\text{GLASS}} = 0.06 \cdot S_1\,, \qquad \tau_{\text{ada}}^{\text{SimpleNet}} = 0.02 \cdot S_1$$

lie in regions where the AUROC curves are close to their maxima while avoiding unnecessarily late stopping. If β is chosen much smaller, the criterion may only be satisfied very late or not at all, in which case the procedure degenerates to selecting the final checkpoint and therefore becomes equivalent to the fixed-epoch baseline.

5.7 Conclusion

This chapter introduced an embedding-stability-guided early stopping strategy for industrial visual anomaly detection models. By monitoring the temporal consistency of image-level embeddings instead of relying on loss values or performance on the test set, the method enables validation-free checkpoint selection for encoder-based models trained only on normal data. The proposed criterion was integrated into two state-of-the-art frameworks, GLASS [44] and SimpleNet [162], without modifying their architectures or objectives.

Extensive experiments on MVTec AD and VisA showed that the fixed-epoch training schedules used in existing work can lead to pronounced performance degradation, particularly due to late-training overfitting or unstable convergence. The stability-guided early stopping mitigated these effects in most settings, recovering most of the performance lost by the fixed baseline and reducing the gap to the upper bound given by the best-test checkpoint selected on the test set. In several cases, the early stopping checkpoint achieved a higher pixel-level AUROC than the checkpoint selected based on test-set performance. Qualitative inspection of the corresponding heatmaps further indicated improved localization, with fewer false-positive and false-negative regions.

Overall, the embedding-stability-guided early stopping developed in this chapter provides a practical and reproducible protocol for training industrial anomaly detection models without labeled validation data. It removes the dependence on test-based checkpoint selection, stabilizes the optimization of both real- and synthetic-data models, and establishes a consistent training setup for the remainder of this work. The next chapter complements this perspective by analyzing how additional synthetic training data influences downstream tasks, including but not limited to anomaly detection, under consistent evaluation protocols.

6 Assessing Synthetic Data and Model Stability Across Downstream Tasks

This chapter investigates how synthetic training data and stability-guided model selection influence the performance of downstream vision tasks in industrial inspection. Building on the synthetic datasets introduced in Chapter 3 and Chapter 4, along with the embedding-stability-based early stopping strategy developed in Chapter 5, the following analyses examine three representative tasks with distinct requirements: granule classification, crack segmentation, and anomaly detection for both defective granules and metallic parts with surface cracks. Each task is evaluated under varying compositions of real and synthetic data, including refined synthetic variants and controlled mixing ratios, while anomaly detection additionally incorporates models selected via the proposed early stopping criterion. The goal of this chapter is to provide a systematic and task-specific assessment of how synthetic data quality and training dynamics affect generalization in practical industrial scenarios.

6.1 Introduction

The previous chapters established a complete pipeline for generating synthetic data for industrial inspection, ranging from modeling-based rendering to neural style transfer refinement and stability-guided model selection. While these components address the generation and optimization of training data, their practical value should be evaluated based on how well they support downstream vision tasks. Industrial quality inspection comprises a diverse set of problem formulations, including object classification, pixel-level segmentation, and anomaly detection,

each with distinct sensitivity to data quality. A systematic evaluation across these tasks is therefore required to understand when synthetic data are beneficial, how refinement strategies affect generalization, and under which conditions stability-based early stopping contributes to more robust model selection.

This chapter investigates these questions through three representative downstream tasks. Granule classification examines global appearance characteristics and provides an initial assessment of how synthetic samples influence discriminative training. Crack segmentation evaluates pixel-wise localization performance when models are trained exclusively on synthetic imagery but deployed in real domains. Anomaly detection evaluates the ability of models to learn stable representations of normality and reveals their sensitivity to distributional differences between real and synthetic normal data.

Across all tasks, the evaluation covers multiple training regimes, including real-only baselines, augmentation with synthetic and refined synthetic data, and controlled mixtures with fixed total sample counts. Performance is measured using task-specific metrics such as classification accuracy, mean intersection-over-union (mIoU), and image- and pixel-level area under the ROC curve (AUROC). The results highlight differences in how each task responds to synthetic data and reveal conditions under which refinement and stability-based selection improve generalization. The remainder of this chapter presents these analyses in detail.

6.2 Classification on Synthetic and Real Granule Images

This section evaluates how synthetic training data influence image classification performance on two granule datasets introduced in Chapter 3. The experiments include models trained only on real samples, models trained with additional synthetic or refined synthetic images, and models trained with specific ratios of real and synthetic data. These experiments provide a view of how synthetic data quality, refinement methods, and training data composition affect downstream classification accuracy. The results also highlight differences between the two datasets, which helps

to assess the robustness and practical value of synthetic data in industrial inspection settings.

6.2.1 Task Description and Experimental Setup

The classification experiments evaluate how synthetic training data influence the detection of defective granules. Two real granule datasets with distinct visual characteristics (*cf.* Chapter 3) are considered. $\mathcal{D}^{\text{rl}}_{\text{real}}$ contains reflected-light images with relatively uniform illumination, while $\mathcal{D}^{\text{tl}}_{\text{real}}$ contains transmitted-light images with stronger lighting variations and more complex reflectance patterns. Both datasets include normal and defective granules and are divided into training, validation, and test sets. For $\mathcal{D}^{\text{rl}}_{\text{real}}$, the split consists of 1000 normal and 200 defective training samples, 600 normal and 120 defective validation samples, and 400 normal and 80 defective test samples. $\mathcal{D}^{\text{tl}}_{\text{real}}$ contains 520 normal and 520 defective training samples, 150 normal and 150 defective samples for validation, and 75 normal and 75 defective samples for testing.

Corresponding synthetic datasets are generated separately for $\mathcal{D}^{\text{rl}}_{\text{real}}$ and $\mathcal{D}^{\text{tl}}_{\text{real}}$ to preserve domain-specific appearance, as introduced in Chapter 3. The synthetic dataset $\mathcal{D}^{\text{rl}}_{\text{syn}}$ contains 1000 normal and 200 defective samples, and the synthetic dataset $\mathcal{D}^{\text{tl}}_{\text{syn}}$ contains 520 normals and 520 defectives. Several refined synthetic variants are included, such as CycleGAN [322]-, CUT [202]-, and NST-based methods, as well as feature- and augmentation-enhanced extensions of the standard NST. Synthetic images are used only as training data.

Three training regimes are examined. The real-only setting, which uses the complete real training set, is considered as the baseline. The second setting augments real data with all available synthetic samples of the same batch. A third setting investigates fixed-size mixtures in which the total number of training images is kept equal to the size of the real training set while the ratio between real and synthetic samples is varied. These controlled mixtures range from real-only to synthetic-only compositions and are evaluated using the refinement method with the best performance under the second setting.

The classifiers are based on standard convolutional architectures, including ResNet-18 [100], MobileNetV2 [238], and ShuffleNetV2 [169]. All

models are trained for 200 epochs using stochastic gradient descent with a learning rate of 0.1, momentum of 0.9, and weight decay of 10^{-4}. The learning rate is reduced by a factor of ten at epochs 50, 100, and 150. Images are normalized with ImageNet [62] statistics and augmented with horizontal flips and rotations by multiples of 90°. Training is performed with a batch size of 16, and the checkpoint achieving the highest validation accuracy is selected. Each configuration is repeated three times. The reported metrics are accuracy, precision, recall, and F1 score, averaged over three runs. More details about these metrics are provided in Appendix A.3.

6.2.2 Experimental Results

The following results summarize the classification performance obtained under different training data configurations. The evaluation includes models trained only on real images, models trained with additional synthetic or refined synthetic samples, and models trained with controlled mixtures of real and synthetic data while keeping the total number of training images fixed. The results are organized into baseline performance, the effect of synthetic and refined synthetic data, controlled mixture experiments, and the behavior of models trained exclusively on synthetic images.

6.2.2.1 Baseline Performance Using Real Training Data

The baseline configuration evaluates the classification performance achieved when the models are trained exclusively on real images. This setting provides a reference for later comparisons with synthetic and refined synthetic training data. Both real granule datasets $\mathcal{D}^{\mathrm{rl}}_{\mathrm{real}}$ and $\mathcal{D}^{\mathrm{tl}}_{\mathrm{real}}$ are included, which vary in illumination, contrast, and texture appearance.

For each dataset, the full real training split is used without modification, and the checkpoint with the highest validation accuracy is selected. The results represent the average performance over three independent runs and are reported in terms of accuracy, precision, recall, and F1 score.

A summary of the baseline results for both datasets is given in Table 6.1. The values reported in this table serve as a reference for the following

Table 6.1 Baseline classification performance using real training data only.

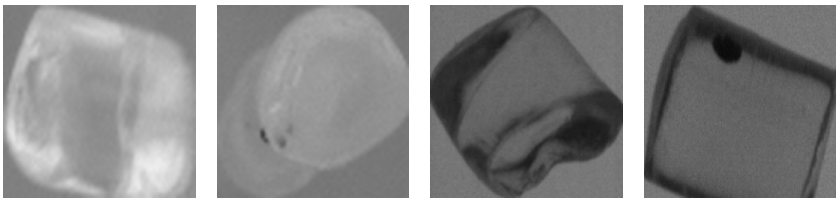
Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
$\mathcal{D}^{\text{rl}}_{\text{real}}$	93.22	96.14	95.23	95.68
$\mathcal{D}^{\text{tl}}_{\text{real}}$	87.78	88.47	86.00	87.22

subsections, where synthetic and refined synthetic training data are introduced.

The results show that $\mathcal{D}^{\text{rl}}_{\text{real}}$ achieves a higher baseline accuracy than $\mathcal{D}^{\text{tl}}_{\text{real}}$. This performance gap is likely related to the different visibility of burn marks under the two imaging conditions. In the reflected-light images of $\mathcal{D}^{\text{rl}}_{\text{real}}$, burn marks appear as distinct dark specks against a comparatively bright and homogeneous background, which makes them visually separable from the surrounding granule structure. In contrast, the transmitted-light images of $\mathcal{D}^{\text{tl}}_{\text{real}}$ contain strong reflections and refraction effects that create darker regions on the granule surface, some of which are visually similar to actual defects. These effects reduce the contrast between defective and non-defective regions and increase the ambiguity in local appearance, which makes defect identification more difficult for the classifier. Representative examples from both datasets are shown in Fig. 6.1.

6.2.2.2 Influence of Synthetic and Refined Synthetic Training Data

This subsection evaluates how synthetic and refined synthetic samples influence the classification performance when added to the real training



(a) normal **(b)** defective **(c)** normal **(d)** defective

Figure 6.1 Examples of normal and defective granules from $\mathcal{D}^{\text{rl}}_{\text{real}}$ (a,b) and $\mathcal{D}^{\text{tl}}_{\text{real}}$ (c,d).

Table 6.2 Classification performance for reflected-light granule images using extra synthetic training data.

Extra Training Data	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
None	93.22	96.14	95.23	95.68
Original synthetic	89.67	91.45	90.02	90.73
CycleGAN [322]	92.89	95.84	94.62	95.23
CUT [202]	91.33	93.59	93.21	93.40
NST	93.78	96.53	95.78	96.15
NST(feat)	94.00	96.89	96.02	96.45
NST(aug)	**94.55**	**97.28**	**96.96**	**97.12**
NST(feat+aug)	92.44	94.18	93.26	93.72

data. In all synthetic-augmentation configurations, the full real training set is used and the same number of additional synthetic samples is added. Thus, the total number of training images is identical across the synthetic variants, while the real-only baseline remains smaller and serves as the reference without synthetic augmentation. Synthetic data are used only during training, while validation and testing rely exclusively on real images. The following comparison assesses whether image refinement can improve the effectiveness of synthetic images as additional training data, and whether this effect is consistent across both acquisition setups.

Tables 6.2 and 6.3 summarize the classification results for $\mathcal{D}^{\mathrm{rl}}_{\mathrm{real}}$ and $\mathcal{D}^{\mathrm{tl}}_{\mathrm{real}}$ using extra synthetic images. For both datasets, the addition of raw synthetic images lowers accuracy and F1 score compared to the real-only

Table 6.3 Classification performance for transmitted-light granule images using extra synthetic training data.

Extra Training Data	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
None	87.78	88.47	86.00	87.22
Original synthetic	83.11	85.23	84.00	84.61
CycleGAN [322]	87.33	89.86	**88.67**	89.26
CUT [202]	86.89	87.88	87.33	87.60
NST	86.67	87.99	86.67	87.33
NST(feat)	87.23	88.89	87.33	88.10
NST(aug)	**88.24**	**90.13**	88.45	**89.28**
NST(feat+aug)	85.33	88.04	85.33	86.66

baseline. This indicates that the unrefined synthetic images do not sufficiently match the visual characteristics of the real data, which introduces inconsistencies in the training distribution and reduces generalization performance.

Refinement improves performance, although the degree of improvement varies between datasets and refinement methods. For synthetic granule images under the reflected-light illumination setup, several refined variants reach or exceed the real-only baseline. NST with augmentation-based alignment achieves the highest performance with an accuracy of 94.55% and an F1 score of 97.12%, providing a clear gain over the baseline. Feature-based alignment in NST also contributes positively, although not to the same extent as the augmentation-based alignment.

Synthetic granule images under the transmitted-light illumination setup exhibit the same qualitative trend but with smaller performance gains. For these synthetic images, NST with augmentation-based alignment again performs best, achieving an accuracy of 88.24% and an F1 score of 89.28%, which slightly exceeds the real-only baseline. CycleGAN [322] and CUT [202] provide moderate improvements over the raw synthetic data. CycleGAN [322] achieves the highest recall, although its overall performance remains below the NST-based configurations, which achieve higher accuracy and F1 scores. Raw synthetic images consistently reduce performance as well, confirming that refinement is essential for achieving reliable improvements.

Overall, these results show that synthetic augmentation is beneficial only when the synthetic data are closely aligned with the appearance of the real images. Refinement methods that adjust local texture, contrast, and illumination characteristics, as introduced in Section 4.1.1, help to reduce the domain gap and improve downstream classification accuracy. Across both datasets, NST with the augmentation-based alignment is the most effective refinement strategy.

6.2.2.3 Controlled Mixtures of Real and Synthetic Samples

This subsection analyzes the effect of varying the proportion of real and synthetic training samples while keeping the total number of training images fixed. As described in Section 6.2.1, the real training set is progres-

Table 6.4 Classification performance for reflected-light granule images using fixed-size mixtures of real and NST(aug) refined synthetic training data.

Real / Synthetic (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
100 / 0	93.22	96.14	95.23	95.68
75 / 25	**93.56**	**96.21**	**95.67**	**95.94**
50 / 50	92.89	94.82	93.36	94.08
25 / 75	91.56	93.23	92.38	92.80
0 / 100	86.23	88.56	88.03	88.29

sively replaced by refined synthetic samples, and the overall training set size remains constant across all configurations. Only the synthetic data refined using NST with augmentation-based alignment are used in this experiment for both datasets, since it achieves the highest accuracy and F1 score among the evaluated refinement variants for both datasets in Section 6.2.2.2. The mixture experiments assess whether refined synthetic data can function as a partial substitute for real training samples, rather than merely as supplementary data. Since the refinement step itself is guided by real images, this substitution should be understood at the level of downstream training samples rather than as a replacement of real data throughout the entire pipeline.

Tables 6.4 and 6.5 report the results for reflected-light granule images and transmitted-light granule images.

For reflected-light granule images, a moderate inclusion of synthetic data improves performance over the real-only baseline. The best result is obtained at a real-to-synthetic ratio of R75/S25, which achieves an accuracy of 93.56% and an F1 score of 95.94%. Increasing the proportion of synthetic data beyond this point leads to a gradual performance decrease, although mixtures with up to 50% synthetic data still remain close to the baseline. Training exclusively on synthetic data results in a clear drop in all metrics, indicating that real samples remain essential for reliable classification.

Transmitted-light granule images exhibit a different trend. The real-only configuration achieves the highest overall performance, and the inclusion of synthetic data does not provide additional gains. As the proportion of synthetic samples increases, accuracy and F1 score decrease steadily. When the training set consists entirely of synthetic images, per-

Table 6.5 Classification performance for transmitted-light granule images using fixed-size mixtures of real and NST(aug) refined synthetic training data.

Real / Synthetic (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
100 / 0	**87.78**	**88.47**	**86.00**	**87.22**
75 / 25	85.33	86.98	85.84	86.41
50 / 50	84.89	86.36	85.59	85.97
25 / 75	83.67	85.38	84.02	84.69
0 / 100	79.33	81.03	80.45	81.13

formance drops substantially, reaching an accuracy of 79.33% and an F1 score of 81.13%. These results suggest that the more complex appearance variations in transmitted-light granule images limit the effectiveness of synthetic substitution, even when refined data are used.

To better explain the mixture results, Fig. 6.2 visualizes t-distributed stochastic neighbor embedding (t-SNE) projections of the learned feature embeddings for three representative settings: real-only (R100/S0), a moderate mixture (R75/S25), and synthetic-only (R0/S100).

For reflected-light granule images, the real-only setting shows a substantial overlap between the real training and real test embeddings, which is consistent with a comparatively limited domain shift within the real data. When a moderate proportion of refined synthetic samples is introduced (R75/S25), many synthetic embeddings lie in regions that are already populated by real samples, while some extend the occupied space slightly. This distribution is consistent with the quantitative improvement observed at R75/S25 and suggests that, in this setting, the refined synthetic data may act as complementary samples that broaden the training distribution without strongly displacing the real one. In contrast, under synthetic-only training, the synthetic embeddings occupy a broader region and include areas that are only sparsely covered by real test samples. This larger mismatch in embedding space is consistent with the performance drop observed when real data are fully replaced.

For transmitted-light granule images, the real-only setting again shows that the real training and real test samples occupy overlapping regions in the embedding space. In the moderate mixture setting (R75/S25), the refined synthetic embeddings overlap with the real distribution only partially. Moreover, compared with the reflected-light case, the synthetic

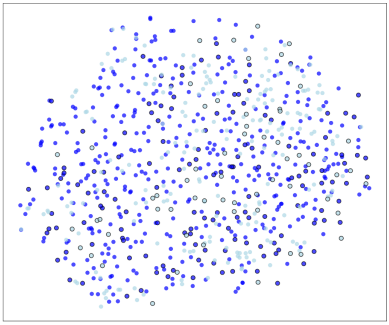
transmitted-light embeddings appear more concentrated in specific regions of the feature space rather than distributed across the broader region occupied by real samples. This pattern suggests that the synthetic data may emphasize a narrower subset of appearance characteristics instead of complementing the real distribution more uniformly. A similar tendency is visible in the synthetic-only setting, where larger parts of the synthetic embedding space remain only weakly connected to the real test distribution. Taken together, these visual patterns help explain why increasing the proportion of refined synthetic data is associated with a monotonic performance degradation for transmitted-light granule images.

Overall, this ablation study demonstrates that the optimal composition of real and synthetic training data is dataset dependent. A limited inclusion of refined synthetic samples can effectively complement real reflected-light granule images, whereas the same strategy does not generalize to transmitted-light granule images. This observation emphasizes the need for controlled mixture studies when assessing the impact of synthetic data in industrial classification tasks. In combination with the results reported in tables 6.2 and 6.3, the findings indicate that refined synthetic data improve classification performance primarily when used as a supplement to real samples, while extensive replacement of real data is detrimental if the synthetic and real embedding distributions are not well aligned.

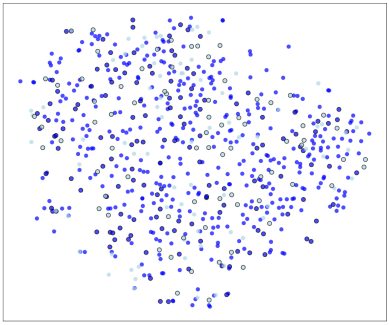
6.2.3 Summary of Classification Tasks

The classification experiments demonstrate that the effectiveness of synthetic data depends on both the refinement strategy and the manner in which synthetic samples are incorporated into the training set. In the synthetic-augmentation experiments, unrefined synthetic images consistently degrade performance, indicating that naive synthetic augmentation introduces distributional mismatches that outweigh potential gains in sample diversity. In contrast, refined synthetic data can provide measurable benefits, but only under specific conditions.

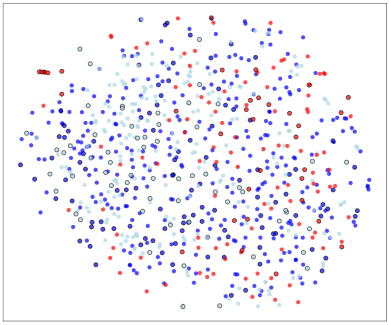
Among the evaluated refinement methods, NST with augmentation-based alignment yields the most consistent improvements in these experiments. Its effectiveness is consistent across both datasets when synthetic



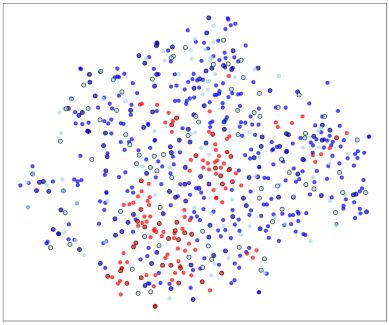
(a) reflected-light images: R100/S0



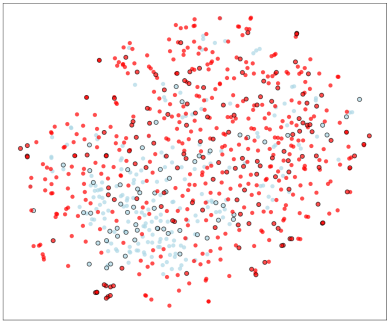
(b) transmitted-light images: R100/S0



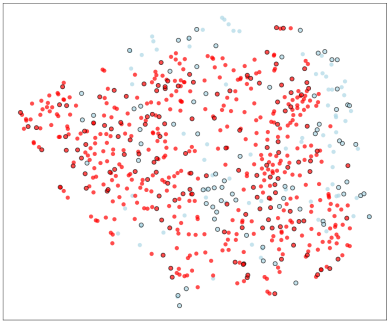
(c) reflected-light images: R75/S25



(d) transmitted-light images: R75/S25



(e) reflected-light images: R0/S100



(f) transmitted-light images: R0/S100

Figure 6.2 t-SNE visualization of feature embeddings under selected real/synthetic-mixture ratios. Blue points denote real training samples, light blue denotes real test samples, and red denotes synthetic training samples. Points with black contours correspond to defective samples.

data are added as supplementary training samples, confirming that refinement quality is a critical factor for successful synthetic augmentation. However, the degree of improvement varies between datasets, reflecting differences in imaging conditions and feature distributions.

The controlled mixture experiments further show that synthetic data should be treated as a complement rather than a replacement for real samples. For reflected-light granule images, a moderate inclusion of refined synthetic data improves performance, whereas for transmitted-light granule images, increasing the proportion of synthetic samples leads to a gradual decline. These findings indicate that the optimal real-to-synthetic ratio is dataset dependent and cannot be assumed a priori.

