ITESO
The Jesuit University of Guadalajara

Official recognition of the validity of higher education studies according to secretarial agreement 15018, published in the Official Gazette of the Federation on November 29, 1976.

Department of Electronics, Systems and Informatics

Master in Computer Systems

STABLE DIFFUSION FOR THE SYNTHETIC IMAGE GENERATION OF MANUFACTURING DEFECTS: AN EVALUATION AGAINST A CVAE-GAN BASELINE MODEL

THESIS WORK presented to obtain the **DEGREE** of **MASTER IN COMPUTER SYSTEMS**

Presents: Estiven Angel Echeverria

Director: Victor Hugo Martínez Sánchez

Tlaquepaque, Jalisco. August 10, 2026

Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

STABLE DIFFUSION PARA LA GENERACIÓN SINTÉTICA DE IMÁGENES DE DEFECTOS DE MANUFACTURA: UNA EVALUACIÓN FRENTE A UN MODELO DE REFERENCIA CVAE-GAN

TRABAJO RECEPCIONAL que para obtener el **GRADO** de **MAESTRO EN SISTEMAS COMPUTACIONALES**

Presenta: Estiven Angel Echeverria

Director: Victor Hugo Martínez Sánchez

Tlaquepaque, Jalisco. August 10, 2026

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the following institutions and individuals for their support throughout this graduate program.

First, I would like to acknowledge the financial support that made this milestone in my academic life possible. I extend my gratitude to the Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) for the institutional scholarship and educational financing provided, as well as to the Secretariat of Innovation, Science, and Technology of the State of Jalisco (SICYT) for their support through a supplementary scholarship.

I also wish to express my deepest gratitude, recognition, and respect to my Thesis Advisor, Dr. Victor Hugo Martínez Sánchez, for his continuous guidance, technical insights, patience, advice, and unconditional support throughout the entirety of this work. His experience and dedication were the foundation for the development of this research.

I wish to thank the evaluation committee of this thesis, composed of Dr. Iván Esteban Villalón Turribiates and Dr. Yehoshua Aguilar Molina, for their valuable contributions, suggestions, and constructive criticism that enriched this work.

I wish to thank my professors; not only those at this institution who accompanied me during this graduate training process, but also all those who planted the seed of knowledge and passion for engineering during my undergraduate studies. A special recognition to Dr. Yehoshua Aguilar Molina, for always believing in and supporting my vocation toward engineering and teaching; and to M.Eng. Ramón Enrique González Ángel, for instilling his passion for engineering and teaching me the importance of quality in the formation of professionals; *rest in peace, my dear professor Ramón.*

Finally, I wish to extend these acknowledgments to my personal circle: my girlfriend, my family, friends, and colleagues who were present and attentive throughout this journey. To my mother, for always believing that *I am the smartest person on the face of the Earth...* To my father, for teaching me what hard work is and not allowing anything or anyone to intimidate me in life; *rest in peace, Scorpion.* To my brothers, who were always present through thick and thin (*mostly thin...*). To my friends and colleagues from work and this master's program, who were the biggest distraction and greatest barrier to finishing my assignments and my thesis thanks to their constant invitations to "go out"... *thank you, my highly esteemed friends.* To my friend Hassan Cabrales Reyes for proving to me that IQ can indeed decrease severely... *thank you, dear friend.* And to anyone else who influenced the culmination of this goal and whom I am probably forgetting: *my apologies, and thank you very much.*

AGRADECIMIENTOS

El autor desea expresar su agradecimiento a las siguientes instituciones y personas por su apoyo a lo largo de este programa de posgrado.

En primer lugar, quiero reconocer el apoyo financiero que hizo posible este hito en mi vida académica. Extiendo mi agradecimiento al Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) por la beca institucional y el financiamiento educativo otorgados, así como a la Secretaría de Innovación, Ciencia y Tecnología del Estado de Jalisco (SICYT) por su apoyo mediante una beca complementaria.

Deseo también expresar mi más profunda gratitud, reconocimiento y respeto a mi director de Trabajo de Obtención de Grado (TOG), el Dr. Victor Hugo Martínez Sánchez, por su continua guía, aportes técnicos, paciencia, consejos y apoyo incondicional durante todo el proceso de este trabajo. Su experiencia y dedicación fueron los cimientos para el desarrollo de esta investigación.