Overall, the classification results highlight two key points. First, refinement is essential for making synthetic data useful in practical settings. Second, even well-refined synthetic samples must be integrated cautiously, as their benefit depends on how well their feature distributions align with those of the real data. These insights motivate a more careful examination of synthetic data effects in tasks with higher sensitivity to domain mismatch, which is addressed in the following sections on segmentation and anomaly detection.

6.3 Crack Segmentation with Synthetic Supervision

This section investigates the use of synthetic data for crack segmentation, a task that requires precise pixel-level localization and is therefore more sensitive to domain mismatch than image classification. The experiments focus on training segmentation models exclusively with synthetic crack images and evaluating their performance on a fixed set of real test images. Different synthetic generation and refinement strategies model selection approaches are compared. This analysis provides insights into the limits of synthetic supervision for fine-grained segmentation tasks and the role of stability-based model selection under domain shift.

6.3.1 Task Description and Experimental Setup

The crack segmentation task aims to identify and localize crack regions at the pixel level. Each image is treated as a binary segmentation problem, where pixels are classified as either background or crack. Compared to image classification, this task places stronger requirements on local texture consistency, boundary accuracy, and spatial alignment between training and test data.

Due to the limited availability of real crack images, all segmentation models are trained using synthetic data only. Multiple synthetic crack datasets are considered, including raw synthetic images and several refined variants generated with different refinement strategies. Each synthetic dataset is treated independently and is split into a training set of 165 images and a validation set of 30 images. A fixed test set consisting of 15 real crack images is used for all experiments. As a result, the training and validation data are always drawn from the same synthetic domain, while evaluation is performed exclusively on real images.

All experiments employ a U-Net [227] architecture with a ResNet-18 [100] encoder. The input images are resized to a fixed resolution, and standard data augmentation techniques are applied during training. The optimization settings, loss function, and augmentation strategy are identical across all synthetic datasets to ensure a fair comparison. Model performance is evaluated using three metrics, including the background IoU, the foreground IoU for the crack class, and the mean IoU over both classes (*cf.* Appendix A.4).

Two model-selection strategies are examined. In the first setting, the checkpoint with the best performance on the synthetic-domain validation set is selected. This approach reflects a common practice when validation data are available, but it introduces a domain mismatch between model selection and final evaluation. In the second setting, the checkpoint is chosen using the embedding-stability-based early stopping strategy introduced in Chapter 5, so the synthetic-domain validation set is not used for model selection. For segmentation, the embedding is extracted from the final output of the ResNet-18 [100] encoder. Stability divergence is computed on the training set, and an adaptive threshold based on the initial divergence is applied. A checkpoint is recorded once the stability divergence remains below the threshold for a fixed number of consecu-

tive epochs. Other than these two model-selection strategies, the final model checkpoint after a fixed number of training epochs is also saved for the comparison.

This experimental design allows a direct comparison between validation-based and stability-guided checkpoint selection under domain shift, while also including the final checkpoint as a fixed-training reference. It therefore provides a controlled setting to assess the effectiveness of synthetic supervision for crack segmentation.

6.3.2 Experimental Results

This subsection presents the quantitative results of the crack segmentation experiments. The evaluation compares segmentation models trained on different synthetic crack datasets and assesses their ability to generalize to real test images. Two model selection strategies are examined. The first approach selects checkpoints based on validation data drawn from the same synthetic domain as the training set, reflecting a common practice when validation data are available. The second approach applies the embedding-stability-based early stopping strategy introduced in Chapter 5, which avoids reliance on synthetic-domain validation.

The results are organized in a way that highlights the influence of synthetic data refinement and checkpoint-selection strategy on foreground and background segmentation performance. In particular, the comparison focuses on how different refinement methods affect crack localization accuracy and how stability-guided checkpoint selection alters generalization behavior under domain shift. Together, these experiments provide a detailed view of the limits and potential of synthetic supervision for crack segmentation.

6.3.2.1 Validation-Based Model Selection

This subsection reports the crack segmentation performance obtained when model checkpoints are selected based on validation data drawn from the same synthetic domain as the training data. Each synthetic dataset is used independently, and the checkpoint achieving the best mean IoU on the corresponding synthetic validation set is evaluated on the fixed set of real test images.

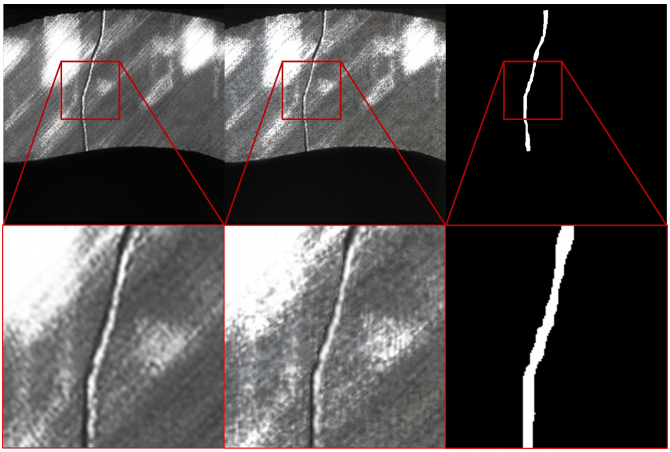
Table 6.6 Crack segmentation performance on real test images using validation-based model selection. Metrics are reported as background IoU, foreground IoU (crack), and mean IoU.

Synthetic Dataset	bg IoU (%)	fg IoU (%)	mean IoU (%)
Original synthetic	**99.81**	58.31	79.06
NST	99.68	28.69	64.19
CycleGAN [322]	99.80	**60.19**	**80.00**
CUT [202]	99.80	57.23	78.51
NST + Copy-Paste	99.78	53.84	76.81
NST + Soft-Freeze	99.80	58.17	78.98

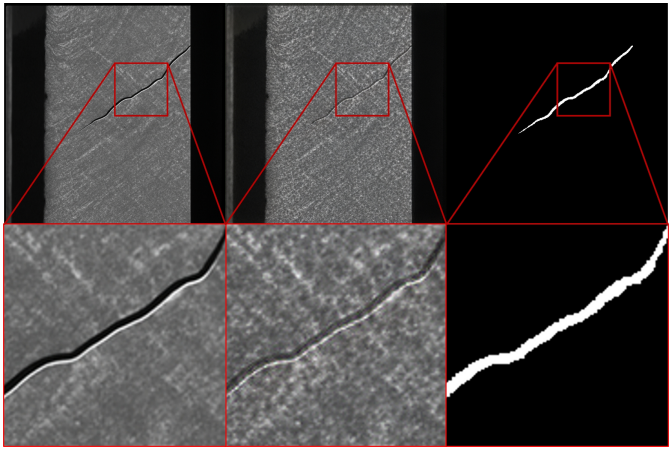
Table 6.6 summarizes the crack segmentation results. In both the synthetic training data and the real test images, crack pixels occupy less than 5 % of the image area on average, leading to a class imbalance. As a result, background IoU values are consistently high across all configurations and are dominated by the prevalence of background pixels rather than by the quality of crack modeling. Consequently, the following analysis focuses primarily on foreground IoU and mean IoU, which more accurately reflect crack localization performance.

Among all evaluated configurations, CycleGAN [322]-refined synthetic data achieve the highest foreground IoU of 60.19 % and highest mean IoU of 80.00 %, slightly outperforming both the original synthetic dataset and other refinement strategies. This suggests that CycleGAN [322] improves the appearance alignment between synthetic and real crack images while preserving crack geometry sufficiently for generalization to the real test data. CUT [202] and NST with Soft-Freeze alignment yield performance comparable to the original synthetic data, suggesting that moderate refinement neither degrades nor consistently improves crack segmentation performance.

In contrast, standard NST refinement results in a pronounced degradation of crack segmentation accuracy, reducing the foreground IoU to 28.69 %. NST with Copy-Paste also leads to a clear performance drop relative to the best-performing configurations. These results indicate that refinement strategies introducing artifacts can negatively affect general-



(a) NST with Copy-Paste introduces unnaturally sharp crack boundaries.



(b) Standard NST introduces discolored or textured pixels within crack regions.

Figure 6.3 Refinement-induced artifacts in synthetic training images. From left to right: original synthetic images, refined synthetic images with characteristic artifacts, and ground truth masks.

ization to real crack images, even when checkpoint selection is based on synthetic-domain validation.

To analyze the underlying causes of these performance differences, Fig. 6.3 presents qualitative examples of refinement-induced artifacts in the synthetic training data. For each refinement strategy, the figure shows original synthetic images generated by the GPT-image-1 model together with zoomed-in crack regions, followed by the corresponding refined synthetic images and their ground truth masks. As illustrated in Fig. 6.3(a), NST with Copy-Paste refinement introduces unnaturally sharp crack boundaries with abrupt intensity transitions. While such boundaries simplify crack localization in synthetic training and validation data, real cracks typically exhibit smoother transitions and locally ambiguous edges. In contrast, Fig. 6.3(b) shows that standard NST refinement introduces discolored or textured pixels within crack regions that are not present in real images.

The impact of these artifacts becomes evident when analyzing inference results on real test images. Fig. 6.4 shows representative failure cases for models trained on NST with Copy-Paste refined data and on standard NST refined data. For each example, the figure presents the real test image, the ground truth crack mask, and the predicted segmentation produced by the model trained on the respective synthetic dataset.

For NST with Copy-Paste refinement, the unnaturally sharp crack boundaries introduced during refinement may encourage the model to rely more strongly on edge patterns that are less common in real crack images. As shown in the third column of Fig. 6.4, this behavior is consistent with fragmented or incomplete crack predictions when the real cracks exhibit weak contrast or gradual intensity transitions. For standard NST refinement, the discolored and textured pixels introduced within synthetic crack regions may provide artificial visual cues that are associated with the crack class during training. When such patterns are absent in the real test images, the model response on true crack regions becomes less reliable, which is consistent with the missed detections and discontinuous predictions shown in the fourth column of Fig. 6.4.

Overall, the combined quantitative and qualitative results suggest that not all synthetic data refinement strategies are equally suitable for crack segmentation. While appearance-alignment methods such as CycleGAN [322] are associated with better generalization to real crack images, refinement approaches that introduce unrealistic crack boundaries or arti-

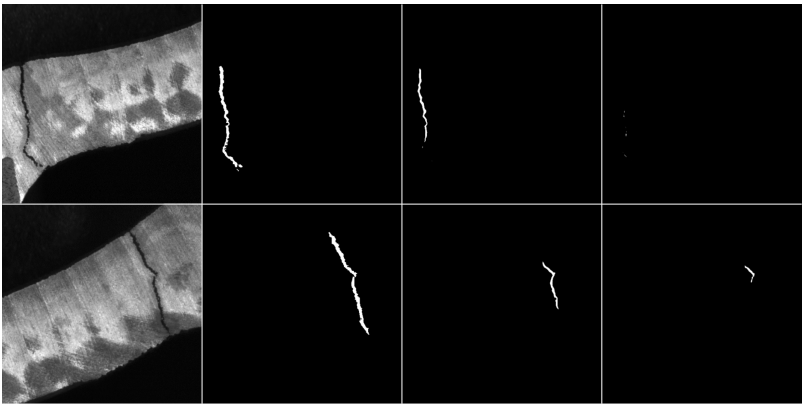


Figure 6.4 Failure cases on real test images caused by refinement-induced domain mismatch. Columns from left to right: real test image, ground truth mask, predicted segmentation of model trained on NST with Copy-Paste refined data, and predicted segmentation of model trained on standard NST refined data.

ficial textures within the crack region can increase the mismatch between synthetic training data and real test images. In this setting, such mismatch is associated with reduced pixel-level segmentation performance, which underlines the importance of evaluating refinement methods not only with quantitative metrics but also with qualitative inference results on real data.

6.3.2.2 Stability-Guided Model Selection

This subsection evaluates the embedding-stability-based checkpoint-selection strategy for crack segmentation and compares it with validation-based selection and the final training checkpoint. As introduced in Chapter 5, the stability-guided approach selects checkpoints based on the temporal consistency of learned embeddings and does not rely on validation data. This property is particularly relevant for the current setting, where validation images are drawn from the same synthetic domain as the training data and may therefore provide misleading guidance under domain shift. Following the terminology in Chapter 5, three check-

Table 6.7 Segmentation performance on the real crack test set for different checkpoint-selection strategies: synthetic-domain validation selection (Best-val), fixed final checkpoint (Fixed), and stability-guided early stopping (ES). Background, foreground, and mean IoU are reported in the format bg / fg / mean (in %).

Training data	Best-val	Fixed	ES
Original synthetic	99.81 / 58.31 / 79.06	99.80 / 56.48 / 78.14	99.82 / 60.29 / 80.05
NST	99.68 / 28.69 / 64.19	99.64 / 19.09 / 59.36	99.80 / 56.48 / 78.14
CycleGAN [322]	99.80 / 60.19 / 80.00	99.78 / 53.31 / 76.54	99.79 / 55.60 / 77.69
CUT [202]	99.80 / 57.23 / 78.51	99.80 / 55.31 / 77.55	99.80 / 57.41 / 78.60
NST + Copy-Paste	99.78 / 53.84 / 76.81	99.74 / 41.08 / 70.41	99.77 / 50.76 / 75.27
NST + Soft-Freeze	99.80 / 58.17 / 78.98	99.75 / 47.49 / 73.62	**99.83** / **64.42** / **82.12**
Mean	99.78 / 52.74 / 76.26	99.75 / 45.46 / 72.60	**99.80** / **57.49** / **78.65**

points are compared, including the checkpoint with the best validation (best-val) performance within the synthetic domain, the final checkpoint after a fixed number of training epochs, and the checkpoint selected by stability-guided early stopping. The stability divergence is computed from the final encoder output of the ResNet-18 [100] backbone, and the stopping rule requires $k = 5$ consecutive epochs below the adaptive threshold $\tau_{\text{ada}} = \beta \cdot S_1$ with $\beta = 0.01$ and EMA parameter $\alpha = 0.6$. The value of β is chosen empirically from preliminary stability-divergence curves and then kept fixed for all segmentation experiments.

Table 6.7 reports segmentation performance on the fixed set of 15 real test images for three checkpoints per synthetic dataset. As in the previous subsection, background IoU remains close to saturation across all configurations due to the strong class imbalance, and the analysis again focuses primarily on foreground IoU and mean IoU. For training sets that introduce clear domain artifacts, the checkpoint chosen by synthetic-domain validation can be misleading for real-data generalization. This behavior is most obvious for standard NST. The best-val checkpoint achieves foreground IoU of 28.69 % and mean IoU of 64.19 %, while the stability-selected checkpoint increases foreground IoU to 56.48 % and mean IoU to 78.14 %. The fixed checkpoint performs even worse in this case with only 19.09 % foreground IoU and 59.36 % mean IoU, which

suggests continued overfitting to synthetic-domain patterns rather than improved generalization to real cracks.

A similar pattern is observed for NST with Copy-Paste. The best-val checkpoint based on validation performance still yields the highest performance among the three in this case, but the fixed checkpoint drops to foreground IoU of 41.08 %. The early stopping checkpoint partially recovers performance with foreground IoU of 50.76 %. This is consistent with the earlier observation that Copy-Paste introduces unnaturally sharp crack boundaries in the synthetic domain, which may be amplified during continued training (*cf.* Fig. 6.3(a)), while stability-guided selection can stop before the model becomes overly specialized to this structure.

For refinement methods that already yield better cross-domain alignment, the differences between checkpoint strategies become smaller. The model trained on CycleGAN [322]-refined synthetic images achieves the highest mean IoU among the best-val checkpoints with 80.00 %, while the stability-selected checkpoint reaches mean IoU of 77.69 %. This suggests that, when the synthetic validation set is already a reasonable proxy for the real test distribution, the validation-selected checkpoint remains competitive. The model trained on CUT [202]-refined synthetic images shows minor differences across checkpoints, and the stability-selected checkpoint yields a small improvement over the best-val checkpoint.

The clearest benefit of stability-guided selection is observed for NST with Soft-Freeze. The early stopping checkpoint achieves the overall best mean IoU of 82.12 % and the highest foreground IoU of 64.42 %. In this setting, the soft-freeze constraint reduces harmful texture changes while preserving crack appearance, and stability-guided selection further avoids overfitting.

Beyond individual training configurations, the mean values reported in the last row provide an aggregated view of checkpoint-selection performance across all synthetic refinement strategies. On average, early stopping achieves the highest mean IoU, outperforming both best-val and fixed checkpoint selection. This improvement is primarily driven by consistently higher foreground IoU, while background IoU remains nearly saturated for all strategies due to the strong class imbalance in crack segmentation. These results suggest that stability-guided early stopping provides the most robust checkpoint-selection strategy among

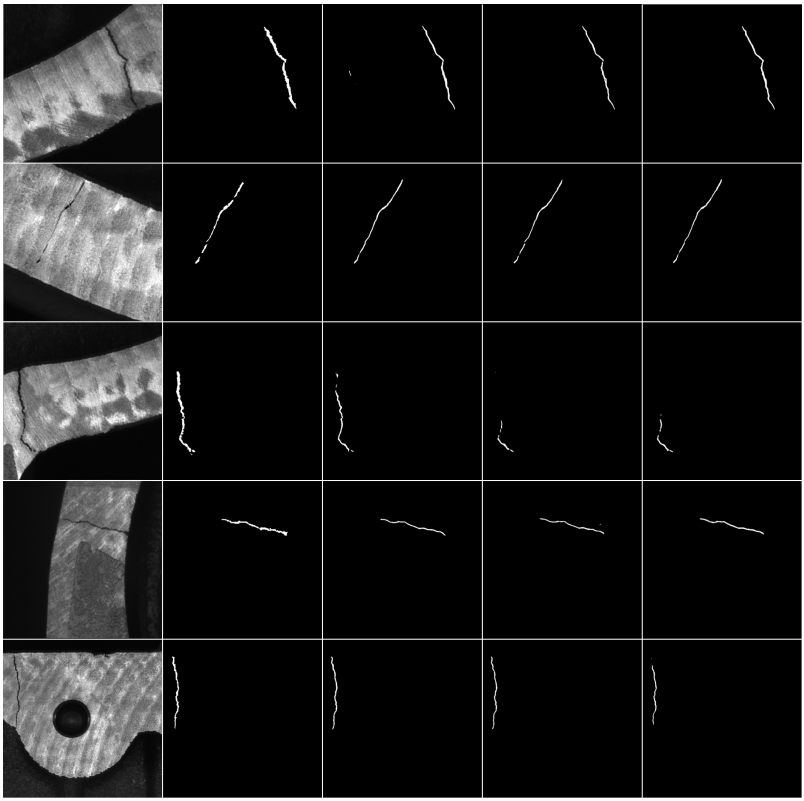


Figure 6.5 Qualitative comparison of checkpoint selection for CycleGAN [322] refinement.

the compared options across these synthetic training conditions, even when individual refinement methods exhibit varying levels of domain alignment.

Figs. 6.5 to 6.7 present qualitative comparisons of checkpoint-selection strategies for three representative training datasets. Each figure shows the inference results for five real test images.

For CycleGAN-refined training data (*cf.* Fig. 6.5), all three checkpoint-selection strategies produce visually similar predictions. Crack structures

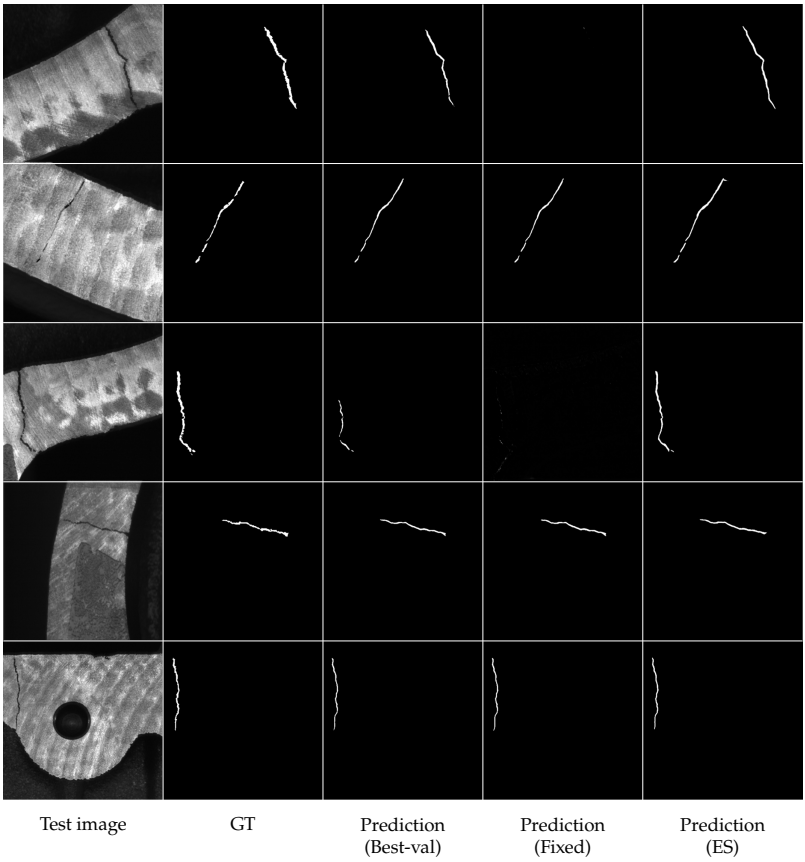


Figure 6.6 Qualitative comparison of checkpoint selection for NST with Soft-Freeze refinement.

are generally continuous, and missing segments are limited across best-val, fixed, and early stopping checkpoints. This visual consistency aligns with the quantitative results in Table 6.7, where CycleGAN shows relatively small performance gaps between checkpoint choices. The strong alignment between refined synthetic images and real crack appearance reduces the sensitivity to checkpoint selection, making validation-based selection relatively reliable in this case.

In contrast, the model trained using NST with Soft-Freeze refinement (*cf.* Fig. 6.6) exhibits clear differences between checkpoint strategies. The fixed checkpoint produces fragmented or incomplete crack predictions, while the best-val checkpoint improves continuity but still misses fine segments in several cases. The early stopping checkpoint appears to preserve longer crack traces and recover thin structures that are absent in the other two settings. These observations also correspond closely to the substantial foreground IoU gains achieved by early stopping for this refinement method in Table 6.7.

The most pronounced differences appear for standard NST refinement (*cf.* Fig. 6.7). Here, the best-val checkpoint often produces incomplete or discontinuous crack masks, which is consistent with overfitting to synthetic validation patterns. The fixed checkpoint further amplifies this behavior, frequently missing large portions of the crack. In contrast, the early stopping checkpoint recovers substantially more continuous crack structures and reduces false negatives across most examples.

Overall, the results indicate that checkpoint selection is a non-negligible factor in synthetic-to-real crack segmentation. Validation-based selection within the synthetic domain can be reliable when the refined images are well aligned with real appearance, as in CycleGAN. In contrast, when refinement introduces domain-specific artifacts that make the synthetic validation task easier than the real test task, stability-guided early stopping mitigates overfitting and yields markedly improved real-test segmentation performance.

6.4 Anomaly Detection Across Synthetic and Real Domains

This section investigates the impact of synthetic training data and checkpoint-selection strategies on industrial visual anomaly detection. Compared to classification and segmentation, anomaly detection operates under more restrictive conditions, as training is typically performed using only normal samples and without access to labeled validation data. These constraints make anomaly detection particularly sensitive to both training data distribution and model convergence behavior.

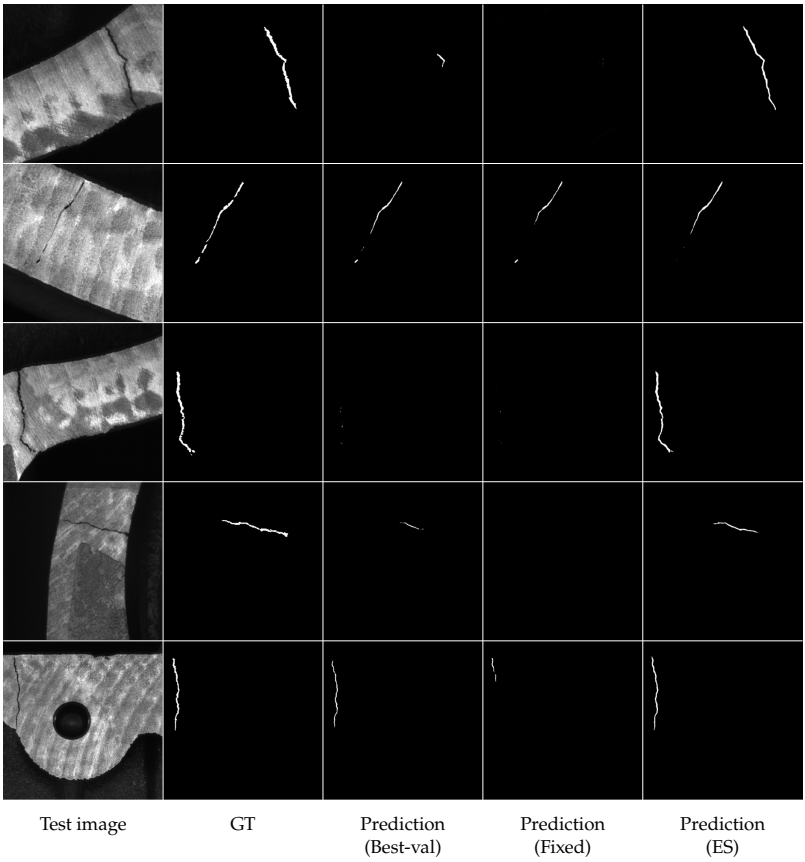


Figure 6.7 Qualitative comparison of checkpoint selection for standard NST refinement.

The experiments in this section evaluate anomaly detection performance on granule and crack inspection tasks using GLASS [44] and SimpleNet [162]. For granule datasets, the analysis considers real-only training, augmentation with synthetic normal images, and controlled mixtures with fixed total training size. For crack inspection, where no synthetic data are used, the focus is placed on checkpoint-selection behavior under purely real training conditions. Across all settings, performance is assessed at both image and pixel levels, and special attention is given

to the role of stability-guided early stopping as a practical alternative to fixed-length training.

6.4.1 Task Description and Experimental Setup

This subsection describes the anomaly detection tasks, datasets, models, and evaluation protocols used throughout Section 6.4. All experiments follow a validation-free training paradigm, which reflects common practice in industrial anomaly detection (*cf.* Section 2.5.3) and aligns with the original experimental settings of GLASS [44] and SimpleNet [162].

Anomaly detection experiments are conducted on three datasets: the granule dataset $\mathcal{D}^{\mathrm{rl}}_{\mathrm{real}}$ under the reflected-light illumination setup, the granule dataset $\mathcal{D}^{\mathrm{tl}}_{\mathrm{real}}$ under the transmitted-light illumination setup, and the crack dataset $\mathcal{D}^{\mathrm{crack}}_{\mathrm{real}}$. For the granule datasets, the task is to detect visually defective granules using only normal samples during training. For the crack dataset, training is likewise performed using only real crack-free images, while both normal and defective samples are included in the test set.

For $\mathcal{D}^{\mathrm{rl}}_{\mathrm{real}}$, the training set consists of 1000 real normal images. The test set contains 1000 real normal and 200 real defective images. For $\mathcal{D}^{\mathrm{tl}}_{\mathrm{real}}$, 520 real normal images are used for training, and the test set contains 225 real normal and 225 real defective images. The test sets of both granule datasets are formed by combining the original validation and test splits, since no validation set is allowed in our anomaly detection setting. In the crack dataset, only real crack-free images are available for training, and all evaluations are performed on real test images. Hence, 160 out of 195 crack-free images are used as training data, and the remaining 35 crack-free images, together with the 15 images with cracks, form the test set.

For the granule datasets, three training regimes are considered. First, a real-only baseline uses all available real normal images for training. Second, additional synthetic normal images are introduced as extra training data, with the number of synthetic normal samples matching the number of real normal training samples. Third, controlled mixture experiments are conducted in which the total number of training images is fixed, and the proportion of real and synthetic samples is varied. No

defective samples are included in training for any anomaly detection experiment. For the crack dataset, only the real-only training regime is evaluated, as no synthetic crack data are used in this task. This differs from the segmentation experiments in Section 6.3, where synthetic crack images are used for supervised mask training.

Two anomaly detection models are evaluated: GLASS [44] and SimpleNet [162]. Both models are trained using their official implementations and default hyperparameter settings, including optimizer configuration, learning rate schedules, data augmentation strategies, and training duration. No task-specific tuning is applied beyond the checkpoint-selection strategies described below.

Three checkpoint-selection strategies are reported for comparison. The fixed checkpoint corresponds to the model state obtained at the final training epoch. The ES checkpoint is selected using the stability-guided criterion introduced in Chapter 5, without access to labeled validation data. In addition, a best-test checkpoint is reported as an oracle reference by selecting the checkpoint that achieves the highest test-set performance during training, following the convention in GLASS [44] and SimpleNet [162]. This oracle setting is included solely to indicate an upper bound and is not considered a practical or deployable strategy.

Anomaly detection performance is evaluated using image-level and pixel-level area under the ROC curve. Image AUROC measures the ability to distinguish normal and defective samples, while pixel AUROC evaluates localization performance by comparing predicted anomaly maps to pixel-level ground truth. More details about the applied metrics for anomaly detection are provided in Appendix A.5. All reported results are computed on real test images.

6.4.2 Anomaly Detection on Granule Datasets

This subsection presents anomaly detection results on the granule datasets under different training regimes and checkpoint-selection strategies. The analysis focuses on how the inclusion of synthetic normal data and its refinement affect anomaly detection performance, as well as how early stopping influences model generalization in the absence of labeled validation data. Results are reported separately for

Table 6.8 Baseline anomaly detection performance on reflected-light granule images using only real normal training data. Image-level AUROC is reported in %.

Model	Best-test	Fixed	ES
GLASS	98.24	95.09	95.05
SimpleNet	96.40	92.15	93.52

reflected-light and transmitted-light granule images to account for their different imaging conditions and visual characteristics.

6.4.2.1 Baseline Performance Using Real Training Data

This subsection reports baseline anomaly detection performance on the granule datasets using only real normal images for training. The results serve as reference points for later comparisons involving synthetic data augmentation and controlled mixture strategies. All results follow the validation-free setting described in Section 6.4.1. Alongside deployable checkpoint strategies, oracle results obtained by selecting the best-performing checkpoint on the test set are reported as upper bounds.

Baseline results for reflected-light granule images are summarized in Table 6.8. Both GLASS [44] and SimpleNet [162] achieve strong anomaly detection performance when trained solely on real normal samples. GLASS attains higher image-level AUROC than SimpleNet for all checkpoint strategies, suggesting more stable separation between normal and defective samples. For reflected-light granule images, the differences between fixed and early stopping checkpoints are comparatively small for both models, suggesting that training dynamics are more stable under reflected-light imaging conditions when only real normal data are used.

Baseline results for transmitted-light granule images are shown in Table 6.9. Compared to reflected-light granule images, anomaly detection performance is substantially lower for both models due to the more challenging transmitted-light imaging conditions (*cf.* Fig. 6.1), which introduce strong reflections and low-contrast regions that complicate the modeling of normal appearance. In this setting, checkpoint selection becomes more consequential. For both models, early stopping improves over the fixed checkpoint at the image level, indicating that stability-

Table 6.9 Baseline anomaly detection performance on transmitted-light granule images using only real normal training data. Image-level AUROC is reported in %.

Model	Best-test	Fixed	ES
GLASS	72.28	50.16	51.51
SimpleNet	83.36	70.14	73.17

guided selection provides a more robust alternative to fixed-length training under transmitted-light conditions.

Taken together, the baseline results highlight the strong influence of imaging conditions on anomaly detection performance and confirm that checkpoint selection is a non-negligible factor in validation-free training, particularly for transmitted-light granule inspection.

6.4.2.2 Influence of Synthetic and Refined Synthetic Training Data

This subsection examines how the inclusion of synthetic normal data affects anomaly detection performance on the granule datasets. The analysis compares real-only training with augmentation using synthetic data generated by different refinement strategies. For anomaly detection, the key issue is whether refined synthetic normal data improve deployable performance, rather than only oracle performance. Results are reported for both GLASS [44] and SimpleNet [162], and checkpoint selection effects are discussed alongside each comparison.

Results for reflected-light granule images are summarized in Tables 6.10 and 6.11. For GLASS, adding synthetic normal data without refinement leads to a clear degradation in image-level AUROC for deployable checkpoints. Refinement strategies mitigate this effect to varying degrees. Among all variants, NST with augmentation-based alignment yields the strongest practical performance, with early stopping achieving the highest image-level AUROC. When combined with stability-guided early stopping, the NST-based variants with feature- or augmentation-based alignment generally yield higher deployable image-level AUROC than the other refinement methods.

Fig. 6.8 visualizes the effect of refinement when the oracle checkpoint is used for each training setting. On the normal samples in the top three

Table 6.10 Effect of synthetic and refined synthetic data on anomaly detection performance for reflected-light granule images using GLASS. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Extra training data	Best-test	Fixed	ES
None	98.24	95.09	95.05
Original synthetic	97.77	91.41	91.43
CUT	98.32	89.83	94.88
CycleGAN	98.29	95.11	95.22
NST	97.93	86.82	95.15
NST (feat)	97.81	92.43	95.83
NST (aug)	97.94	95.82	**96.20**
NST (feat+aug)	97.82	93.33	95.72

rows, the model trained using CycleGAN-refined images shows an elevated background response compared to the other models. NST with augmentation-based alignment also exhibits structure-dependent responses. On the defective samples in the bottom three rows, localized high-intensity activations appear consistently near the visible burn marks across most columns except for the model trained using the original synthetic images without refinement, which shows weaker activations in the defect regions. Models trained using no synthetic data, CUT-refined data

Table 6.11 Effect of synthetic and refined synthetic data on anomaly detection performance for reflected-light granule images using SimpleNet. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Extra training data	Best-test	Fixed	ES
None	96.40	92.15	93.52
Original synthetic	96.81	86.99	89.64
CUT	96.00	95.75	95.83
CycleGAN	97.90	94.82	**96.82**
NST	96.26	79.14	79.58
NST (feat)	96.87	79.82	87.32
NST (aug)	96.70	82.65	94.91
NST (feat+aug)	96.59	83.89	94.93

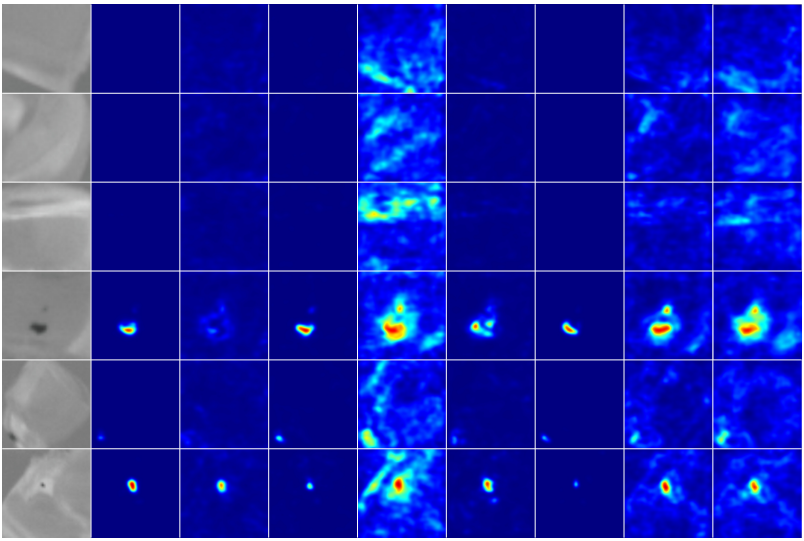


Figure 6.8 Qualitative comparison for GLASS on reflected-light granule images using different extra training data. All heatmaps in this figure are generated using the oracle checkpoint for each training setting.

and NST-refined data show compact hotspots concentrated around the defect region, whereas models trained using either CycleGAN-refined data or data refined using NST with augmentation-based pairing show broader and more diffuse activations. These visual differences are consistent with the quantitative trends in Table 6.10. As with the later qualitative visualizations in this chapter, these examples should be understood as illustrative cases that complement the corresponding quantitative results rather than as exhaustive evidence on their own.

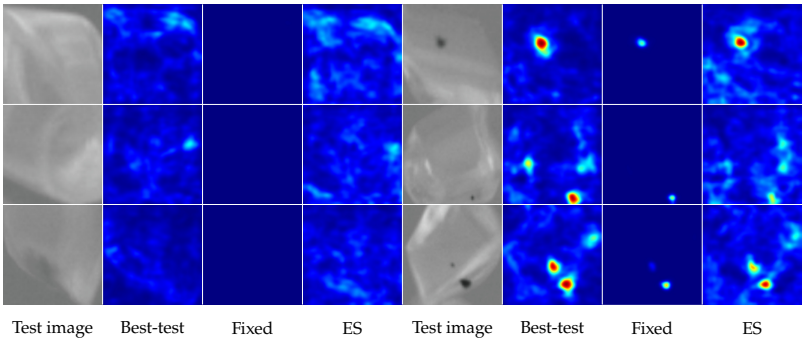


Figure 6.9 Qualitative comparison of checkpoint selection for GLASS on reflected-light granule images using NST (aug) refined synthetic training data.

Fig. 6.9 isolates checkpoint effects for the model trained using synthetic images refined with NST with augmentation-based pairing as augmentation data. For the normal samples (left), the fixed checkpoint produces near-zero responses across the full image, while the oracle and early stopping checkpoints show weak, spatially structured background activations. For the defective samples (right), the oracle and early stopping checkpoints yield distinct localized hotspots at the defect locations, whereas the fixed checkpoint largely collapses to relatively small responses that do not cover the defect region. This qualitative behavior is consistent with Table 6.10, where early stopping slightly improves over the fixed checkpoint for the model trained using extra synthetic images refined with NST with augmentation-based pairing.

For SimpleNet, the effect of synthetic data is more sensitive to the refinement method. Among all variants, CycleGAN [322]-refined synthetic data yields the strongest practical improvement, achieving the highest image-level AUROC under early stopping. In contrast, standard NST causes a pronounced performance degradation for both fixed and early stopping checkpoints. CUT improves over unrefined synthetic data for deployable checkpoints but remains inferior to CycleGAN in this setting. Meanwhile, both NST variants with augmentation-based alignment also improve the utility of synthetic images as augmentation data, resulting in higher image-level AUROC than using real data only for training.

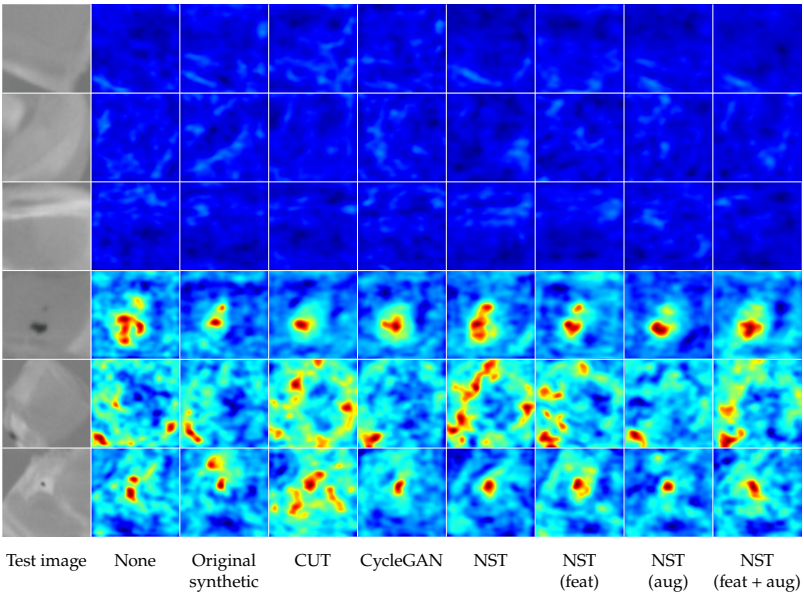


Figure 6.10 Qualitative comparison for SimpleNet on reflected-light granule images using different extra training data. All heatmaps in this figure are generated using the oracle checkpoint for each training setting.

Fig. 6.10 illustrates the influence of refinement under oracle checkpoint selection. On the normal samples, all models show relatively uniform low-intensity heatmaps with slightly different spatial structure. For the defective samples, localized high-intensity responses appear near the visible defect regions across most refinement strategies. Among them, models trained using CycleGAN- and augmentation data refined with NST with augmentation-based pairing appear to show the most compact and well-aligned activations in the defect regions. Models trained using either the original synthetic images or synthetic images refined using NST with feature-based pairing can produce well-localized activations for defect regions on some test samples, but they also have a higher false positive rate by showing broader activations than the actual defect regions for some other test samples, such as the example in the fifth row.

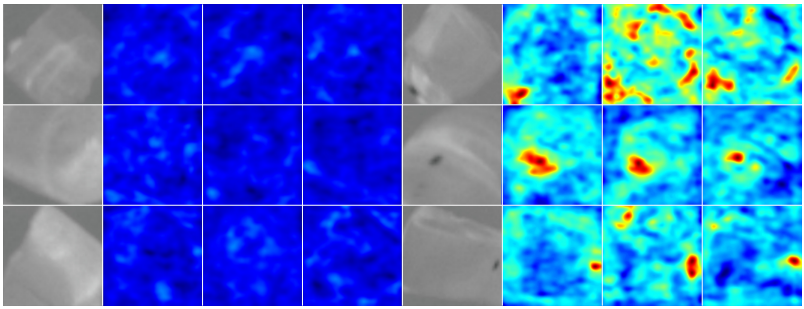


Figure 6.11 Qualitative comparison of checkpoint selection for SimpleNet on reflected-light granule images using CycleGAN-refined synthetic training data.

Fig. 6.11 focuses on checkpoint effects for the model trained using CycleGAN-refined synthetic augmentation data, as it shows the highest deployable image-level AUROC among all models. For normal samples, all three checkpoints produce similarly suppressed and irregular background activations. For defective samples, the oracle and early stopping checkpoints generally generate localized hotspots at the defect locations, whereas the fixed checkpoint yields more diffuse activations with a higher false positive rate.

Results for transmitted-light granule images are summarized in Tables 6.12 and 6.13. Compared to reflected-light granule images, anomaly detection performance under the transmitted-light illumination is substantially lower, and the gap between deployable checkpoint strategies becomes markedly larger. The strong reflections and low-contrast regions characteristic of transmitted-light granule images increase the difficulty of modeling the normal appearance distribution and amplify the impact of both refinement quality and checkpoint selection. For GLASS, raw synthetic data provide only limited benefit for deployable checkpoints, despite moderate improvements in oracle performance. NST-based methods outperform CycleGAN and CUT under early stopping. Among all variants, NST with augmentation-based alignment achieves the highest deployable image-level AUROC. Fixed checkpoints perform poorly across most refinement strategies, highlighting strong sensitivity of GLASS to

Table 6.12 Effect of synthetic and refined synthetic data on anomaly detection performance for transmitted-light granule images using GLASS. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Extra training data	Best-test	Fixed	ES
None	72.28	50.16	51.51
Original synthetic	76.66	48.58	52.92
CUT	80.90	55.85	71.08
CycleGAN	82.05	62.79	71.49
NST	84.87	64.77	75.62
NST (feat)	86.73	71.81	76.47
NST (aug)	84.20	71.71	**78.25**
NST (feat+aug)	82.12	64.51	74.80

training dynamics under transmitted-light conditions. Early stopping substantially mitigates this degradation and enables GLASS to exploit refined synthetic data more effectively.

The influence of refinement under oracle checkpoint selection is illustrated in Fig. 6.12. On the normal samples, the models trained using real data only, original synthetic data, and CUT-refined synthetic data exhibit widespread low-to-medium intensity responses across large regions of the granules, indicating weak suppression of background acti-

Table 6.13 Effect of synthetic and refined synthetic data on anomaly detection performance for transmitted-light granule images using SimpleNet. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Extra training data	Best-test	Fixed	ES
None	83.36	70.14	73.17
Original synthetic	76.90	56.26	63.61
CUT	83.36	66.15	73.07
CycleGAN	78.70	70.81	70.81
NST	78.08	59.61	66.68
NST (feat)	83.24	57.55	68.75
NST (aug)	88.76	61.41	**80.03**
NST (feat+aug)	87.86	77.45	74.78

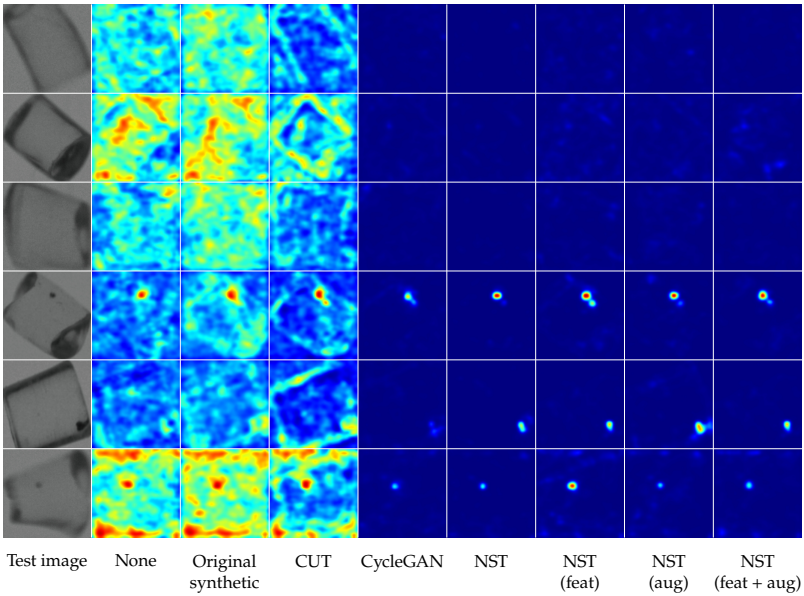


Figure 6.12 Qualitative comparison for GLASS on transmitted-light granule images using different extra training data. All heatmaps in this figure are generated using the oracle checkpoint for each training setting.

vations under transmitted-light conditions. In contrast, the other models show comparatively lower and more spatially uniform responses on normal samples. For the defective samples, the models trained using real data only, original synthetic data, and CUT-refined synthetic data show broader and more diffuse activations extending beyond the actual defect region. The models trained using synthetic augmentation data refined with NST with feature-based pairing tend to yield the most precise prediction of the defect region, which is consistent with its highest oracle AUROC in Table 6.12.