Deseo agradecer la participación del comité de evaluación de este Trabajo de Obtención de Grado, integrado por el Dr. Iván Esteban Villalón Turribiates y el Dr. Yehoshua Aguilar Molina, por sus valiosas aportaciones, sugerencias y críticas constructivas que permitieron enriquecer este trabajo.

Deseo agradecer a mis profesores; no solo a los de esta institución que me acompañaron durante este proceso de formación de posgrado, sino también a todos aquellos que sembraron la semilla del saber y la pasión por la ingeniería durante mis estudios de licenciatura. Un especial reconocimiento al Dr. Yehoshua Aguilar Molina, por siempre creer y apoyar mi vocación hacia la ingeniería y la docencia; y al Mtro. Ramón Enrique González Ángel, por infundir su pasión por la ingeniería y enseñarme la importancia de la calidad en la formación de profesionistas; *descanse en paz, mi querido profe Ramón.*

Finalmente, deseo extender estos agradecimientos a mi círculo personal: mi novia, mi familia, amigos y compañeros que estuvieron presentes y pendientes durante todo este camino. A mi madre, por siempre creer que *soy la persona más inteligente sobre la faz de la Tierra...* A mi padre, por enseñarme lo que es el trabajo duro y no permitir que nada ni nadie me intimide en la vida; *descansa en paz, Scorpion.* A mis hermanos, que siempre estuvieron presentes en las buenas y en las malas (*más malas que buenas...*). A mis amigos y compañeros del trabajo y de esta maestría, quienes fueron la mayor distracción y más grande barrera para poder terminar mis tareas y mi tesis gracias a sus constantes invitaciones a "salir"... *gracias, mis estimadísimos.* A mi amigo Hassan Cabrales Reyes por demostrarme que el coeficiente intelectual sí puede disminuir severamente... *gracias, querido amigo.* Y a toda aquella persona que influyó en la culminación de esta meta y que probablemente estoy olvidando: *una disculpa, y muchas gracias.*

DEDICATION

The author dedicates this thesis to ***Rubí Moreno Zazueta****,*

To you, because without all your emotional support, patience, and encouragement, this work would not have been possible. For every word, every hug, every cheer, every laugh, every warm meal, every outing to clear my mind, and every canceled plan; for every time you listened to me talk about my battles against each chapter, genuinely trying to understand my doubts and concerns; for every coffee and snack you prepared for me during my long days dedicated both to this research and to every assignment over these two years... and because you were always the motivation and the driving force that pushed me to start and finish this master's degree...

For this, and for everything I may have already forgotten, this achievement is as much yours as it is mine.

Thank you, "Pishosha".

DEDICATORIA

El autor dedica esta tesis a ***Rubí Moreno Zazueta****,*

A ti, porque sin todo tu apoyo emocional, paciencia y soporte, este trabajo no hubiera sido posible. Por cada palabra, cada abrazo, cada ánimo, cada risa, cada comida caliente, cada salida para despejar la mente y cada plan cancelado; por cada vez que me escuchaste hablar de mis batallas contra cada capítulo intentando genuinamente entender mis dudas y preocupaciones; por cada café y botana que me preparaste durante mis largas jornadas dedicadas tanto a esta investigación como a cada tarea durante estos dos años... y porque siempre fuiste la motivación y el motor que me impulsó a comenzar y terminar esta maestría...

Por esto y todo lo que haya olvidado ya, este logro es tan tuyo como mío.

Gracias, "Pishosha".

ABSTRACT

This thesis addresses the critical challenges of data scarcity and class imbalance in automated visual inspection within industrial manufacturing, focusing on the synthesis of casting defects in metallic submersible pump impellers. To resolve this bottleneck, we propose and evaluate a generative architecture based on Latent Diffusion Models (LDMs)—specifically Stable Diffusion v1.5—adapted parameter-efficiently using Low-Rank Adaptation (LoRA) and strict geometric conditioning via ControlNet (utilizing Canny edge maps). This proposed pipeline is compared quantitatively and qualitatively against a custom baseline model, an Auxiliary Classifier Conditional Variational Autoencoder Generative Adversarial Network (AC-CVAE-GAN) trained from scratch. Quantitative results demonstrate the significant superiority of the diffusion-based model, which achieves a Fréchet Inception Distance (FID) of 67.65 for the nominal class and 96.17 for the defective class (outperforming the baseline's scores of 209.36 and 180.72, respectively). Perceptually, LPIPS metrics confirm enhanced structural variance and visual realism in the diffusion-generated samples. Qualitatively, the proposed architecture successfully synthesizes realistic high-frequency metallic textures and defect topologies (e.g., porosity), though minor limitations such as concept bleeding and spatial texture dislocation are identified. In conclusion, the concurrent integration of LoRA and ControlNet within Latent Diffusion Models establishes a robust and scalable framework for high-fidelity synthetic data augmentation, effectively mitigating the unavailability of physical defects on highly optimized production lines.