Fig. 6.13 highlights the effect of checkpoint selection for the model trained using NST (aug)-refined synthetic augmentation data. For normal samples, the early stopping checkpoint produces slightly more elevated background responses, while both the oracle and fixed checkpoints suppress these responses more effectively, yielding darker and more ho-

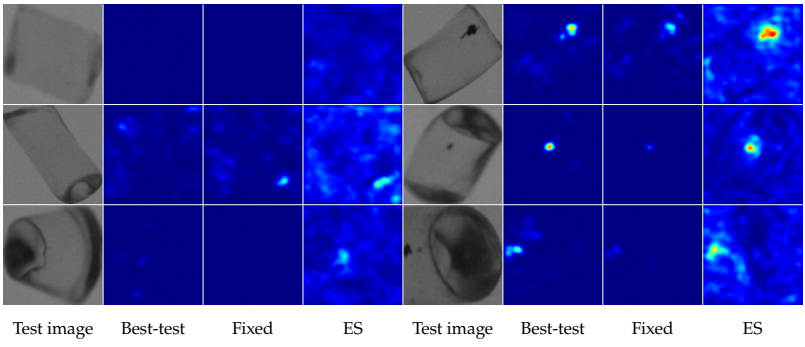


Figure 6.13 Qualitative comparison of checkpoint selection for GLASS on transmitted-light granule images using NST (aug)-refined synthetic training data.

mogeneous heatmaps. However, for defective samples, the oracle and early stopping checkpoints generate clear and localized hotspots at the defect locations, whereas the fixed checkpoint yields much weaker and less concentrated responses.

For SimpleNet, NST with augmentation-based alignment again provides the strongest practical performance, with early stopping achieving the highest image-level AUROC. Several refinement strategies that perform well under oracle checkpoint selection exhibit substantially reduced performance for deployable checkpoints, indicating increased sensitivity to both refinement-induced appearance changes and training instability. Compared to GLASS, SimpleNet shows smaller absolute gains from refinement, but early stopping remains important for maintaining robust performance in the transmitted-light setting.

The oracle-refinement comparison in Fig. 6.14 reveals clear differences in how refinement strategies affect the spatial distribution of anomaly responses. On the normal samples, the behavior is similar to that in Fig. 6.10, where all models show relatively uniform low-intensity heatmaps with slightly varying spatial structure. For the defective samples, refinement-dependent differences are more pronounced. Models trained with augmentation data refined by the NST variants with feature- or augmentation-based alignment show localized activations around the actual defect regions. By contrast, models trained using

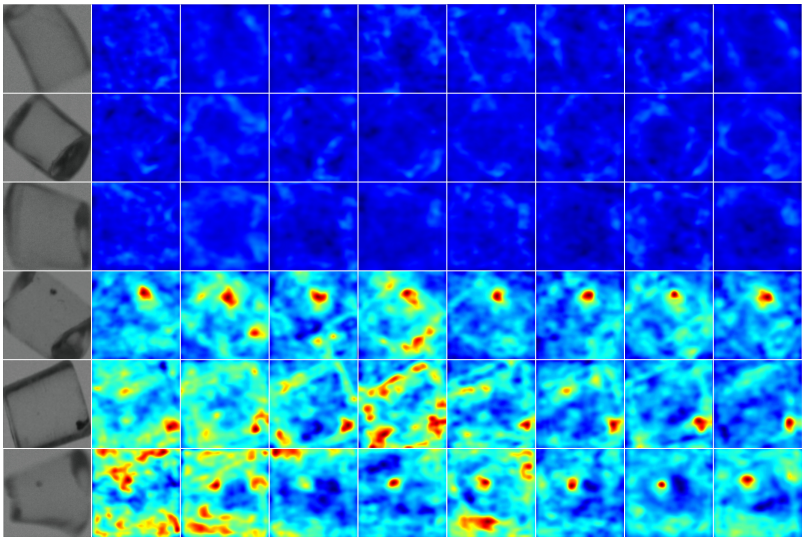


Figure 6.14 Qualitative comparison for SimpleNet on transmitted-light granule images using different extra training data. All heatmaps in this figure are generated using the oracle checkpoint for each training setting.

original and CycleGAN-refined synthetic data exhibit a higher false positive rate, with additional activations in normal regions. These visual patterns correspond closely to the quantitative results in Table 6.13.

The effect of checkpoint selection for the model trained using NST (aug)-refined synthetic augmentation data is illustrated in Fig. 6.15. For normal samples, all three checkpoints produce slightly higher activations on the boundary of granules, whereas the activations in other regions remain low. For defective samples, the fixed checkpoint often generates widespread false responses in normal regions, while the early stopping checkpoint produces more distinct and spatially concentrated hotspots that more closely resemble the oracle output.

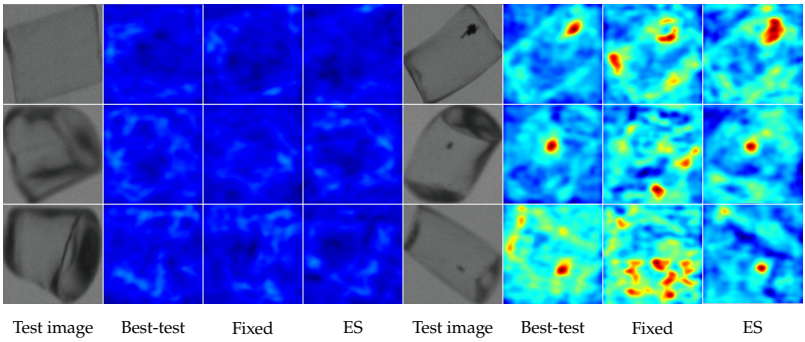


Figure 6.15 Qualitative comparison of checkpoint selection for SimpleNet on transmitted-light granule images using NST (aug)-refined synthetic training data.

6.4.2.3 Controlled Mixtures of Real and Synthetic Samples

This subsection analyzes anomaly detection performance under controlled mixtures of real and synthetic normal training data, where the total number of training samples is fixed and the proportion of real and synthetic data is varied. This setting isolates the effect of substituting real data with synthetic data and provides a more stringent evaluation of robustness under distribution shift. As in the previous subsections, results are reported for oracle, fixed, and early stopping checkpoints. The fixed-total mixture setting provides a stricter test by evaluating whether refined synthetic data can replace real normal samples under a controlled training budget.

Results for reflected-light granule images are shown in Tables 6.14 and 6.15. For GLASS, anomaly detection performance remains high across all mixture ratios when NST (aug) refined synthetic data are used. However, the results show a gradual decrease in image-level AUROC as the proportion of synthetic data increases. This indicates that, for reflected-light granule images, refined synthetic samples are effective as a supplement to real data but cannot fully replace real training samples without incurring a loss of performance.

Across all mixture ratios, early stopping and fixed checkpoints yield similar image-level AUROC for reflected-light granule images, with only

Table 6.14 Fixed-total mixture analysis for reflected-light granule images using GLASS with NST (aug) refined synthetic data. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Real / Synthetic (%)	Best-test	Fixed	ES
100 / 0	98.24	95.09	95.05
75 / 25	98.28	95.04	**95.15**
50 / 50	98.19	94.89	95.08
25 / 75	97.93	94.32	94.95
0 / 100	97.58	93.21	94.36

small differences between the two. This suggests that, under reflected-light imaging conditions, the training dynamics of GLASS are comparatively stable even when real samples are partially replaced by refined synthetic data.

For SimpleNet, the effect of synthetic substitution is more limited. Using CycleGAN-refined synthetic data, early stopping yields image-level AUROC close to the real-only baseline for moderate mixture ratios, while fixed checkpoints degrade more clearly as the proportion of synthetic data increases. Under fully synthetic training, the fixed checkpoint shows a noticeable performance drop, whereas early stopping mitigates part of this degradation at the image level.

Results for transmitted-light granule images are shown in Tables 6.16 and 6.17. For GLASS, oracle results indicate that introducing refined synthetic data can increase the achievable performance compared to the

Table 6.15 Fixed-total mixture analysis for reflected-light granule images using SimpleNet with CycleGAN-refined synthetic data. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Real / Synthetic (%)	Best-test	Fixed	ES
100 / 0	96.40	92.15	**93.52**
75 / 25	96.42	91.85	93.48
50 / 50	96.36	92.02	93.50
25 / 75	95.98	88.52	92.18
0 / 100	96.19	87.82	93.12

Table 6.16 Fixed-total mixture analysis for transmitted-light granule images using GLASS with NST (aug) refined synthetic data. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Real / Synthetic (%)	Best-test	Fixed	ES
100 / 0	72.28	50.16	51.51
75 / 25	75.54	53.25	63.29
50 / 50	77.51	54.20	**68.38**
25 / 75	76.32	52.35	67.94
0 / 100	71.84	49.34	64.21

real-only setting, with the strongest oracle values observed at moderate mixture ratios. For deployable checkpoint strategies, early stopping consistently outperforms the fixed checkpoint across all mixture ratios at the image level, indicating that stability-guided checkpoint selection is important for robust anomaly detection under synthetic substitution.

As the proportion of synthetic data increases, fixed checkpoints remain at a consistently low level, while early stopping yields a substantial gain and maintains relatively stable performance across mixture ratios. Even when training is performed using only refined synthetic data, early stopping preserves markedly higher image-level AUROC than fixed checkpoints. These results indicate that, for transmitted-light granule images, replacing real training data with refined synthetic samples can be feasible, but only when checkpoint selection is explicitly controlled under the validation-free setting.

Table 6.17 Fixed-total mixture analysis for transmitted-light granule images using SimpleNet with NST (aug) refined synthetic data. Image-level AUROC is reported in %. The best result for the deployable checkpoint is shown in bold, while Best-test is reported only as an oracle reference.

Real / Synthetic (%)	Best-test	Fixed	ES
100 / 0	83.36	70.14	73.17
75 / 25	85.34	71.43	73.46
50 / 50	86.23	74.65	**77.82**
25 / 75	83.75	71.46	75.43
0 / 100	82.08	69.83	73.66

In contrast to GLASS, both oracle and deployable performance values are higher for SimpleNet in this dataset, which is consistent with the refinement experiment in Table 6.13. The oracle results peak at moderate mixture ratios, indicating that a limited substitution of real samples by refined synthetic data can improve the achievable performance for transmitted-light granule images.

For deployable checkpoint strategies, early stopping again consistently outperforms the fixed checkpoint for all mixture ratios. The performance gap remains substantial even when a majority of real samples is retained, and early stopping yields more stable performance as the synthetic proportion increases. Even under fully synthetic training, early stopping maintains higher image-level AUROC than fixed checkpoints, indicating improved robustness to synthetic substitution under transmitted-light conditions.

Overall, the fixed-total mixture analysis shows that the effect of replacing real training data with synthetic data is strongly dataset- and model-dependent. For reflected-light granule images, performance generally decreases as the synthetic proportion increases, whereas transmitted-light granule images benefit from moderate synthetic substitution when stability-guided checkpoint selection is used. Early stopping consistently mitigates degradation from fixed-length training and provides more stable image-level performance across mixture ratios, particularly under transmitted-light imaging conditions. These findings highlight the importance of stability-aware training strategies when synthetic data are used to substitute real samples in anomaly detection tasks.

6.4.3 Anomaly Detection on Crack Dataset

This subsection presents anomaly detection results on the crack inspection dataset under real-only training conditions. In contrast to the granule anomaly detection experiments and to the crack segmentation experiments, no synthetic data are used in this task, which allows the analysis to focus exclusively on model convergence behavior and checkpoint selection under validation-free training. Results are reported for GLASS [44] and SimpleNet [162] using both image-level and pixel-level AUROC.

Table 6.18 Anomaly detection performance on the crack dataset using only real crack-free training data. Image-level and pixel-level AUROC are reported in the format image / pixel (in %).

Model	Best-test	Fixed	ES
GLASS	96.67 / 98.82	84.00 / 96.47	92.00 / 96.98
SimpleNet	94.67 / 95.16	77.67 / 93.21	84.67 / 96.31

6.4.3.1 Results and Checkpoint-Selection Behavior

Baseline anomaly detection performance on the crack dataset is summarized in Table 6.18. Both models are trained using only real crack-free images, and no labeled validation set is available. Alongside deployable checkpoint strategies, oracle results obtained by selecting the best-performing checkpoint on the test set are reported as upper bounds.

Checkpoint selection has a pronounced impact on practical performance. At the image level, early stopping improves AUROC from 84.00% to 92.00% for GLASS and from 77.67% to 84.67% for SimpleNet. At the pixel level, early stopping yields a smaller improvement for GLASS, from 96.47% to 96.98%, and a more substantial improvement for SimpleNet, from 93.21% to 96.31%.

To better understand these differences, qualitative inference results are shown in Figs. 6.16 and 6.17.

Several observations can be drawn from these results. First, GLASS outperforms SimpleNet across all checkpoint-selection strategies at both image and pixel levels, suggesting greater robustness under this crack-detection setting. Second, the pixel-level AUROC values are high for both models, suggesting that anomaly responses align relatively well with crack regions. The lower image-level AUROC indicates that aggregating pixel-level responses into image-level decisions remains less separable in this dataset.

For GLASS, the fixed checkpoint often produces diffuse or fragmented responses along the crack, with reduced contrast between defective and surrounding regions. In contrast, early stopping preserves more continuous and spatially coherent anomaly responses that better align with the

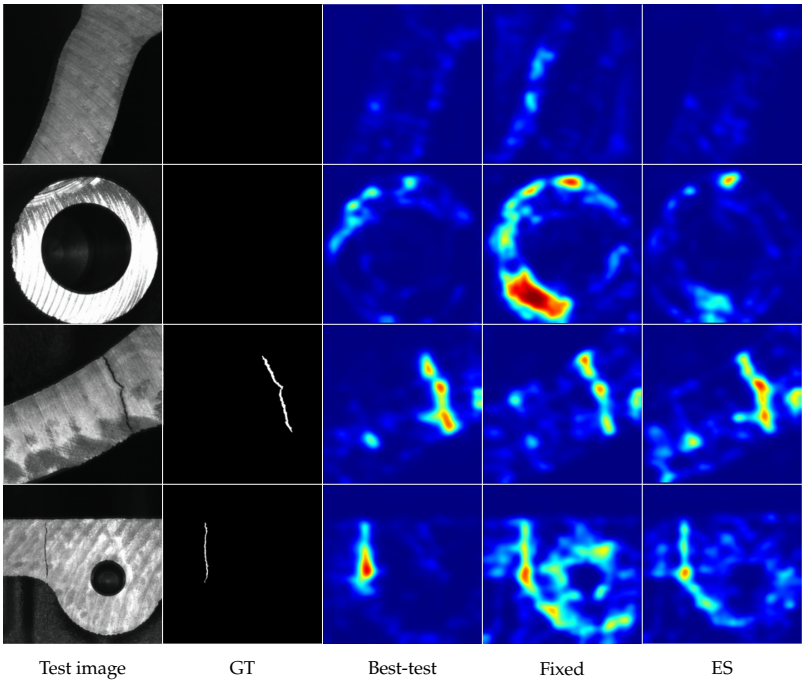


Figure 6.16 Qualitative anomaly detection results on the crack dataset using GLASS. From left to right: test image, ground truth mask, heatmaps from the best-test, fixed, and early stopping checkpoints.

ground truth crack geometry. This qualitative behavior is consistent with the higher pixel-level AUROC observed for early stopping in Table 6.18.

A similar trend is observed for SimpleNet, although overall localization quality remains lower than for GLASS. Fixed checkpoints tend to either suppress weak crack segments or produce scattered activations, while early stopping yields more concentrated responses along the crack path. These differences are consistent with the substantial pixel-level gains achieved by early stopping relative to fixed-length training.

Overall, the crack anomaly detection results demonstrate that checkpoint selection is a critical factor even in purely real-data settings. In the absence of a validation set, fixed-length training can lead to overly

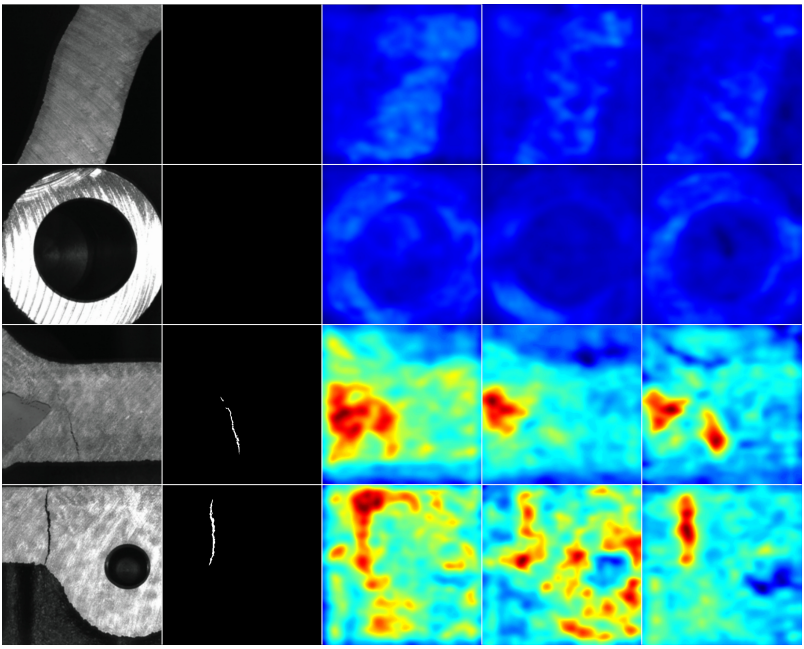


Figure 6.17 Qualitative anomaly detection results on the crack dataset using SimpleNet. From left to right: test image, ground truth mask, heatmaps from the best-test, fixed, and early stopping checkpoints.

specialized representations of the normal training distribution, reducing sensitivity to subtle defect structures. Stability-guided early stopping mitigates this effect by selecting checkpoints with more stable feature representations, thereby improving both image-level detection and pixel-level localization performance.

6.4.4 Conclusion

This chapter examined how synthetic data, refinement quality, and stability-guided checkpoint selection affect downstream performance across three industrial visual inspection tasks: classification, crack

segmentation, and anomaly detection. Rather than evaluating synthetic data in isolation, the chapter considered these factors jointly in order to assess under which conditions synthetic training data improve generalization and when their benefit remains limited by domain mismatch or training instability.

For granule classification, the results showed that the usefulness of synthetic data depends strongly on both refinement quality and data composition. Raw synthetic images consistently reduced performance, indicating that unrefined synthetic data do not provide a sufficient match to the appearance characteristics of the real granule images. In contrast, refined synthetic data were beneficial in several settings, particularly for the reflected-light dataset, where moderate amounts of refined synthetic samples improved classification accuracy over the real-only baseline. At the same time, the mixture experiments showed that this benefit does not extend uniformly to synthetic substitution: replacing substantial portions of real training data by synthetic samples led to performance degradation, especially for the transmitted-light dataset. These findings indicate that, for classification, refined synthetic data are most useful as a supplement to real data rather than as a direct replacement.

For crack segmentation, the experiments highlighted a stronger sensitivity to refinement-induced artifacts and to checkpoint selection under synthetic-to-real domain shift. Validation-based model selection within the synthetic domain did not always translate to good real-domain performance, particularly for refinement strategies that introduced unrealistic crack boundaries or texture artifacts. In this setting, stability-guided early stopping provided a useful alternative to synthetic-domain validation and, on average, yielded the strongest checkpoint performance across the compared refinement strategies. The segmentation results therefore suggest that, for dense prediction tasks with strong domain mismatch, both the quality of synthetic refinement and the choice of checkpoint-selection strategy have a substantial influence on real-domain generalization.

For anomaly detection, the results showed that the effect of synthetic normal data is highly dependent on the imaging setup, the refinement strategy, and the model architecture. On the reflected-light granule dataset, refined synthetic data could improve or maintain deployable performance in several settings, although the gains remained moderate

and full replacement of real data was generally not advantageous. On the transmitted-light dataset, the results were more sensitive to training dynamics: fixed checkpoints often performed poorly, whereas stability-guided early stopping substantially improved deployable performance and reduced the gap to oracle checkpoint selection. On the crack dataset, where only real training data were used, the experiments further showed that early stopping remained beneficial even without synthetic augmentation, indicating that stability-guided checkpoint selection addresses a broader training-control problem rather than a synthetic-data-specific one.

Taken together, the results of this chapter show that the effect of synthetic data cannot be assessed independently of refinement quality and checkpoint-selection strategy. Across the considered downstream tasks, refined synthetic data were most useful when they reduced the appearance gap to the real domain without introducing task-relevant artifacts, and when training was controlled in a way that prevented overfitting to synthetic-domain patterns or unstable late-stage solutions. Conversely, when the refined data remained poorly aligned with the real target domain, or when checkpoint selection relied on synthetic-domain validation signals that were not representative of the real test domain or on fixed training schedules, the potential benefit of synthetic data was limited or reversed.

Overall, this chapter suggests that synthetic data, refinement, and training control should be treated as interdependent components of a downstream inspection pipeline. Their usefulness is task dependent: classification benefited mainly from carefully refined augmentation, segmentation depended strongly on artifact-free refinement and robust checkpoint selection, and anomaly detection required particularly careful training control in the absence of labeled validation data. These findings provide the empirical basis for the final discussion of the dissertation, where the methodological contributions and their limitations are summarized in a broader context.

7 Summary and Outlook

7.1 Summary of Contributions

Industrial visual inspection systems based on deep learning face a set of tightly coupled challenges. In many industrial scenarios, representative defect data are scarce or costly to acquire, while inspection models must nevertheless generalize reliably across varying acquisition conditions and defect appearances. At the same time, training dynamics are often unstable under domain shift, and common model selection strategies rely on oracle assumptions that are not available in deployment. This dissertation addresses these challenges through a unified methodology that combines synthetic data generation, synthetic-to-real refinement, and stability-guided training control, with the aim of improving deployable inspection performance under realistic constraints.

At the foundation of this methodology lies the systematic use of synthetic data as a controllable source of supervision for industrial visual inspection tasks. Synthetic datasets are generated for multiple inspection settings, including granule sorting and crack detection, using physically motivated rendering as well as prompt-guided generation with large language vision models. These synthetic datasets provide a controlled and fully annotated basis for training and analyzing inspection models across different tasks, thereby enabling systematic investigation of data composition, refinement strategies, and training dynamics without requiring large-scale real defect datasets.