RESUMEN

Este trabajo de tesis aborda el problema de la escasez de datos y el desequilibrio de clases en la inspección visual automatizada de manufactura industrial, enfocándose en la síntesis de defectos de fundición en impulsores metálicos de bombas sumergibles. Para resolver esto, se propone y evalúa una arquitectura basada en Modelos de Difusión Latente (LDM), específicamente Stable Diffusion v1.5 adaptada de manera eficiente mediante matrices de Bajo Rango (LoRA) y condicionamiento geométrico estricto por medio de ControlNet (usando mapas de bordes Canny). Esta propuesta se compara cuantitativa y cualitativamente frente a un modelo base híbrido personalizado de Red Neuronal Convolucional Condicional Variacional Autoencoder y Red de Confrontación Adversaria (AC-CVAE-GAN) entrenado desde cero. Los resultados cuantitativos demuestran la superioridad del modelo de difusión, alcanzando una distancia de Fréchet (FID) de 67.65 para la clase nominal y 96.17 para la clase con defectos (en comparación con 209.36 y 180.72 del modelo base). Perceptualmente, la métrica LPIPS confirma una mayor diversidad y realismo estructural en las muestras generadas por la propuesta de difusión. Cualitativamente, Stable Diffusion logra sintetizar detalles de alta frecuencia y texturas metalúrgicas realistas (como porosidad), aunque se documentan desafíos inherentes como el concepto de sangrado (concept bleeding) y la dislocación espacial de fallas. En conclusión, la integración de LoRA y ControlNet en Stable Diffusion ofrece un marco de trabajo altamente robusto y escalable para la generación de datos sintéticos realistas, mitigando de manera efectiva la escasez de fallas físicas en líneas de producción optimizadas.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AC Auxiliary Classifier. 60, 64, 85, 86

ACGAN Auxiliary Classifier GAN. 46, 60, 81

ACWGAN Auxiliary Conditional Wasserstein GAN. 29

ADA Adaptive Discriminator Augmentation. 30

AE Autoencoder. 40

AFMS Attentive Fourier-Modulation Synthesis. 30

CFG Classifier-Free Guidance. 71

CLIP Contrastive Language-Image Pre-training. 19, 20, 51, 58, 68–70, 83, 86

CNN Convolutional Neural Network. 15, 17, 35, 36, 39, 49, 55

Con-GAN Contrastive Generative Adversarial Network. 18, 29

CVAE Conditional Variational Autoencoder. 17

CVAE-GAN Conditional Variational Autoencoder Generative Adversarial Network. 15, 18, 22–24, 46, 60, 64, 74, 81, 82, 85, 86

DAAM Diffusion Attentive Attribution Maps. 86

DCCVAE Deep Convolutional Conditional VAE. 29, 32

DCGAN Deep Convolutional GAN. 45, 46

DDPM Denoising Diffusion Probabilistic Model. 19, 20, 23, 27, 30–32, 48–50, 60, 70

DG Defect Generation. 29

DJS Jensen-Shannon Divergence. 29

DL Deep Learning. 17, 20–22, 35, 36, 59

ELBO Evidence Lower Bound. 41

FAM Feature Attention Matching. 29

FID Fréchet Inception Distance. 18, 23, 24, 54, 55, 74, 76, 81

GAN Generative Adversarial Network. 15, 17–19, 21–24, 27–29, 31, 32, 42, 45, 47

GP Gradient Penalty. 29

GPU Graphics Processing Unit. 20, 59, 69

Grad-CAM Gradient-weighted Class Activation Mapping. 86

HOG Histogram of Oriented Gradients. 17

IoT Internet of Things. 20, 22

KLD Kullback-Leibler Divergence. 42, 47, 65, 81

LDM Latent Diffusion Model. 15, 19, 20, 23, 24, 32, 50–53, 58, 68, 70, 76, 81, 83, 85