To further reduce the domain gap between synthetic and real imagery, synthetic-to-real refinement strategies are investigated. Feature-consistent neural style transfer and task-specific refinement variants are employed to align appearance statistics while preserving inspection-relevant content. The analysis reveals that the effectiveness of refinement depends not only on fidelity to real images, but also on the resulting data

variety. In the investigated settings, a level of variety that approximates the real data distribution is beneficial for downstream learning, whereas insufficient variety leads to limited coverage and excessive variety may extend the synthetic domain beyond the target operating regime. Consequently, refinement can be understood as a calibration process that jointly balances fidelity, consistency, and diversity, rather than as a purely realism-driven enhancement.

While refined synthetic data can improve oracle performance, the dissertation shows that their practical usefulness depends critically on training stability and checkpoint selection. To address this issue, a stability-guided early stopping strategy is introduced, based on monitoring the temporal consistency of learned feature embeddings. This approach enables convergence detection without reliance on labeled validation anomalies or test-set feedback, thereby providing a deployable alternative to oracle checkpoint selection. Across the investigated settings, stability-guided early stopping reduced overfitting effects and mitigated misleading training dynamics introduced by refinement or synthetic domain bias.

The proposed pipeline is validated across a range of downstream inspection tasks and model architectures. In addition to anomaly detection, which motivates the early stopping design, the experiments in Chapter 6 demonstrate that stability-guided checkpoint selection also improves robustness for supervised tasks such as crack segmentation, particularly when synthetic validation performance becomes unreliable. Classification experiments further illustrate how the interaction between synthetic data composition, refinement, and training control influences generalization behavior. Together, these results show that the methodology can be transferred across different inspection formulations while remaining sensitive to task- and model-specific characteristics.

Overall, this dissertation establishes a coherent framework for industrial visual inspection in which synthetic data generation, refinement with calibrated variety, and stability-aware training are treated as interconnected components. By explicitly accounting for data characteristics and training dynamics under realistic constraints, the presented approach supports more robust and deployable inspection performance

in the investigated settings, while reducing dependence on real defect data and oracle model selection.

7.2 Limitations

Despite the scope and results of this dissertation, several limitations remain that constrain the validity range of the presented conclusions and motivate further research.

First, while the dissertation targets industrial visual inspection and evaluates representative formulations including classification, segmentation, and anomaly detection, the empirical studies are necessarily conducted on a finite set of inspection scenarios, acquisition conditions, and defect families. The investigated applications span granule inspection and crack inspection with specific imaging setups and domain shifts, and the reported conclusions are therefore best interpreted as evidence for these practically motivated settings. Extending the evaluation to additional product classes, sensor modalities, and broader classes of defect mechanisms is still required to fully characterize generalization across the diversity of industrial inspection problems.

Second, the synthetic data generation pipelines rely on explicit design assumptions about scene composition, defect morphology, and nuisance factors. Although these assumptions enable controllability and scalable annotation, they also imply that the generated data may not capture all variability present in real production lines, especially rare defects, process drift, or unanticipated appearance changes. As a consequence, synthetic data can introduce biases toward the modeled factors. Such bias cannot be fully avoided in practice and must therefore be managed through careful modeling choices.

Third, the trade-off between refinement fidelity and data variety remains an open limitation. The experimental results indicate that different refinement paradigms occupy distinct points along this trade-off. GAN-based refinement methods typically achieve higher appearance fidelity with respect to the real domain, but are more prone to producing refined datasets with limited variety, which is consistent with the use of a single unified generator that learns a dominant target distribution. In contrast, neural style transfer–based refinement operates on a per-image

optimization process, resulting in substantially higher variety across refined samples. However, due to the explicit decoupling of content and style at the feature level, NST-based refinement tends to yield results that are less globally coherent than GAN-based approaches in the investigated settings, leading to slightly reduced overall fidelity despite improved interpretability and control. The present work does not identify a refinement strategy that simultaneously achieves high fidelity and high variety across the investigated settings, and reconciling these competing objectives remains an unresolved challenge.

Finally, the stability-guided early stopping strategy improves deployable checkpoint selection across the investigated settings, but its practical use depends on hyperparameter choices, *e.g.*, the dynamic threshold parameter β. In the current workflow, these hyperparameters are tuned using grid search or expert experience, which limits automation and may affect reproducibility when transferring the method to new tasks or domains.

7.3 Future Work

The limitations identified in this dissertation suggest several directions for future research that can further strengthen the proposed methodology and extend its applicability.

First, future work may focus on more adaptive strategies for synthetic data generation. Rather than relying exclusively on predefined defect models and manually designed variation ranges, synthetic generation could be guided by limited real data through iterative or feedback-driven mechanisms. For instance, discrepancies observed during downstream training or evaluation could be used to steer the synthesis process toward underrepresented defect patterns or appearance regimes. Such adaptive generation strategies may help reduce synthetic bias and improve coverage of rare or evolving defect types while preserving the controllability of simulation-based approaches.

Second, the refinement stage offers substantial room for further development. Beyond selective application and strength control of existing refinement methods, future work may explore refinement paradigms based on flow matching or diffusion models. These models have al-

ready demonstrated strong capabilities for high-fidelity image generation with rich sample diversity, as evidenced by their successful use in synthetic crack generation within this dissertation. Extending such models to synthetic-to-real refinement remains largely unexplored in industrial inspection scenarios. Due to their stochastic sampling mechanisms and multi-step denoising processes, diffusion- or flow-based refinement methods may offer an improved balance between fidelity and variety that more closely matches real data distributions. Whether these models can simultaneously preserve inspection-relevant content and achieve practical refinement performance, however, must be determined through systematic evaluation.

Third, an important direction for future research lies in the automatic tuning of the stability-guided early stopping hyperparameters. In particular, data-driven or self-adaptive strategies for determining hyperparameters would reduce reliance on expert knowledge and manual grid search. Possible approaches include coupling hyperparameter adjustment to training dynamics, embedding statistics, or uncertainty estimates, as well as integrating stability-based criteria into broader hyperparameter optimization frameworks. Progress in this direction would be a key step toward fully automated, reproducible, and deployment-ready training pipelines for industrial visual inspection.

Overall, these future research directions aim to extend the proposed methodology toward greater adaptability and automation, while maintaining the core principles of controllable data generation and stability-aware training introduced in this dissertation.

Appendix

A Evaluation Metrics

A.1 Distribution-Level Image Similarity Metrics

In several experiments of this dissertation, the goal is to compare a set of synthetic images with a set of real images at the distribution level rather than in a pairwise manner. This setting arises when there is no strict one-to-one correspondence between the images in the two sets. In such cases, pairwise metrics such as the mean squared error (MSE) or the structural similarity index (SSIM) are not directly applicable, since they require aligned image pairs. Instead, distribution-level metrics are used to quantify how closely the synthetic image set matches the real image set in terms of overall statistical properties. In this work, the Fréchet Inception Distance (FID) and the maximum mean discrepancy (MMD) are used for this purpose. In a general sense, both FID and MMD quantify the similarity between two image sets at the distribution level. In the specific context of this dissertation, however, they are used to compare synthetic images against real images, where the real image set serves as the reference distribution. Under this setting, both metrics can also be interpreted as fidelity measures, since they indicate how faithfully the synthetic image set reproduces the statistical properties of the real image set.

Let

$$\mathcal{D}_{\text{real}} = \left\{ \mathbf{X}^i_{\text{real}} \right\}_{i=1}^{N_{\text{real}}} \quad \text{and} \quad \mathcal{D}_{\text{syn}} = \left\{ \mathbf{X}^j_{\text{syn}} \right\}_{j=1}^{N_{\text{syn}}} \tag{A.1}$$

denote the real and synthetic image sets, respectively. Rather than comparing images directly in pixel space, both metrics are computed on feature representations extracted by a fixed pretrained feature encoder. This is motivated by the observation that perceptually meaningful image characteristics are often represented more appropriately in a deep feature space than by raw pixel intensities.

A.1.1 Fréchet Inception Distance

The Fréchet Inception Distance measures the discrepancy between the two image sets by comparing the distributions of their deep features. In practice, the features are commonly extracted from a fixed pretrained network, and each feature distribution is approximated by a multivariate Gaussian. Let

$$\mathbf{e}^i_{\text{real}} \in \mathbb{R}^d \quad \text{and} \quad \mathbf{e}^j_{\text{syn}} \in \mathbb{R}^d \tag{A.2}$$

denote the feature vectors extracted from images $\mathbf{X}^i_{\text{real}}$ and $\mathbf{X}^j_{\text{syn}}$, respectively. The empirical means are defined as

$$\mu_{\text{real}} = \frac{1}{N_{\text{real}}} \sum_{i=1}^{N_{\text{real}}} \mathbf{e}^i_{\text{real}} \quad \text{and} \quad \mu_{\text{syn}} = \frac{1}{N_{\text{syn}}} \sum_{j=1}^{N_{\text{syn}}} \mathbf{e}^j_{\text{syn}}, \tag{A.3}$$

and the corresponding empirical covariance matrices are given by

$$\mathbf{\Sigma}_{\text{real}} = \frac{1}{N_{\text{real}} - 1} \sum_{i=1}^{N_{\text{real}}} \left(\mathbf{e}^i_{\text{real}} - \mu_{\text{real}}\right)\left(\mathbf{e}^i_{\text{real}} - \mu_{\text{real}}\right)^\top \tag{A.4}$$

and

$$\mathbf{\Sigma}_{\text{syn}} = \frac{1}{N_{\text{syn}} - 1} \sum_{j=1}^{N_{\text{syn}}} \left(\mathbf{e}^j_{\text{syn}} - \mu_{\text{syn}}\right)\left(\mathbf{e}^j_{\text{syn}} - \mu_{\text{syn}}\right)^\top. \tag{A.5}$$

Based on these statistics, the FID is defined as

$$\text{FID} = \left\|\mu_{\text{real}} - \mu_{\text{syn}}\right\|_2^2 + \text{Tr}\left(\mathbf{\Sigma}_{\text{real}} + \mathbf{\Sigma}_{\text{syn}} - 2\left(\mathbf{\Sigma}_{\text{real}}\mathbf{\Sigma}_{\text{syn}}\right)^{1/2}\right). \tag{A.6}$$

The first term measures the discrepancy between the mean feature vectors of the two image sets, whereas the second term captures differences in their covariance structure. A lower FID indicates that the synthetic image set is closer to the real image set in the chosen feature space. In the ideal case, the FID becomes zero if the two feature distributions are identical.

A.1.2 Maximum Mean Discrepancy

The maximum mean discrepancy is another metric for comparing two image sets without requiring pairwise correspondence. In contrast to the FID, which assumes a Gaussian approximation of the feature distributions, the MMD is a non-parametric measure defined in a reproducing kernel Hilbert space. It quantifies the discrepancy between two distributions by comparing their mean embeddings induced by a kernel function.

Using the same feature vectors $\mathbf{e}^i_{\text{real}}$ and $\mathbf{e}^j_{\text{syn}}$ as in Section A.1.1, the squared MMD is defined as

$$\begin{aligned}\text{MMD}^2\left(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{syn}}\right) = &\frac{1}{N^2_{\text{real}}} \sum_{i=1}^{N_{\text{real}}} \sum_{i'=1}^{N_{\text{real}}} k\left(\mathbf{e}^i_{\text{real}}, \mathbf{e}^{i'}_{\text{real}}\right) \\ &+ \frac{1}{N^2_{\text{syn}}} \sum_{j=1}^{N_{\text{syn}}} \sum_{j'=1}^{N_{\text{syn}}} k\left(\mathbf{e}^j_{\text{syn}}, \mathbf{e}^{j'}_{\text{syn}}\right) \\ &- \frac{2}{N_{\text{real}} N_{\text{syn}}} \sum_{i=1}^{N_{\text{real}}} \sum_{j=1}^{N_{\text{syn}}} k\left(\mathbf{e}^i_{\text{real}}, \mathbf{e}^j_{\text{syn}}\right). \end{aligned} \tag{A.7}$$

In this work, the kernel function $k(\cdot, \cdot)$ is chosen as a Gaussian radial basis function kernel,

$$k(\mathbf{e}^i, \mathbf{e}^j) = \exp\left(-\frac{\left\|\mathbf{e}^i - \mathbf{e}^j\right\|_2^2}{2\sigma^2}\right), \tag{A.8}$$

where $\sigma > 0$ denotes the kernel bandwidth.

The MMD is non-negative and becomes zero if the two underlying distributions are identical under a characteristic kernel. Accordingly, a smaller MMD indicates a higher similarity between the two image sets. Compared with the FID, the MMD does not impose a Gaussian assumption on the feature distributions and can therefore capture more general differences between them. However, its value depends on the choice of kernel and its associated hyperparameters, especially the bandwidth σ. For this reason, MMD values are typically interpreted comparatively within the same experimental setting rather than absolutely across different studies.

A.2 Pairwise Image Similarity Metrics

In contrast to the distribution-level metrics discussed in the previous section, the metrics described here require a one-to-one correspondence between the compared images. This setting is relevant when each image in one set is associated with a specific counterpart in the other set, for example when a generated, refined, or reconstructed image is compared directly with its ground-truth reference. In such cases, the similarity between two image sets can be evaluated by first computing a pairwise similarity or distortion measure for each corresponding image pair and then averaging the resulting values over the full set. In this dissertation, the mean squared error (MSE), the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS) are used for this purpose.

In a general sense, these metrics quantify the similarity between two corresponding images. In the context of this dissertation, however, they are used to compare a (refined) synthetic image against its real reference image. Under this setting, they can also be interpreted as fidelity measures, since they indicate how faithfully the compared image preserves the visual content of the corresponding real image.

Let $\mathbf{X}_{\text{real}} \in \mathbb{R}^{H \times W \times C}$ denote a real image as reference, and $\mathbf{X}_{\text{syn}} \in \mathbb{R}^{H \times W \times C}$ the corresponding synthetic image, where H, W, and C denote the image height, width, and number of channels, respectively. Furthermore, let $(x_{\text{real}})_{u,v,c}$ and $(x_{\text{syn}})_{u,v,c}$ denote the pixel values at spatial position (u, v) and channel c in $\mathbf{X}_{\text{real}}$ and $\mathbf{X}_{\text{syn}}$, respectively. The following metrics are first computed for each corresponding image pair $(\mathbf{X}_{\text{real}}, \mathbf{X}_{\text{syn}})$ and then averaged over all image pairs in the evaluated dataset.

A.2.1 Mean Squared Error

The MSE is one of the most fundamental distortion measures for pairwise image comparison. It quantifies the average squared difference between corresponding pixel values of two images and is defined as

$$\mathrm{MSE}(\mathbf{X}_{\mathrm{real}}, \mathbf{X}_{\mathrm{syn}}) = \frac{1}{HWC} \sum_{u=1}^{H} \sum_{v=1}^{W} \sum_{c=1}^{C} \left((x_{\mathrm{real}})_{u,v,c} - (x_{\mathrm{syn}})_{u,v,c} \right)^2 . \tag{A.9}$$

A lower MSE indicates that the compared image is closer to the reference image in pixel space. In the ideal case, the MSE becomes zero if the two images are identical. However, since the MSE operates directly on pixel intensities, it is sensitive to small local shifts or slight structural changes and does not necessarily reflect perceptual similarity well.

A.2.2 Peak Signal-to-Noise Ratio

The PSNR is closely related to the MSE and is commonly used to measure reconstruction quality. It expresses the ratio between the maximum possible signal intensity and the distortion introduced by the reconstruction. The PSNR is defined as

$$\mathrm{PSNR}(\mathbf{X}_{\mathrm{real}}, \mathbf{X}_{\mathrm{syn}}) = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}(\mathbf{X}_{\mathrm{real}}, \mathbf{X}_{\mathrm{syn}})} \right) , \tag{A.10}$$

where 255 denotes the maximum possible pixel value of the 8-bit images.

Since the PSNR is inversely related to the MSE on a logarithmic scale, a higher PSNR indicates a higher similarity between the compared image and the reference image. In the limit of zero MSE, the PSNR approaches infinity. Compared with the MSE, the PSNR is often more convenient for reporting reconstruction quality, but it remains a pixel-wise measure and shares the same limitations regarding perceptual relevance.

A.2.3 Structural Similarity Index Measure

The SSIM is proposed to better reflect perceived image quality by comparing local image structures rather than only pixel-wise errors. It decomposes image similarity into three components, namely luminance, contrast, and structural similarity.

Given two corresponding local patches $\mathbf{X}_{\text{real}}$ and $\mathbf{X}_{\text{syn}}$ of a real reference image and a synthetic image, respectively, the SSIM is defined as

$$\text{SSIM}(\mathbf{X}_{\text{real}}, \mathbf{X}_{\text{syn}}) = \frac{\left(2\mu_{\mathbf{X}_{\text{real}}}\mu_{\mathbf{X}_{\text{syn}}} + \epsilon_1\right)\left(2Cov(\mathbf{X}_{\text{real}}, \mathbf{X}_{\text{syn}}) + \epsilon_2\right)}{\left(\mu^2_{\mathbf{X}_{\text{real}}} + \mu^2_{\mathbf{X}_{\text{syn}}} + \epsilon_1\right)\left(\sigma^2_{\mathbf{X}_{\text{real}}} + \sigma^2_{\mathbf{X}_{\text{syn}}} + \epsilon_2\right)}, \tag{A.11}$$

where $\mu_{\mathbf{X}_{\text{real}}}$ and $\mu_{\mathbf{X}_{\text{syn}}}$ denote the mean intensities of the corresponding image patches, $\sigma^2_{\mathbf{X}_{\text{real}}}$ and $\sigma^2_{\mathbf{X}_{\text{syn}}}$ their variances, and $Cov(\mathbf{X}_{\text{real}}, \mathbf{X}_{\text{syn}})$ their covariance. The constants ϵ_1 and ϵ_2 are small positive values introduced to stabilize the division when the denominators are close to zero. Specifically, ϵ_1 stabilizes the luminance comparison term, which depends on the local means, whereas ϵ_2 stabilizes the contrast-structure term, which depends on the local variances and covariance. This separate treatment is necessary because the two components can become numerically unstable under different local imaging conditions, for example in nearly uniform regions with very small mean or variance values.

In practice, the SSIM is computed locally over sliding windows across the image and then averaged to obtain a single score for the full image pair. The resulting value typically lies in the range from 0 to 1, where a higher value indicates a higher structural similarity. Compared with MSE and PSNR, the SSIM is generally more consistent with human perception, since it emphasizes the preservation of structural image content.

A.2.4 Learned Perceptual Image Patch Similarity

The learned perceptual image patch similarity is a feature-based metric designed to better align with perceptual judgments of image similarity. Instead of comparing images directly in pixel space, LPIPS measures the distance between deep feature representations extracted from a pretrained neural network.

Let $F(\mathbf{X}_{\text{real}})^{(l)}$ and $F(\mathbf{X}_{\text{syn}})^{(l)}$ denote the feature maps of the real reference image and the synthetic image at layer l of a fixed pretrained network. The LPIPS is computed by comparing the normalized feature

activations across multiple layers and aggregating their weighted distances. A general form can be written as

$$\begin{aligned}&\mathrm{LPIPS}(\mathbf{X}_{\mathrm{real}}, \mathbf{X}_{\mathrm{syn}}) \\ &= \sum_{l} \frac{1}{H^{(l)} W^{(l)}} \sum_{u=1}^{H^{(l)}} \sum_{v=1}^{W^{(l)}} \left\| \mathbf{w}^{(l)} \odot \left(\frac{F(\mathbf{X}_{\mathrm{real}})_{u,v}^{(l)}}{\left\| F(\mathbf{X}_{\mathrm{real}})_{u,v}^{(l)} \right\|_2} - \frac{F(\mathbf{X}_{\mathrm{syn}})_{u,v}^{(l)}}{\left\| F(\mathbf{X}_{\mathrm{syn}})_{u,v}^{(l)} \right\|_2} \right) \right\|_2^2 .\end{aligned} \tag{A.12}$$

where $H^{(l)}$ and $W^{(l)}$ denote the spatial dimensions of the corresponding feature maps at layer l, $\mathbf{w}^{(l)}$ is a learned channel-wise weighting vector, and $\odot$ denotes element-wise multiplication.

A lower LPIPS indicates that the compared image is more similar to the reference image in the learned perceptual feature space. Since LPIPS operates on deep features rather than directly on pixels, it is often more sensitive to perceptually relevant differences in texture, structure, and appearance. It is therefore particularly useful when the visual plausibility of synthetic images is of interest.

A.3 Metrics for Binary Classification

For binary classification, the objective is to assign each input sample to one of two classes. In this dissertation, these two classes correspond to the negative and positive class, respectively. The performance of a binary classifier is evaluated by comparing its predicted labels with the corresponding ground-truth labels. In this context, different metrics emphasize different aspects of the classification behavior. In this dissertation, the classification performance is evaluated by the accuracy (Acc), precision, recall, and F1 score.

Let

$$\mathcal{D}_{\mathrm{test}} = \left\{ \left(\mathbf{X}^i, y^i \right) \right\}_{i=1}^{N} \tag{A.13}$$

denote a test set with N samples, where $\mathbf{X}^i$ is the i-th input sample and $y^i \in \{0, 1\}$ is its ground-truth class label. Here, $y^i = 0$ denotes the negative class and $y^i = 1$ the positive class. Let $\hat{y}^i \in \{0, 1\}$ denote the predicted class label for sample $\mathbf{X}^i$.

By comparing a predicted binary label with its corresponding ground-truth label, each decision can be assigned to one of four cases. A true positive (TP) is obtained when a positive instance is correctly predicted as positive. A true negative (TN) is obtained when a negative instance is correctly predicted as negative. A false positive (FP) occurs when a negative instance is incorrectly predicted as positive. A false negative (FN) occurs when a positive instance is incorrectly predicted as negative. These four quantities can be defined at different levels depending on the task, for example at the image level or pixel level. They form the basis of the binary classification metrics described below and are also used in the following sections for binary segmentation and anomaly detection.

A.3.1 Accuracy

The accuracy measures the proportion of correctly classified samples among all samples in the test set. It is defined as

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} . \tag{A.14}$$

A higher accuracy indicates that a larger fraction of the samples is classified correctly. While accuracy provides an overall measure of classification performance, it can be less informative when the class distribution is imbalanced, since a classifier may achieve a high accuracy by favoring the majority class.

A.3.2 Precision

The precision measures the proportion of correctly predicted positive samples among all samples predicted as positive. It is defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} . \tag{A.15}$$

A higher precision indicates that positive predictions are more reliable because fewer negative samples are incorrectly classified as positive. Precision is therefore particularly relevant when false positives are undesirable.

A.3.3 Recall

The recall, also referred to as sensitivity or true positive rate, measures the proportion of correctly identified positive samples among all actual positive samples. It is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{A.16}$$

A higher recall indicates that fewer positive samples are missed by the classifier. This metric is therefore particularly relevant when false negatives are undesirable.

A.3.4 F1 Score

The F1 score combines precision and recall into a single metric by computing their harmonic mean. It is defined as

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{A.17}$$

By substituting the definitions of precision and recall, the F1 score can equivalently be written as

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{A.18}$$

A higher F1 score indicates a better balance between precision and recall. It is particularly useful when both false positives and false negatives should be taken into account simultaneously.

A.4 Metrics for Binary Segmentation

In binary segmentation, the objective is to assign each pixel of an input image to one of two classes, namely the foreground class and the background class. The performance of a binary segmentation model is therefore evaluated by comparing the predicted binary mask with the corresponding ground-truth mask at the pixel level. In this dissertation, the segmentation performance is evaluated by the foreground intersection over union (foreground IoU), background intersection over union (background IoU), and mean intersection over union (mean IoU).

Let

$$\mathcal{D}_{\text{test}} = \left\{\left(\mathbf{X}^i, \mathbf{M}^i\right)\right\}_{i=1}^{N} \tag{A.19}$$

denote a test set with N images, where $\mathbf{X}^i$ is the i-th input image and $\mathbf{M}^i \in \{0,1\}^{H\times W}$ is the corresponding binary ground-truth mask. Here, a value of 1 indicates the foreground class and a value of 0 the background class. Let $\hat{\mathbf{M}}^i \in \{0,1\}^{H\times W}$ denote the predicted binary mask for image $\mathbf{X}^i$.

In the context of binary segmentation, the quantities TP, TN, FP, and FN are computed at the pixel level by comparing the predicted mask $\hat{\mathbf{M}}^i$ with the ground-truth mask $\mathbf{M}^i$. Based on these quantities, the segmentation metrics described below are defined.

A.4.1 Foreground Intersection over Union

The foreground intersection over union measures the overlap between the predicted foreground region and the ground-truth foreground region. It is defined as

$$\text{IoU}_{\text{fg}} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{A.20}$$

Equivalently, using set notation, the foreground IoU can be written as

$$\text{IoU}_{\text{fg}} = \frac{\left|\hat{\mathbf{M}} \cap \mathbf{M}\right|}{\left|\hat{\mathbf{M}} \cup \mathbf{M}\right|}, \tag{A.21}$$

where the intersection and union are understood pixel-wise for the foreground class. A higher foreground IoU indicates a better overlap between the predicted and ground-truth foreground regions.

A.4.2 Background Intersection over Union

Analogously, the background intersection over union measures the overlap between the predicted background region and the ground-truth background region. It is defined as

$$\text{IoU}_{\text{bg}} = \frac{\text{TN}}{\text{TN} + \text{FP} + \text{FN}}. \tag{A.22}$$

This metric measures the overlap between the predicted and ground-truth background regions. A higher background IoU indicates a more accurate segmentation of the background class.

A.4.3 Mean Intersection over Union

Since binary segmentation involves both foreground and background classes, it is often useful to summarize the segmentation performance across both classes. In this dissertation, this is done by the mean intersection over union, which is defined as the arithmetic mean of the foreground and background IoU:

$$\mathrm{IoU}_{\mathrm{mean}} = \frac{\mathrm{IoU}_{\mathrm{fg}} + \mathrm{IoU}_{\mathrm{bg}}}{2}. \tag{A.23}$$

The mean IoU provides a balanced summary of the segmentation quality over both classes. It is particularly useful when both foreground and background accuracy are of interest, since a model may achieve a high score for one class while performing worse on the other.

A.5 Metrics for Anomaly Detection

In anomaly detection, the objective is to distinguish normal samples from anomalous ones without restricting the evaluation to a fixed decision threshold. For this reason, threshold-independent ranking metrics are commonly used. In this dissertation, the performance of anomaly detection methods is evaluated by the image-level area under the receiver operating characteristic curve (Image AUROC) and the pixel-level area under the receiver operating characteristic curve (Pixel AUROC). These two metrics assess anomaly detection performance at different granularities: the former evaluates whether an image is correctly ranked as normal or anomalous, whereas the latter evaluates whether anomalous pixels are correctly ranked higher than normal pixels.

Let $\mathcal{D}_{\mathrm{test}} = \{(\mathbf{X}^i, y^i, \mathbf{M}^i)\}_{i=1}^N$ denote a test set with N images, where $\mathbf{X}^i$ is the i-th test image, $y^i \in \{0, 1\}$ is its image-level label, and $\mathbf{M}^i \in \{0, 1\}^{H \times W}$ is the corresponding binary anomaly mask. Here, $y^i = 0$ indicates a normal image and $y^i = 1$ an anomalous image. For normal images, $\mathbf{M}^i$ is an all-zero mask. Let $s(\mathbf{X}^i) \in \mathbb{R}$ denote the anomaly score predicted for image $\mathbf{X}^i$, where a higher value indicates a higher likelihood of anomaly. Furthermore, let $\mathbf{S}(\mathbf{X}^i) \in \mathbb{R}^{H \times W}$ denote the predicted anomaly score map of image $\mathbf{X}^i$, where a larger score at a pixel indicates a higher anomaly likelihood at that location.

A.5.1 Receiver Operating Characteristic Curve

The receiver operating characteristic (ROC) curve describes the trade-off between the true positive rate and the false positive rate under varying decision thresholds. Given a threshold τ, a sample is classified as anomalous if its anomaly score is greater than or equal to τ, and as normal otherwise. Based on this threshold, the true positive rate and false positive rate are defined as

$$\mathrm{TPR}(\tau) = \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau) + \mathrm{FN}(\tau)} \tag{A.24}$$

and

$$\mathrm{FPR}(\tau) = \frac{\mathrm{FP}(\tau)}{\mathrm{FP}(\tau) + \mathrm{TN}(\tau)}, \tag{A.25}$$

where $\mathrm{TP}(\tau)$, $\mathrm{FP}(\tau)$, $\mathrm{TN}(\tau)$, and $\mathrm{FN}(\tau)$ denote the numbers of true positives, false positives, true negatives, and false negatives at threshold τ, respectively. Depending on the evaluation level, these quantities are computed either over images or over pixels, corresponding to image-level anomaly detection or pixel-level anomaly localization.

The ROC curve is obtained by plotting $\mathrm{TPR}(\tau)$ against $\mathrm{FPR}(\tau)$ while varying τ over all possible score values. The corresponding area under this curve is referred to as the AUROC. Since it integrates over all thresholds, the AUROC evaluates the ranking quality of the anomaly scores rather than the performance at one specific operating point. Its value lies in the range $[0, 1]$, where a value of 1 indicates perfect separation between normal and anomalous samples, and a value of 0.5 corresponds to random ranking.

A.5.2 Image AUROC

The Image AUROC evaluates anomaly detection performance at the image level. It measures how well the predicted image-level anomaly scores $\{s(\mathbf{X}^i)\}_{i=1}^{N}$ separate normal images from anomalous ones. In other words, it quantifies whether anomalous images tend to receive higher anomaly scores than normal images across the test set.

For a given threshold τ, the predicted image-level label is defined as

$$\hat{y}^i(\tau) = \begin{cases} 1, & s(\mathbf{X}^i) \geq \tau, \\ 0, & s(\mathbf{X}^i) < \tau. \end{cases} \tag{A.26}$$

Based on these predicted labels and the ground-truth image labels y^i, the ROC curve is constructed, and the corresponding area under the curve gives the Image AUROC.

A higher Image AUROC indicates that the anomaly detection method is more effective at ranking anomalous images above normal images. Since it is threshold-independent, this metric is particularly suitable when the anomaly scores of different methods are to be compared without choosing a fixed classification threshold.

A.5.3 Pixel AUROC

The Pixel AUROC evaluates anomaly localization performance at the pixel level. It measures how well the predicted anomaly score maps $\{\mathbf{S}(\mathbf{X}^i)\}_{i=1}^N$ separate anomalous pixels from normal pixels over the full test set. Compared with the Image AUROC, this metric provides a more fine-grained assessment, since it evaluates not only whether an image is anomalous, but also whether the anomalous regions are assigned high scores at the correct spatial locations.

For a given threshold τ, the predicted binary anomaly mask is obtained by thresholding the anomaly score map:

$$\hat{\mathbf{M}}^i(\tau)_{u,v} = \begin{cases} 1, & (\mathbf{S}(\mathbf{X}^i))_{u,v} \geq \tau, \\ 0, & (\mathbf{S}(\mathbf{X}^i))_{u,v} < \tau. \end{cases} \tag{A.27}$$

Using all pixels from all test images, the true positive rate and false positive rate are computed with respect to the ground-truth masks $\{\mathbf{M}^i\}_{i=1}^N$, and the resulting ROC curve yields the Pixel AUROC.

A higher Pixel AUROC indicates that anomalous pixels are more consistently assigned higher scores than normal pixels. This metric is therefore particularly relevant when the spatial localization quality of anomaly detection is of interest.

Bibliography

[1] **Abu Alhaija, H., S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother**. *Augmented reality meets computer vision: Efficient data generation for urban driving scenes*. In: *International Journal of Computer Vision* 126.9 (2018), pp. 961–972.

[2] **Agapaki, E. and I. Brilakis**. *Instance segmentation of industrial point cloud data*. In: *Journal of Computing in Civil Engineering* 35.6 (2021), p. 04021022.

[3] **Ai, D., G. Jiang, S.-K. Lam, P. He, and C. Li**. *Computer vision framework for crack detection of civil infrastructure—A review*. In: *Engineering Applications of Artificial Intelligence* 117 (2023), p. 105478.

[4] **Akcay, S., A. Atapour-Abarghouei, and T. P. Breckon**. *Ganomaly: Semi-supervised anomaly detection via adversarial training*. In: *Asian conference on computer vision*. Springer. 2018, pp. 622–637.

[5] **Alzubaidi, L., J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-Dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, *et al.*** *A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications*. In: *Journal of Big Data* 10.1 (2023), p. 46.

[6] **Andrews, J., T. Tanay, E. J. Morton, and L. D. Griffin**. *Transfer representation-learning for anomaly detection*. In: JMLR. 2016.

[7] **Antwi-Bekoe, E., G. Liu, J.-P. Ainam, G. Sun, and X. Xie**. *A deep learning approach for insulator instance segmentation and defect detection*. In: *Neural Computing and Applications* 34.9 (2022), pp. 7253–7269.

[8] **Arjovsky, M., S. Chintala, and L. Bottou**. *Wasserstein generative adversarial networks*. In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[9] **Aydın, E. Ö., E. Tepe, and C. Balcan**. *Identification of determinants during the registration process of industrial heritage using a regression analysis*. In: *Journal of Cultural Heritage* 58 (2022), pp. 23–32.

[10] **Bacon, D. J. and Y. N. Osetsky**. *Multiscale modelling of radiation damage in metals: from defect generation to material properties*. In: *Materials Science and Engineering: A* 365.1-2 (2004), pp. 46–56.

[11] **Bai, J., S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, *et al.*** *Qwen technical report*. In: *arXiv preprint arXiv:2309.16609* (2023).

[12] **Bao, Y., K. Song, J. Liu, Y. Wang, Y. Yan, H. Yu, and X. Li**. *Triplet-graph reasoning network for few-shot metal generic surface defect segmentation*. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11.

[13] **Basseville, M., I. V. Nikiforov, *et al.*** *Detection of abrupt changes: theory and application*. Vol. 104. Prentice hall Englewood Cliffs, 1993.

[14] **Baumes, L., J. Serra, P. Serna, and A. Corma**. *Support vector machines for predictive modeling in heterogeneous catalysis: a comprehensive introduction and overfitting investigation based on two real applications*. In: *Journal of combinatorial chemistry* 8.4 (2006), pp. 583–596.

[15] **Bengio, Y.** *Practical recommendations for gradient-based training of deep architectures*. In: *Neural networks: Tricks of the trade: Second edition*. Springer, 2012, pp. 437–478.

[16] **Bengio, Y., P. Simard, and P. Frasconi**. *Learning long-term dependencies with gradient descent is difficult*. In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[17] **Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle**. *Greedy layer-wise training of deep networks*. In: *Advances in neural information processing systems* 19 (2006).

[18] **Bergmann, P., M. Fauser, D. Sattlegger, and C. 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.

[19] **Bishop, C. M.** *Training with noise is equivalent to Tikhonov regularization*. In: *Neural computation* 7.1 (1995), pp. 108–116.

[20] **Bishop, C. M. and N. M. Nasrabadi**. *Pattern recognition and machine learning*. Vol. 4. Springer, 2006.

[21] **Bjorck, N., C. P. Gomes, B. Selman, and K. Q. Weinberger**. *Understanding batch normalization*. In: *Advances in neural information processing systems* 31 (2018).

[22] **Bogdoll, D., M. Zhang, M. Nitsche, and J. M. Zöllner**. *Experiments on anomaly detection in autonomous driving by forward-backward style transfers*. In: *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE. 2022, pp. 1–7.

[23] **Boikov, A., V. Payor, R. Savelev, and A. Kolesnikov**. *Synthetic data generation for steel defect detection and classification using deep learning*. In: *Symmetry* 13.7 (2021), p. 1176.

[24] **Boroujeni, S. R., H. Abedi, and T. Bush**. *Enhancing Glass Defect Detection with Diffusion Models: Addressing Imbalanced Datasets in Manufacturing Quality Control*. In: *arXiv preprint* (2025). arXiv: 2505.03134.

[25] **Bottou, L.** *Large-scale machine learning with stochastic gradient descent*. In: *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer. 2010, pp. 177–186.

[26] **Bottou, L., F. E. Curtis, and J. Nocedal**. *Optimization methods for large-scale machine learning*. In: *SIAM review* 60.2 (2018), pp. 223–311.

[27] **Bousmalis, K., N. Silberman, D. Dohan, D. Erhan, and D. Krishnan**. *Unsupervised pixel-level domain adaptation with generative adversarial networks*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3722–3731.

[28] **Bousmalis, K., A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*** *Using simulation and domain adaptation to improve efficiency of deep robotic grasping*. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 4243–4250.

[29] **Brock, A., J. Donahue, and K. Simonyan**. *Large scale GAN training for high fidelity natural image synthesis*. In: *arXiv preprint arXiv:1809.11096* (2018).

[30] **Burnaev, E., P. Erofeev, and D. Smolyakov**. *Model selection for anomaly detection*. In: *Eighth International Conference on Machine Vision (ICMV 2015)*. Vol. 9875. SPIE. 2015, pp. 445–450.

[31] **Cao, J., G. Yang, and X. Yang**. *A pixel-level segmentation convolutional neural network based on deep feature fusion for surface defect detection*. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2020), pp. 1–12.

[32] **Chakrabarti, A., K. Shea, R. Stone, J. Cagan, M. Campbell, N. V. Hernandez, and K. L. Wood**. *Computer-based design synthesis research: an overview*. In: *Journal of Computing and Information Science in Engineering* 11 (2011), p. 021003.

[33] **Chalapathy, R. and S. Chawla**. *Deep learning for anomaly detection: A survey*. In: *arXiv preprint arXiv:1901.03407* (2019).

[34] **Chand, K., T. Fritsch, S. Oster, A. Ulbricht, and G. Bruno**. *Review on image registration methods for the quality control in additive manufacturing*. In: *Progress in Additive Manufacturing* (2025), pp. 1–27.

[35] **Chandola, V., A. Banerjee, and V. Kumar**. *Anomaly detection: A survey*. In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[36] **Chen, C. and X. Jiang**. *Cluttered Reflective Metal Parts Instance Segmentation*. In: *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2022, pp. 1146–1150.

[37] **Chen, C.-F. R., Q. Fan, and R. Panda**. *Crossvit: Cross-attention multi-scale vision transformer for image classification*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 357–366.

[38] **Chen, H., Z. Wang, H. Zhang, Z. Zuo, A. Li, W. Xing, D. Lu, *et al.*** *Artistic style transfer with internal-external learning and contrastive learning*. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26561–26573.

[39] **Chen, H., Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao**. *Pre-trained image processing transformer*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12299–12310.

[40] **Chen, H., T. Kiyokawa, W. Wan, and K. Harada**. *Category-association based similarity matching for novel object pick-and-place task*. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2961–2968.

[41] **Chen, J., Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou**. *Transunet: Transformers make strong encoders for medical image segmentation*. In: *arXiv preprint arXiv:2102.04306* (2021).

[42] **Chen, M., A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever**. *Generative pretraining from pixels*. In: *International conference on machine learning*. PMLR. 2020, pp. 1691–1703.

[43] **Chen, Q. and V. Koltun**. *Photographic image synthesis with cascaded refinement networks*. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1511–1520.

[44] **Chen, Q., H. Luo, C. Lv, and Z. Zhang**. *A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization*. In: *European Conference on Computer Vision*. Springer. 2024, pp. 37–54.

[45] **Chen, R. T., Y. Rubanova, J. Bettencourt, and D. K. Duvenaud**. *Neural ordinary differential equations*. In: *Advances in neural information processing systems* 31 (2018).

[46] **Chen, T., S. Kornblith, M. Norouzi, and G. Hinton**. *A simple framework for contrastive learning of visual representations*. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[47] **Chen, Y., Y. Ding, F. Zhao, E. Zhang, Z. Wu, and L. Shao**. *Surface defect detection methods for industrial products: A review*. In: *Applied Sciences* 11.16 (2021), p. 7657.

[48] **Chen, Y., X. Li, M. Jia, J. Li, T. Hu, and J. Luo**. *Instance segmentation and number counting of grape berry images based on deep learning*. In: *Applied Sciences* 13.11 (2023), p. 6751.

[49] **Chin, R. T. and C. A. Harlow**. *Automated visual inspection: A survey*. In: *IEEE transactions on pattern analysis and machine intelligence* 6 (1982), pp. 557–573.

[50] **Cho, Y., G. Kim, S. Safarov, S. Bang, and J. Park**. *Background-Aware Defect Generation for Robust Industrial Anomaly Detection*. In: *arXiv preprint*. 2024. arXiv: 2411.16767.

[51] **Cholakkal, H., G. Sun, F. S. Khan, and L. Shao**. *Object counting and instance segmentation with image-level supervision*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12397–12405.

[52] **Chung, H., J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye**. *Diffusion posterior sampling for general noisy inverse problems*. In: *arXiv preprint arXiv:2209.14687* (2022).

[53] **Chung, J., S. Hyun, and J.-P. Heo**. *Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 8795–8805.

[54] **Community, B. O.** *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018.

[55] **Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele**. *The cityscapes dataset for semantic urban scene understanding*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[56] **Cubuk, E. D., B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le**. *Autoaugment: Learning augmentation strategies from data*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 113–123.

[57] **Cui, Y., Z. Liu, and S. Lian**. *A survey on unsupervised anomaly detection algorithms for industrial images*. In: *IEEE Access* 11 (2023), pp. 55297–55315.

[58] **Czimmermann, T., G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, and P. Dario**. *Visual-based defect detection and classification approaches for industrial applications—a survey*. In: *Sensors* 20.5 (2020), p. 1459.

[59] **d'Ascoli, S., H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun**. *Convit: Improving vision transformers with soft convolutional inductive biases*. In: *International conference on machine learning*. PMLR. 2021, pp. 2286–2296.

[60] **Dais, D., I. E. Bal, E. Smyrou, and V. Sarhosis**. *Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning*. In: *Automation in Construction* 125 (2021), p. 103606.

[61] **Dean, J., G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, *et al.*** *Large scale distributed deep networks*. In: *Advances in neural information processing systems* 25 (2012).

[62] **Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei**. *Imagenet: A large-scale hierarchical image database*. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[63] **Dhariwal, P. and A. Nichol**. *Diffusion models beat gans on image synthesis*. In: *Advances in neural information processing systems* 34 (2021), pp. 8780–8794.

[64] **Ding, X., Y. Zhao, and L. Akoglu**. *Fast unsupervised deep outlier model selection with hypernetworks*. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024, pp. 585–596.

[65] **Dobrzycki, A. D., A. M. Bernardos, and J. R. Casar**. *An Analysis of Layer-Freezing Strategies for Enhanced Transfer Learning in YOLO Architectures*. In: *Mathematics* 13.15 (2025), p. 2539.

[66] **Dosovitskiy, A.** *An image is worth 16x16 words: Transformers for image recognition at scale*. In: *arXiv preprint arXiv:2010.11929* (2020).

[67] **Duchi, J., E. Hazan, and Y. Singer**. *Adaptive subgradient methods for online learning and stochastic optimization.* In: *Journal of machine learning research* 12.7 (2011).

[68] **Dundar, A., M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz**. *Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation*. In: *arXiv preprint arXiv:1807.09384* (2018).

[69] **Esser, P., S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, *et al.*** *Scaling rectified flow transformers for high-resolution image synthesis*. In: *Forty-first international conference on machine learning*. 2024.

[70] **Eversberg, L. and J. Lambrecht**. *Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization*. In: *Sensors* 21.23 (2021), p. 7901.

[71] **Frid-Adar, M., I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan**. *GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification*. In: *Neurocomputing* 321 (2018), pp. 321–331.

[72] **Fukushima, K.** *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*. In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[73] **Fung, C., C. Qiu, A. Li, and M. Rudolph**. *Model selection of anomaly detectors in the absence of labeled validation data*. In: *IEEE Transactions on Artificial Intelligence* (2025).

[74] **Gao, Y., M. Zhou, and D. N. Metaxas**. *UTNet: a hybrid transformer architecture for medical image segmentation*. In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2021, pp. 61–71.

[75] **Gatys, L. A., A. S. Ecker, and M. Bethge**. *A neural algorithm of artistic style*. In: *arXiv preprint arXiv:1508.06576* (2015).

[76] **Gatys, L. A., A. S. Ecker, and M. Bethge**. *Image style transfer using convolutional neural networks*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[77] **Gatys, L. A., M. Bethge, A. Hertzmann, and E. Shechtman**. *Preserving color in neural artistic style transfer*. In: *arXiv preprint arXiv:1606.05897* (2016).

[78] **Ge, S., T. Park, J.-Y. Zhu, and J.-B. Huang**. *Expressive text-to-image generation with rich text*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7545–7556.

[79] **Geng, Z., M. Deng, X. Bai, J. Z. Kolter, and K. He**. *Mean flows for one-step generative modeling*. In: *arXiv preprint arXiv:2505.13447* (2025).

[80] **Glorot, X. and Y. Bengio**. *Understanding the difficulty of training deep feedforward neural networks*. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[81] **Goodfellow, I.** *Deep learning*. MIT press, 2016.

[82] **Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio**. *Generative adversarial networks*. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[83] **Goshtasby, A. A.** *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*. John Wiley & Sons, 2005.

[84] **Goshtasby, A. A.** *Theory and applications of image registration*. John Wiley & Sons, 2017.

[85] **Goswami, M., C. Challu, L. Callot, L. Minorics, and A. Kan**. *Unsupervised model selection for time-series anomaly detection*. In: *arXiv preprint arXiv:2210.01078* (2022).

[86] **Grard, M.** *Generic instance segmentation for object-oriented bin-picking*. PhD thesis. Université de Lyon, 2019.

[87] **Gretton, A., K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola**. *A kernel two-sample test*. In: *The journal of machine learning research* 13.1 (2012), pp. 723–773.

[88] **Grill, J.-B., F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*** *Bootstrap your own latent-a new approach to self-supervised learning*. In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284.

[89] **Gu, A. and T. Dao**. *Mamba: Linear-time sequence modeling with selective state spaces*. In: *First conference on language modeling*. 2024.

[90] **Gualtieri, M. and R. Platt**. *Robotic pick-and-place with uncertain object instance segmentation and shape completion*. In: *IEEE robotics and automation letters* 6.2 (2021), pp. 1753–1760.

[91] **Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville**. *Improved training of wasserstein gans*. In: *Advances in neural information processing systems* 30 (2017).

[92] **Guo, F., Y. Qian, Y. Wu, Z. Leng, and H. Yu**. *Automatic railroad track components inspection using real-time instance segmentation*. In: *Computer-Aided Civil and Infrastructure Engineering* 36.3 (2021), pp. 362–377.

[93] **Gupta, A., A. Vedaldi, and A. Zisserman**. *Synthetic data for text localisation in natural images*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2315–2324.

[94] **Hagelskjær, F., A. Kramberger, A. Wolniakowski, T. R. Savarimuthu, and N. Krüger**. *Combined optimization of gripper finger design and pose estimation processes for advanced industrial assembly*. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 2022–2029.

[95] **Hardt, M., B. Recht, and Y. Singer**. *Train faster, generalize better: Stability of stochastic gradient descent*. In: *International conference on machine learning*. PMLR. 2016, pp. 1225–1234.

[96] **Hastie, T., R. Tibshirani, J. Friedman, *et al.*** *The elements of statistical learning*. Springer series in statistics New-York, 2009.

[97] **Hatamizadeh, A., Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu**. *Unetr: Transformers for 3d medical image segmentation*. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2022, pp. 574–584.

[98] **He, D., K. Xu, and P. Zhou**. *Defect detection of hot rolled steels with a new object detection framework called classification priority network*. In: *Computers & Industrial Engineering* 128 (2019), pp. 290–297.

[99] **He, K., X. Zhang, S. Ren, and J. Sun**. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[100] **He, K., X. Zhang, S. Ren, and J. Sun**. *Deep residual learning for image recognition*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[101] **He, X., Y. Chen, and Z. Lin**. *Spatial-spectral transformer for hyperspectral image classification*. In: *Remote Sensing* 13.3 (2021), p. 498.

[102] **He, Y., K. Song, Q. Meng, and Y. Yan**. *An end-to-end steel surface defect detection approach via fusing multiple hierarchical features*. In: *IEEE transactions on instrumentation and measurement* 69.4 (2019), pp. 1493–1504.

[103] **He, Y., K. Song, H. Dong, and Y. Yan**. *Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network*. In: *Optics and Lasers in Engineering* 122 (2019), pp. 294–302.

[104] **Heckler-Kram, L., J.-H. Neudeck, U. Scheler, R. König, and C. Steger**. *The mvtec ad 2 dataset: Advanced scenarios for unsupervised anomaly detection*. In: *arXiv preprint arXiv:2503.21622* (2025).

[105] **Heusel, M., H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter**. *Gans trained by a two time-scale update rule converge to a local nash equilibrium*. In: *Advances in neural information processing systems* 30 (2017).

[106] **Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov**. *Improving neural networks by preventing co-adaptation of feature detectors*. In: *arXiv preprint arXiv:1207.0580* (2012).

[107] **Ho, J., A. Jain, and P. Abbeel**. *Denoising diffusion probabilistic models*. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[108] **Ho, J. and T. Salimans**. *Classifier-free diffusion guidance*. In: *arXiv preprint arXiv:2207.12598* (2022).

[109] **Hochreiter, S.** *The vanishing gradient problem during learning recurrent neural nets and problem solutions*. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[110] **Hodge, V. and J. Austin**. *A survey of outlier detection methodologies*. In: *Artificial intelligence review* 22.2 (2004), pp. 85–126.

[111] **Hoogeboom, E., J. Heek, and T. Salimans**. *Simple diffusion: End-to-end diffusion for high resolution images*. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 13213–13232.

[112] **Hu, J., F. Xiao, Q. Jin, G. Zhao, and P. Lou**. *Synthetic data generation based on RDB-CycleGAN for industrial object detection*. In: *Mathematics* 11.22 (2023), p. 4588.

[113] **Huang, W. and P. Wei**. *A PCB dataset for defects detection and classification*. In: *arXiv preprint arXiv:1901.08204* (2019).

[114] **Huang, W., P. Wei, M. Zhang, and H. Liu**. *HRIPCB: a challenging dataset for PCB defects detection and classification*. In: *The Journal of Engineering* 2020.13 (2020), pp. 303–309.

[115] **Huang, Y., J. Jing, and Z. Wang**. *Fabric defect segmentation method based on deep learning*. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–15.

[116] **Huber, C., D. Knoll, and M. Guthe**. *Fully-Synthetic Training for Visual Quality Inspection in Automotive Production*. In: *arXiv preprint* (2025). arXiv: 2503.09354.

[117] **Hussain, M.** *YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection*. In: *Machines* 11.7 (2023), p. 677.

[118] **Hütten, N., R. Meyes, and T. Meisen**. *Vision transformer in industrial visual inspection*. In: *Applied Sciences* 12.23 (2022), p. 11981.

[119] **Hütten, N., M. Alves Gomes, F. Hölken, K. Andricevic, R. Meyes, and T. Meisen**. *Deep learning for automated visual inspection in manufacturing and maintenance: a survey of open-access papers*. In: *Applied System Innovation* 7.1 (2024), p. 11.

[120] **Ikeda, T., S. Tanishige, A. Amma, M. Sudano, H. Audren, and K. Nishiwaki**. *Sim2real instance-level style transfer for 6d pose estimation*. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 3225–3232.

[121] **Imbusch, B. T., M. Schwarz, and S. Behnke**. *Synthetic-to-real domain adaptation using contrastive unpaired translation*. In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2022, pp. 595–602.

[122] **Ioffe, S. and C. Szegedy**. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[123] **Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros**. *Image-to-image translation with conditional adversarial networks*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[124] **Jain, S., G. Seth, A. Paruthi, U. Soni, and G. Kumar**. *Synthetic data augmentation for surface defect detection and classification using deep learning*. In: *Journal of Intelligent Manufacturing* 33.4 (2022), pp. 1007–1020.

[125] **Jayasumana, S., S. Ramalingam, A. Veit, D. Glasner, A. Chakrabarti, and S. Kumar**. *Rethinking fid: Towards a better evaluation metric for image generation*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9307–9315.

[126] **Jezek, S., M. Jonak, R. Burget, P. Dvorak, and M. 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)*. 2021, pp. 66–71.

[127] **Ji, S. and H. Zhang**. *ISAT with Segment Anything: An Interactive Semi-Automatic Annotation Tool*. Version 1.33. Updated on 2025-02-07. 2024.

[128] **Jiang, K., T. Xie, R. Yan, X. Wen, D. Li, H. Jiang, N. Jiang, L. Feng, X. Duan, and J. Wang**. *An attention mechanism-improved YOLOv7 object detection algorithm for hemp duck count estimation*. In: *Agriculture* 12.10 (2022), p. 1659.

[129] **Jiang, P.-T., C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei**. *Layercam: Exploring hierarchical class activation maps for localization*. In: *IEEE transactions on image processing* 30 (2021), pp. 5875–5888.

[130] **Karras, T., S. Laine, and T. Aila**. *A style-based generator architecture for generative adversarial networks*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[131] **Karras, T., M. Aittala, T. Aila, and S. Laine**. *Elucidating the design space of diffusion-based generative models*. In: *Advances in neural information processing systems* 35 (2022), pp. 26565–26577.

[132] **Kawar, B., M. Elad, S. Ermon, and J. Song**. *Denoising diffusion restoration models*. In: *Advances in neural information processing systems* 35 (2022), pp. 23593–23606.

[133] **Keskar, N. S., D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang**. *On large-batch training for deep learning: Generalization gap and sharp minima*. In: *arXiv preprint arXiv:1609.04836* (2016).

[134] **Kim, S., J. Lee, K. Hong, D. Kim, and N. Ahn**. *Diffblender: Scalable and composable multimodal text-to-image diffusion models*. In: *arXiv preprint arXiv:2305.15194* (2023).

[135] **Kingma, D. P.** *Adam: A method for stochastic optimization*. In: *arXiv preprint arXiv:1412.6980* (2014).

[136] **Kingma, D. P. and P. Dhariwal**. *Glow: Generative flow with invertible 1x1 convolutions*. In: *Advances in neural information processing systems* 31 (2018).

[137] **Kirillov, A., E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*** *Segment anything*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026.

[138] **Kleeberger, K., C. Landgraf, and M. F. Huber**. *Large-scale 6d object pose estimation dataset for industrial bin-picking*. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 2573–2578.

[139] **Kotovenko, D., A. Sanakoyeu, S. Lang, and B. Ommer**. *Content and style disentanglement for artistic style transfer*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4422–4431.

[140] **Krizhevsky, A., I. Sutskever, and G. E. Hinton**. *Imagenet classification with deep convolutional neural networks*. In: *Advances in neural information processing systems* 25 (2012).

[141] **Krogh, A. and J. Hertz**. *A simple weight decay can improve generalization*. In: *Advances in neural information processing systems* 4 (1991).

[142] **Kumar, K., P. Kumar, V. Kshirsagar, R. H. Bhalerao, K. Shah, P. K. Vaidhya, and S. K. Panda**. *A real-time object counting and collecting device for industrial automation process using machine vision*. In: *IEEE sensors journal* 23.12 (2023), pp. 13052–13059.

[143] **Kurzman, L., D. Vazquez, and I. Laradji**. *Class-based styling: Real-time localized style transfer with semantic segmentation*. In: *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 2019.

[144] **Kuzmicz, W., W. Pleskacz, J. Raik, and R. Ubar**. *Defect-oriented fault simulation and test generation in digital circuits*. In: *Proceedings of the IEEE 2001. 2nd International Symposium on Quality Electronic Design*. IEEE. 2001, pp. 365–371.

[145] **Kynkäänniemi, T., T. Karras, S. Laine, J. Lehtinen, and T. Aila**. *Improved precision and recall metric for assessing generative models*. In: *Advances in neural information processing systems* 32 (2019).

[146] **LeCun, Y., Y. Bengio, and G. Hinton**. *Deep learning*. In: *nature* 521.7553 (2015), pp. 436–444.

[147] **LeCun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel**. *Handwritten digit recognition with a back-propagation network*. In: *Advances in neural information processing systems* 2 (1989).

[148] **LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner**. *Gradient-based learning applied to document recognition*. In: *Proceedings of the IEEE* 86.11 (2002), pp. 2278–2324.

[149] **Li, C., L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, *et al.*** *YOLOv6: A single-stage object detection framework for industrial applications*. In: *arXiv preprint arXiv:2209.02976* (2022).

[150] **Li, H., Z. Zhang, H. Chen, L. Wu, B. Li, D. Liu, and M. Wang**. *A Novel Approach to Industrial Defect Generation through Blended Latent Diffusion Model with Online Adaptation*. In: *arXiv preprint* (2024). arXiv: 2402.19330.

[151] **Lin, M., Q. Chen, and S. Yan**. *Network in network*. In: *arXiv preprint arXiv:1312.4400* (2013).

[152] **Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár**. *Focal loss for dense object detection*. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[153] **Lin, Z., D. Pathak, B. Li, J. Li, X. Xia, G. Neubig, P. Zhang, and D. Ramanan**. *Evaluating text-to-visual generation with image-to-text generation*. In: *European Conference on Computer Vision*. Springer. 2024, pp. 366–384.

[154] **Lipman, Y., R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le**. *Flow matching for generative modeling*. In: *arXiv preprint arXiv:2210.02747* (2022).

[155] **Liu, A., B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, *et al.*** *Deepseek-v3 technical report*. In: *arXiv preprint arXiv:2412.19437* (2024).

[156] **Liu, D., J. Zhao, A. Xi, C. Wang, X. Huang, K. Lai, and C. Liu**. *Data augmentation technology driven by image style transfer in self-driving car based on end-to-end learning*. In: *Computer Modeling in Engineering & Sciences* 122.2 (2020), pp. 593–617.

[157] **Liu, G., F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro**. *Image inpainting for irregular holes using partial convolutions*. In: *European conference on computer vision*. Springer. 2018, pp. 89–105.

[158] **Liu, H., T. Liu, Z. Zhang, A. K. Sangaiah, B. Yang, and Y. Li**. *ARHPE: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction*. In: *IEEE Transactions on Industrial Informatics* 18.10 (2022), pp. 7107–7117.

[159] **Liu, J., G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin**. *Deep industrial image anomaly detection: A survey*. In: *Machine Intelligence Research* 21.1 (2024), pp. 104–135.

[160] **Liu, X., C. Gong, and Q. Liu**. *Flow straight and fast: Learning to generate and transfer data with rectified flow*. In: *arXiv preprint arXiv:2209.03003* (2022).

[161] **Liu, Z., Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo**. *Swin transformer: Hierarchical vision transformer using shifted windows*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[162] **Liu, Z., Y. Zhou, Y. Xu, and Z. 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.

[163] **Long, J., E. Shelhamer, and T. Darrell**. *Fully convolutional networks for semantic segmentation*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[164] **Loshchilov, I. and F. Hutter**. *Decoupled weight decay regularization*. In: *arXiv preprint arXiv:1711.05101* (2017).

[165] **Lu, D., P. Mausel, E. Brondizio, and E. Moran**. *Change detection techniques*. In: *International journal of remote sensing* 25.12 (2004), pp. 2365–2401.

[166] **Lu, H., M. Du, K. Qian, X. He, and K. Wang**. *GAN-based data augmentation strategy for sensor anomaly detection in industrial robots*. In: *IEEE Sensors Journal* 22.18 (2021), pp. 17464–17474.

[167] **Lugmayr, A., M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool**. *Repaint: Inpainting using denoising diffusion probabilistic models*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11461–11471.

[168] **Ma, M. Q., Y. Zhao, X. Zhang, and L. Akoglu**. *The need for unsupervised outlier model selection: A review and evaluation of internal evaluation strategies*. In: *ACM SIGKDD Explorations Newsletter* 25.1 (2023), pp. 19–35.

[169] **Ma, N., X. Zhang, H.-T. Zheng, and J. Sun**. *Shufflenet v2: Practical guidelines for efficient cnn architecture design*. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 116–131.

[170] **Ma, Y., J. Yin, F. Huang, and Q. Li**. *Surface defect inspection of industrial products with object detection deep networks: A systematic review*. In: *Artificial Intelligence Review* 57.12 (2024), p. 333.

[171] **Maguire, M., S. Dorafshan, and R. J. Thomas**. *SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks*. In: *Data in Brief* 21 (2018).

[172] **Malamas, E. N., E. G. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat**. *A survey on industrial vision systems, applications and tools*. In: *Image and vision computing* 21.2 (2003), pp. 171–188.

[173] **Maleki, S., C. Bingham, and Y. Zhang**. *Development and realization of changepoint analysis for the detection of emerging faults on industrial systems*. In: *IEEE Transactions on Industrial Informatics* 12.3 (2016), pp. 1180–1187.

[174] **Marques, H. O., R. J. Campello, J. Sander, and A. Zimek**. *Internal evaluation of unsupervised outlier detection*. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14.4 (2020), pp. 1–42.

[175] **Matabosch, C., J. Salvi, D. Fofi, and F. Meriaudeau**. *Range image registration for industrial inspection*. In: *Machine Vision Applications in Industrial Inspection XIII*. Vol. 5679. SPIE. 2005, pp. 216–227.

[176] **Maurício, J., I. Domingues, and J. Bernardino**. *Comparing vision transformers and convolutional neural networks for image classification: A literature review*. In: *Applied Sciences* 13.9 (2023), p. 5521.

[177] **Meister, S., N. Möller, J. Stüve, and R. M. Groves**. *Synthetic image data augmentation for fibre layup inspection processes: Techniques to enhance the data set*. In: *Journal of Intelligent Manufacturing* 32.6 (2021), pp. 1767–1789.

[178] **Meng, C., Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon**. *Sdedit: Guided image synthesis and editing with stochastic differential equations*. In: *arXiv preprint arXiv:2108.01073* (2021).

[179] **Milletari, F., N. Navab, and S.-A. Ahmadi**. *V-net: Fully convolutional neural networks for volumetric medical image segmentation*. In: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 565–571.

[180] **Mirza, M. and S. Osindero**. *Conditional generative adversarial nets*. In: *arXiv preprint arXiv:1411.1784* (2014).

[181] **Mohtashami, A., M. Jaggi, and S. Stich**. *Masked training of neural networks with partial gradients*. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 5876–5890.

[182] **Monnet, J., O. Petrovic, and W. Herfs**. *Investigating the generation of synthetic data for surface defect detection: A comparative analysis*. In: *Procedia CIRP* 130 (2024), pp. 767–773.

[183] **Moonen, S., B. Vanherle, J. de Hoog, T. Bourgana, A. Bey-Temsamani, and N. Michiels**. *Cad2render: A modular toolkit for gpu-accelerated photorealistic synthetic data generation for the manufacturing industry*. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2023, pp. 583–592.

[184] **Mourtzis, D.** *Simulation in the design and operation of manufacturing systems: state of the art and new trends*. In: *International Journal of Production Research* 58.7 (2020), pp. 1927–1949.

[185] **Mullick, K., H. Jain, S. Gupta, and A. A. Kale**. *Domain adaptation of synthetic driving datasets for real-world autonomous driving*. In: *arXiv preprint arXiv:2302.04149* (2023).

[186] **Mumuni, A. and F. Mumuni**. *Data augmentation: A comprehensive survey of modern approaches*. In: *Array* 16 (2022), p. 100258.

[187] **Naeem, M. F., S. J. Oh, Y. Uh, Y. Choi, and J. Yoo**. *Reliable fidelity and diversity metrics for generative models*. In: *International conference on machine learning*. PMLR. 2020, pp. 7176–7185.

[188] **Nair, N. G., J. M. J. Valanarasu, and V. M. Patel**. *Maxfusion: Plug&play multi-modal generation in text-to-image diffusion models*. In: *European Conference on Computer Vision*. Springer. 2024, pp. 93–110.

[189] **Naveed, H., A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian**. *A comprehensive overview of large language models*. In: *ACM Transactions on Intelligent Systems and Technology* 16.5 (2025), pp. 1–72.

[190] **Nesterov, Y.** *A method for solving the convex programming problem with convergence rate O (1/k2)*. In: *Dokl akad nauk Sssr*. Vol. 269. 1983, p. 543.

[191] **Neyshabur, B., S. Bhojanapalli, D. McAllester, and N. Srebro**. *Exploring generalization in deep learning*. In: *Advances in neural information processing systems* 30 (2017).

[192] **Nguyen, T. T., U. Q. Nguyen, *et al.*** *An evaluation method for unsupervised anomaly detection algorithms*. In: *Journal of Computer Science and Cybernetics* 32.3 (2016), pp. 259–272.

[193] **Nichol, A., P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen**. *Glide: Towards photorealistic image generation and editing with text-guided diffusion models*. In: *arXiv preprint arXiv:2112.10741* (2021).

[194] **Nikolenko, S. I.** *Synthetic-to-real domain adaptation and refinement*. In: *Synthetic data for deep learning*. Springer, 2021, pp. 235–268.

[195] **Odena, A., C. Olah, and J. Shlens**. *Conditional image synthesis with auxiliary classifier gans*. In: *International conference on machine learning*. PMLR. 2017, pp. 2642–2651.

[196] **Oquab, M., T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*** *Dinov2: Learning robust visual features without supervision*. In: *arXiv preprint arXiv:2304.07193* (2023).

[197] **Pan, T., J. Chen, T. Zhang, S. Liu, S. He, and H. Lv**. *Generative adversarial network in mechanical fault diagnosis under small sample: A systematic review on applications and future perspectives*. In: *ISA transactions* 128 (2022), pp. 1–10.

[198] **Pang, G., C. Shen, L. Cao, and A. V. D. Hengel**. *Deep learning for anomaly detection: A review*. In: *ACM computing surveys (CSUR)* 54.2 (2021), pp. 1–38.

[199] **Papamakarios, G., E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan**. *Normalizing flows for probabilistic modeling and inference*. In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64.

[200] **Park, D. Y. and K. H. Lee**. *Arbitrary style transfer with style-attentional networks*. In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5880–5888.

[201] **Park, J.-M., J.-H. Jang, S.-M. Yoo, S.-K. Lee, U.-H. Kim, and J.-H. Kim**. *Changesim: Towards end-to-end online scene change detection in industrial indoor environments*. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 8578–8585.

[202] **Park, T., A. A. Efros, R. Zhang, and J.-Y. Zhu**. *Contrastive learning for unpaired image-to-image translation*. In: *European conference on computer vision*. Springer. 2020, pp. 319–345.

[203] **Peebles, W. and S. Xie**. *Scalable diffusion models with transformers*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4195–4205.

[204] **Peng, X., B. Sun, K. Ali, and K. Saenko**. *Learning deep object detectors from 3d models*. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1278–1286.

[205] **Peng, X., B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko**. *Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation*. In: *arXiv preprint arXiv:1806.09755* (2018).

[206] **Perera, P., R. Nallapati, and B. Xiang**. *Ocgan: One-class novelty detection using gans with constrained latent representations*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2898–2906.

[207] **Peres, R. S., M. Guedes, F. Miranda, and J. Barata**. *Simulation-based data augmentation for the quality inspection of structural adhesive with deep learning*. In: *IEEE Access* 9 (2021), pp. 76532–76541.

[208] **Petsiuk, A., H. Singh, H. Dadhwal, and J. M. Pearce**. *Synthetic-to-real composite semantic segmentation in additive manufacturing*. In: *Journal of Manufacturing and Materials Processing* 8.2 (2024), p. 66.

[209] **Planche, B., Z. Wu, K. Ma, S. Sun, S. Kluckner, O. Lehmann, T. Chen, A. Hutter, S. Zakharov, H. Kosch, *et al.*** *Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition*. In: *2017 International conference on 3d vision (3DV)*. IEEE. 2017, pp. 1–10.

[210] **Podell, D., Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach**. *Sdxl: Improving latent diffusion models for high-resolution image synthesis*. In: *arXiv preprint arXiv:2307.01952* (2023).

[211] **Polyak, B. T.** *Some methods of speeding up the convergence of iteration methods*. In: *Ussr computational mathematics and mathematical physics* 4.5 (1964), pp. 1–17.

[212] **Poole, B., A. Jain, J. T. Barron, and B. Mildenhall**. *Dreamfusion: Text-to-3d using 2d diffusion*. In: *arXiv preprint arXiv:2209.14988* (2022).

[213] **Prasad, V. and M. Arashpour**. *Optimally leveraging depth features to enhance segmentation of recyclables from cluttered construction and demolition waste streams*. In: *Journal of Environmental Management* 354 (2024), p. 120313.

[214] **Prasad, V. and M. Arashpour**. *Real-time instance segmentation of recyclables from highly cluttered construction and demolition waste streams*. In: *Journal of Environmental Management* 372 (2024), p. 123365.

[215] **Prechelt, L.** *Early stopping-but when?* In: *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[216] **Qi, S., J. Yang, and Z. Zhong**. *A review on industrial surface defect detection based on deep learning technology*. In: *Proceedings of the 2020 3rd international conference on machine learning and machine intelligence*. 2020, pp. 24–30.

[217] **Radford, A., L. Metz, and S. Chintala**. *Unsupervised representation learning with deep convolutional generative adversarial networks*. In: *arXiv preprint arXiv:1511.06434* (2015).

[218] **Ramesh, A., P. Dhariwal, A. Nichol, C. Chu, and M. Chen**. *Hierarchical text-conditional image generation with clip latents*. In: *arXiv preprint arXiv:2204.06125* 1.2 (2022), p. 3.

[219] **Rane, N.** *YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities*. In: *Available at SSRN 4624206* (2023).

[220] **Ravi, N., V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*** *Sam 2: Segment anything in images and videos*. In: *arXiv preprint arXiv:2408.00714* (2024).

[221] **Reddi, S. J., S. Kale, and S. Kumar**. *On the convergence of adam and beyond*. In: *arXiv preprint arXiv:1904.09237* (2019).

[222] **Ren, R., T. Hung, and K. C. Tan**. *A generic deep-learning-based approach for automated surface inspection*. In: *IEEE transactions on cybernetics* 48.3 (2017), pp. 929–940.

[223] **Riesenhuber, M. and T. Poggio**. *Hierarchical models of object recognition in cortex*. In: *Nature neuroscience* 2.11 (1999), pp. 1019–1025.

[224] **Robbins, H. and S. Monro**. *A stochastic approximation method*. In: *The annals of mathematical statistics* (1951), pp. 400–407.

[225] **Rocha, L. F., M. Ferreira, V. Santos, and A. P. Moreira**. *Object recognition and pose estimation for industrial applications: A cascade system*. In: *Robotics and Computer-Integrated Manufacturing* 30.6 (2014), pp. 605–621.

[226] **Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer**. *High-resolution image synthesis with latent diffusion models*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[227] **Ronneberger, O., P. Fischer, and T. Brox**. *U-net: Convolutional networks for biomedical image segmentation*. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[228] **Roth, K., L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. 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.

[229] **Rublee, E., V. Rabaud, K. Konolige, and G. Bradski**. *ORB: An efficient alternative to SIFT or SURF*. In: *2011 International conference on computer vision*. IEEE. 2011, pp. 2564–2571.

[230] **Ruder, M., A. Dosovitskiy, and T. Brox**. *Artistic style transfer for videos*. In: *German conference on pattern recognition*. Springer. 2016, pp. 26–36.

[231] **Ruff, L., R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft**. *Deep one-class classification*. In: *International conference on machine learning*. PMLR. 2018, pp. 4393–4402.

[232] **Sader, S. A., M. Bertrand, and E. H. Wilson**. *Satellite change detection of forest harvest patterns on an industrial forest landscape*. In: *Forest Science* 49.3 (2003), pp. 341–353.

[233] **Saharia, C., W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, *et al.*** *Photorealistic text-to-image diffusion models with deep language understanding*. In: *Advances in neural information processing systems* 35 (2022), pp. 36479–36494.

[234] **Saiz, F. A., G. Alfaro, I. Barandiaran, S. Garcia, M. Carretero, and M. Graña**. *Synthetic Data Set Generation for the Evaluation of Image Acquisition Strategies Applied to Deep Learning Based Industrial Component Inspection Systems.* In: *CEIG*. 2021, pp. 1–8.

[235] **Sajjadi, M. S., O. Bachem, M. Lucic, O. Bousquet, and S. Gelly**. *Assessing generative models via precision and recall*. In: *Advances in neural information processing systems* 31 (2018).

[236] **Salimans, T., I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen**. *Improved techniques for training gans*. In: *Advances in neural information processing systems* 29 (2016).

[237] **Salvi, J., C. Matabosch, D. Fofi, and J. Forest**. *A review of recent range image registration methods with accuracy evaluation*. In: *Image and Vision computing* 25.5 (2007), pp. 578–596.

[238] **Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen**. *Mobilenetv2: Inverted residuals and linear bottlenecks*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[239] **Santurkar, S., D. Tsipras, A. Ilyas, and A. Madry**. *How does batch normalization help optimization?* In: *Advances in neural information processing systems* 31 (2018).

[240] **Sass, L.** *Synthesis of design production with integrated digital fabrication*. In: *Automation in construction* 16.3 (2007), pp. 298–310.

[241] **Schlagenhauf, T. and M. Landwehr**. *Industrial machine tool component surface defect dataset*. In: *Data in Brief* 39 (2021), p. 107643.

[242] **Schmedemann, O., M. Baaß, D. Schoepflin, and T. Schüppstuhl**. *Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection*. In: *Procedia CIRP* 107 (2022), pp. 1101–1106.

[243] **Schmedemann, O., S. Schlodinski, D. Holst, and T. Schüppstuhl**. *Adapting synthetic training data in deep learning-based visual surface inspection to improve transferability of simulations to real-world environments*. In: *Automated Visual Inspection and Machine Vision V*. Vol. 12623. SPIE. 2023, pp. 25–35.

[244] **Schnieders, B., S. Luo, G. Palmer, and K. Tuyls**. *Fully convolutional one-shot object segmentation for industrial robotics*. In: *arXiv preprint arXiv:1903.00683* (2019).

[245] **Schölkopf, B., J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson**. *Estimating the support of a high-dimensional distribution*. In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[246] **Schraml, D. and G. Notni**. *Synthetic training data in AI-driven quality inspection: The significance of camera, lighting, and noise parameters*. In: *Sensors* 24.2 (2024), p. 649.

[247] **Shorten, C. and T. M. Khoshgoftaar**. *A survey on image data augmentation for deep learning*. In: *Journal of big data* 6.1 (2019), pp. 1–48.

[248] **Shrivastava, A., T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb**. *Learning from simulated and unsupervised images through adversarial training*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2107–2116.

[249] **Simonyan, K. and A. Zisserman**. *Very deep convolutional networks for large-scale image recognition*. In: *arXiv preprint arXiv:1409.1556* (2014).

[250] **Smith, L. N.** *A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay*. In: *arXiv preprint arXiv:1803.09820* (2018).

[251] **Song, J., D. Park, K. Baek, S. Lee, J. Choi, E. Kim, and S. Yoon**. *DefectFill: Realistic Defect Generation with Inpainting Diffusion Model for Visual Inspection*. In: *arXiv preprint*. 2025. arXiv: 2503.13985.

[252] **Song, J., C. Meng, and S. Ermon**. *Denoising diffusion implicit models*. In: *arXiv preprint arXiv:2010.02502* (2020).

[253] **Song, Y. and S. Ermon**. *Generative modeling by estimating gradients of the data distribution*. In: *Advances in neural information processing systems* 32 (2019).

[254] **Song, Y., J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole**. *Score-based generative modeling through stochastic differential equations*. In: *arXiv preprint arXiv:2011.13456* (2020).

[255] **Song, Y., P. Dhariwal, M. Chen, and I. Sutskever**. *Consistency models*. In: *arXiv preprint arXiv:2303.01469* (2023).

[256] **Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov**. *Dropout: a simple way to prevent neural networks from overfitting*. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[257] **Strudel, R., R. Garcia, I. Laptev, and C. Schmid**. *Segmenter: Transformer for semantic segmentation*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 7262–7272.

[258] **Su, H., C. R. Qi, Y. Li, and L. J. Guibas**. *Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views*. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2686–2694.

[259] **Sutskever, I., J. Martens, G. Dahl, and G. Hinton**. *On the importance of initialization and momentum in deep learning*. In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147.

[260] **Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna**. *Rethinking the inception architecture for computer vision*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[261] **Tabernik, D., S. Šela, J. Skvarč, and D. Skočaj**. *Segmentation-based deep-learning approach for surface-defect detection*. In: *Journal of Intelligent Manufacturing* 31.3 (2020), pp. 759–776.

[262] **Tang, S., F. He, X. Huang, and J. Yang**. *Online PCB defect detector on a new PCB defect dataset*. In: *arXiv preprint arXiv:1902.06197* (2019).

[263] **Tao, X., X. Gong, X. Zhang, S. Yan, and C. Adak**. *Deep learning for unsupervised anomaly localization in industrial images: A survey*. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–21.

[264] **Tax, D. M. and R. P. Duin**. *Support vector data description*. In: *Machine learning* 54.1 (2004), pp. 45–66.

[265] **Team, G., R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, *et al.*** *Gemini: a family of highly capable multimodal models*. In: *arXiv preprint arXiv:2312.11805* (2023).

[266] **Tenscher-Philipp, R., T. Schanz, Y. Wunderle, P. Lickert, and M. Simon**. *Generative Synthesis of Defects in Industrial Computed Tomography Data*. In: (2023).

[267] **Tobin, J., R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel**. *Domain randomization for transferring deep neural networks from simulation to the real world*. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[268] **Touvron, H., M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou**. *Training data-efficient image transformers & distillation through attention*. In: *International conference on machine learning*. PMLR. 2021, pp. 10347–10357.

[269] **Tremblay, J., T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield**. *Deep object pose estimation for semantic robotic grasping of household objects*. In: *arXiv preprint arXiv:1809.10790* (2018).

[270] **Tremblay, J., A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield**. *Training deep networks with synthetic data: Bridging the reality gap by domain randomization*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 969–977.

[271] **Van Dyk, D. A. and X.-L. Meng**. *The art of data augmentation*. In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50.

[272] **Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin**. *Attention is all you need*. In: *Advances in neural information processing systems* 30 (2017).

[273] **Wada, K., S. Kitagawa, K. Okada, and M. Inaba**. *Instance segmentation of visible and occluded regions for finding and picking target from a pile of objects*. In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2018, pp. 2048–2055.

[274] **Wang, C., W. Zhu, B.-B. Gao, Z. Gan, J. Zhang, Z. Gu, S. Qian, M. Chen, and L. Ma**. *Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 22883–22892.

[275] **Wang, C.-Y., I.-H. Yeh, and H.-Y. Mark Liao**. *Yolov9: Learning what you want to learn using programmable gradient information*. In: *European conference on computer vision*. Springer. 2024, pp. 1–21.

[276] **Wang, J., L. Perez, *et al.*** *The effectiveness of data augmentation in image classification using deep learning*. In: *Convolutional Neural Networks Vis. Recognit* 11.2017 (2017), pp. 1–8.

[277] **Wang, J., Z. Liu, L. Zhao, Z. Wu, C. Ma, S. Yu, H. Dai, Q. Yang, Y. Liu, S. Zhang, *et al.*** *Review of large vision models and visual prompt engineering*. In: *Meta-Radiology* 1.3 (2023), p. 100047.

[278] **Wang, K., F. Shi, W. Wang, Y. Nan, and S. Lian**. *Synthetic data generation and adaption for object detection in smart vending machines*. In: *arXiv preprint arXiv:1904.12294* (2019).

[279] **Wang, Q., S. Gao, J. Hu, J. Yu, X. Tong, Y. Li, and W. Zhang**. *HSS-IAD: A Heterogeneous Same-Sort Industrial Anomaly Detection Dataset*. In: *arXiv preprint arXiv:2504.12689* (2025).

[280] **Wang, R., S. Schmedding, and M. F. Huber**. *Improving the effectiveness of deep generative data*. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2024, pp. 4910–4920.

[281] **Wang, T., Y. Chen, M. Qiao, and H. Snoussi**. *A fast and robust convolutional neural network-based defect detection model in product quality control*. In: *The International Journal of Advanced Manufacturing Technology* 94.9 (2018), pp. 3465–3471.

[282] **Wang, T.-C., M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro**. *High-resolution image synthesis and semantic manipulation with conditional gans*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8798–8807.

[283] **Wang, Z., J. Fan, F. Jing, Z. Liu, and M. Tan**. *A pose estimation system based on deep neural network and ICP registration for robotic spray painting application*. In: *The International Journal of Advanced Manufacturing Technology* 104.1 (2019), pp. 285–299.

[284] **Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli**. *Image quality assessment: from error visibility to structural similarity*. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[285] **Wei, T., D. Cao, X. Jiang, C. Zheng, and L. Liu**. *Defective samples simulation through neural style transfer for automatic surface defect segment*. In: *2019 International Conference on Optical Instruments and Technology: Optoelectronic Measurement Technology and Systems*. Vol. 11439. SPIE. 2020, pp. 15–26.

[286] **Weimer, D., B. Scholz-Reiter, and M. Shpitalni**. *Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection*. In: *CIRP annals* 65.1 (2016), pp. 417–420.

[287] **Wen, L., Y. Wang, and X. Li**. *A new cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples*. In: *IEEE Transactions on Industrial Informatics* 18.12 (2022), pp. 8988–8998.

[288] **Wen, Q., L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu**. *Time series data augmentation for deep learning: A survey*. In: *arXiv preprint arXiv:2002.12478* (2020).

[289] **Wu, X., Y. Zhang, C. Cheng, and Z. Peng**. *A hybrid classification autoencoder for semi-supervised fault diagnosis in rotating machinery*. In: *Mechanical Systems and Signal Processing* 149 (2021), p. 107327.

[290] **Xavier, R. J., C. F. Viegas, B. C. Costa, and R. P. Ishii**. *A Deep Learning Approach for Automatic Counting of Bales and Product Boxes in Industrial Production Lines*. In: *International Conference on Computational Science and Its Applications*. Springer. 2022, pp. 619–633.

[291] **Xiao, B. and S.-C. Kang**. *Development of an image data set of construction machines for deep learning object detection*. In: *Journal of Computing in Civil Engineering* 35.2 (2021), p. 05020005.

[292] **Xiao, S., J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim**. *Robust facial landmark detection via recurrent attentive-refinement networks*. In: *European conference on computer vision*. Springer. 2016, pp. 57–72.

[293] **Xie, Q., Z. Dai, E. Hovy, T. Luong, and Q. Le**. *Unsupervised data augmentation for consistency training*. In: *Advances in neural information processing systems* 33 (2020), pp. 6256–6268.

[294] **Xu, L., H. Liu, B. Xiao, X. Luo, Z. Zhu, *et al.*** *A systematic review and evaluation of synthetic simulated data generation strategies for deep learning applications in construction*. In: *Advanced Engineering Informatics* 62 (2024), p. 102699.

[295] **Xu, X., Y. Jiang, W. Chen, Y. Huang, Y. Zhang, and X. Sun**. *Damo-yolo: A report on real-time object detection design*. In: *arXiv preprint arXiv:2211.15444* (2022).

[296] **Xu, Y., S. Arai, D. Liu, F. Lin, and K. Kosuge**. *FPCC: Fast point cloud clustering-based instance segmentation for industrial bin-picking*. In: *Neurocomputing* 494 (2022), pp. 255–268.

[297] **Yang, L., Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang**. *Diffusion models: A comprehensive survey of methods and applications*. In: *ACM computing surveys* 56.4 (2023), pp. 1–39.

[298] **Yang, M., P. Wu, and H. Feng**. *MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities*. In: *Engineering Applications of Artificial Intelligence* 119 (2023), p. 105835.

[299] **Yao, Y., L. Rosasco, and A. Caponnetto**. *On early stopping in gradient descent learning*. In: *Constructive approximation* 26.2 (2007), pp. 289–315.

[300] **Yi, Z., H. Zhang, P. Tan, and M. Gong**. *Dualgan: Unsupervised dual learning for image-to-image translation*. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2849–2857.

[301] **Yoon, Y., G. N. DeSouza, and A. C. Kak**. *Real-time tracking and pose estimation for industrial objects using geometric features*. In: *2003 IEEE International conference on robotics and automation (cat. no. 03CH37422)*. Vol. 3. IEEE. 2003, pp. 3473–3478.

[302] **Yu, J., Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang**. *Free-form image inpainting with gated convolution*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4471–4480.

[303] **Yu, Y., H. Li, J. Cao, and X. Luo**. *Three-dimensional working pose estimation in industrial scenarios with monocular camera*. In: *IEEE Internet of Things Journal* 8.3 (2020), pp. 1740–1748.

[304] **Yun, J. P., W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee**. *Automated defect inspection system for metal surfaces based on deep learning and data augmentation*. In: *Journal of Manufacturing Systems* 55 (2020), pp. 317–324.

[305] **Zaech, J.-N., D. Dai, M. Hahner, and L. Van Gool**. *Texture underfitting for domain adaptation*. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 547–552.

[306] **Zaheer, M. Z., J.-H. Lee, A. Mahmood, M. Astrid, and S.-I. Lee**. *Stabilizing adversarially learned one-class novelty detection using pseudo anomalies*. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 5963–5975.

[307] **Zeiler, M. D. and R. Fergus**. *Visualizing and understanding convolutional networks*. In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[308] **Zeiler, M. D., D. Krishnan, G. W. Taylor, and R. Fergus**. *Deconvolutional networks*. In: *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE. 2010, pp. 2528–2535.

[309] **Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals**. *Understanding deep learning requires rethinking generalization*. In: *arXiv preprint arXiv:1611.03530* (2016).

[310] **Zhang, G., K. Cui, T.-Y. Hung, and S. Lu**. *Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection*. In: *arXiv preprint arXiv:2103.15158* (2021).

[311] **Zhang, L., A. Rao, and M. Agrawala**. *Adding conditional control to text-to-image diffusion models*. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 3836–3847.

[312] **Zhang, R., P. Isola, A. A. Efros, E. Shechtman, and O. Wang**. *The unreasonable effectiveness of deep features as a perceptual metric*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.

[313] **Zhang, W., X. Li, H. Ma, Z. Luo, and X. Li**. *Federated learning for machinery fault diagnosis with dynamic validation and self-supervision*. In: *Knowledge-Based Systems* 213 (2021), p. 106679.

[314] **Zhang, Y., P. Yu, and Y. N. Wu**. *Object-conditioned energy-based attention map alignment in text-to-image diffusion models*. In: *European Conference on Computer Vision*. Springer. 2024, pp. 55–71.

[315] **Zhang, Y., Z. Shi, and C. Zhuang**. *Instance Segmentation of Low-texture Industrial Parts Based on Deep Learning*. In: *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE. 2021, pp. 756–761.

[316] **Zhang, Z., Z. Zhao, X. Zhang, C. Sun, and X. Chen**. *Industrial Anomaly Detection with Domain Shift: A Real-world Dataset and Masked Multi-scale Reconstruction*. In: *arXiv preprint arXiv:2304.02216* (2023).

[317] **Zhang, Z., Y. Zou, Y. Tan, and C. Zhou**. *YOLOv8-seg-CP: a lightweight instance segmentation algorithm for chip pad based on improved YOLOv8-seg model*. In: *Scientific Reports* 14.1 (2024), p. 27716.

[318] **Zhao, R., R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao**. *Deep learning and its applications to machine health monitoring*. In: *Mechanical Systems and Signal Processing* 115 (2019), pp. 213–237.

[319] **Zhao, Y., S. Kong, D. Shin, and C. Fowlkes**. *Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation*. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 3327–3337.

[320] **Zheng, S., J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, *et al.*** *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[321] **Zhou, X., X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, and Q. Jin**. *Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems*. In: *IEEE Transactions on Industrial Informatics* 18.2 (2021), pp. 1377–1386.

[322] **Zhu, J.-Y., T. Park, P. Isola, and A. A. Efros**. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

[323] **Zhu, L., B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang**. *Vision mamba: Efficient visual representation learning with bidirectional state space model*. In: *arXiv preprint arXiv:2401.09417* (2024).

[324] **Zhu, X., T. Bilal, P. Mårtensson, L. Hanson, M. Björkman, and A. Maki**. *Towards sim-to-real industrial parts classification with synthetic dataset*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 4454–4463.

[325] **Zhu, X., P. Mårtensson, L. Hanson, M. Björkman, and A. Maki**. *Automated assembly quality inspection by deep learning with 2D and 3D synthetic CAD data*. In: *Journal of Intelligent Manufacturing* 36.4 (2025), pp. 2567–2582.

[326] **Zhu, X., J. Henningsson, D. Li, P. Mårtensson, L. Hanson, M. Björkman, and A. Maki**. *Towards Automated Assembly Quality Inspection with Synthetic Data and Domain Randomization*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025, pp. 1395–1403.

[327] **Zou, Y., J. Jeong, L. Pemula, D. Zhang, and O. Dabeer**. *Spot-the-difference self-supervised pre-training for anomaly detection and segmentation*. In: *European conference on computer vision*. Springer. 2022, pp. 392–408.

List of publications

[328] **Chu, A., X. Xie, M. Jin, and W. Stork**. *Explainable ai (xai) for spectral analysis via reinforcement learning: Learning to optimize*. In: *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2025, pp. 4609–4615.

[329] **Jin, M. and M. Heizmann**. *Cutout as augmentation in contrastive learning for detecting burn marks in plastic granules*. In: *Journal of Sensors and Sensor Systems* 13.1 (2024), pp. 63–69.

[330] **Jin, M. and M. Heizmann**. *Embedding Stability-Guided Early-Stopping for Industrial Visual Anomaly Detection*. In: *2026 11th International Conference on Machine Learning Technologies (ICMLT)*. IEEE. 2026.

[331] **Jin, M., S. Wezstein, A. Chu, and M. Heizmann**. *Improving Particle Sorting using Style-Transferred Synthetic Images*. In: *Technisches Messen* 91.1 (2024).

[332] **Jin, M., J. Baßler, V. Eichinger, and M. Heizmann**. *From Synthetic Cracks to Real Inspection: Data- and Model-Level Control of Domain Shift*. In: *2026 11th International Conference on Machine Learning Technologies (ICMLT)*. IEEE. 2026.

[333] **Wezstein, S., M. Jin, M. Stelzl, and M. Heizmann**. *Fast semantic segmentation CNNs for FPGAs*. In: *Forum Bildverarbeitung 2024*. KIT Scientific Publishing. 2024, p. 123.

[334] **Wezstein, S., M. Jin, M. Stelzl, and M. Heizmann**. *Comparing fast semantic segmentation CNNs for FPGAs with standard methods*. In: *tm-Technisches Messen* 92.7-8 (2025), pp. 311–320.

[335] **Xie, X., M. Jin, A. Chu, and W. Stork**. *Minimal cost device calibration in spectral analysis via meta learning: Towards efficient deployment of deep neural networks in industry*. In: *2022 IEEE International Conference on Big Data (Big Data)*. IEEE. 2022, pp. 2123–2132.

List of supervised theses

[336] **Baßler, J.** *Studie zu Anomaliedetektionsverfahren und Implementierung auf ressourcenbeschränkten Geräten*. Master thesis. Karlsruher Institut für Technologie (KIT), 2023.

[337] **Liu, J.** *Against label flip in industrial anomaly detection based on PatchCore*. Master thesis. Karlsruher Institut für Technologie (KIT), 2023.

[338] **Thanasaj, R.** *Converting Static Images of Industrial Objects into 3D Models for Data Augmentation*. Master thesis. Karlsruher Institut für Technologie (KIT), 2024.

[339] **Zhang, F.** *Generalization ability of AnomalyGPT on complex industrial dataset*. Master thesis. Karlsruher Institut für Technologie (KIT), 2024.