LoRA Low-Rank Adaptation. 15, 19, 20, 23, 24, 32, 33, 53, 60, 68–71, 76, 78, 82, 85, 86

LPIPS Learned Perceptual Image Patch Similarity. 18, 23, 24, 55, 74, 76, 81

MAE Mean Absolute Error. 65

ML Machine Learning. 35

MLP Multilayer Perceptron. 35, 45

MSE Mean Squared Error. 21, 42, 50, 54, 65, 71

PCB Printed Circuit Board. 29

PEFT Parameter-Efficient Fine-Tuning. 53, 60, 69, 82

ReLU Rectified Linear Unit. 37, 61, 63

ResNet Residual Network. 38, 39, 62

RGB Red, Green, Blue. 57, 58, 68, 70

SD Stable Diffusion. 15, 19, 20, 23, 24, 31, 32, 50, 51, 60, 68, 69, 74, 76, 78, 82, 86

SDA Shared Data Augmentation. 29

SIFT Scale-Invariant Feature Transform. 17

Tanh Hyperbolic Tangent. 58, 62

TF-IDG Training-Free Industrial Defect Generation. 31, 33

VAE Variational Autoencoder. 15, 17, 18, 21, 22, 24, 27, 28, 32, 40–42, 46, 47, 50, 51, 68–71, 81, 82, 85, 86

VLB Variational Lower Bound. 50

VRAM Video Random Access Memory. 69

XAI Explainable Artificial Intelligence. 86

1. INTRODUCTION

The generation of synthetic imagery via deep generative models presents a robust solution to the critical scarcity of defective sample datasets in industrial manufacturing. Statistical evidence suggests that manual visual inspection in manufacturing processes can suffer from missed defect rates between 20% and 30%, primarily due to high mental workload and the subtle nature of the defects [1], [2]. Furthermore, industrial datasets are typically highly imbalanced because manufacturing processes are optimized to minimize errors (with typical defect rates often <10%), making the training of automated systems statistically challenging due to the rarity of anomalous samples [1], [2], [3].

To address this, early automated solutions relied on *Classical Computer Vision* and heuristic-based feature extraction. While these were superseded by Convolutional Neural Network (CNN) with the AlexNet architecture [4], the latter introduced a critical dependency on large annotated datasets. Initial attempts to mitigate data scarcity utilized traditional data augmentation—or data warping—techniques; however, these lack the semantic diversity required to represent complex stochastic defects [5]. Consequently, historical generative approaches have relied on Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). While VAEs struggle with image sharpness [6], [7] and GANs frequently suffer from Mode Collapse [8], the Conditional Variational Autoencoder Generative Adversarial Network (CVAE-GAN) hybrid architecture emerged as a balanced solution [9].

This research aims to demonstrate, through a benchmark analysis, that a Stable Diffusion (SD) [10] approach—leveraging Latent Diffusion Models (LDMs) [11] enhanced with ControlNet [12] and Low-Rank Adaptation (LoRA) [13]—outperforms these earlier paradigms in capturing high-frequency textures like surface roughness and porosity on metal manufactured pieces.

To validate the proposed architectures, this research utilizes a specialized dataset comprising thousands of high-resolution images derived from the casting manufacturing process [14], a fundamental technique that involves the solidification of liquid material within a mold. Despite its industrial prevalence, this process is susceptible to stochastic irregularities, including blow holes, pinholes, burrs, and shrinkage defects.

The structure of this document is organized as follows:

- **Chapter 1** establishes the research context, justification, problem statement, hypothesis, and objectives.
- **Chapter 2** reviews the State of the Art, comparing recent proposals in synthetic defect generation.
- **Chapter 3** (Theoretical Framework) provides an in-depth analysis of the underlying methodology and tools.
- **Chapter 4** focuses on dataset acquisition and the architecture required for model training.

- **Chapter 5** presents the experimental execution, testing protocols, and results.
- **Chapter 6** finally discusses conclusions regarding the scientific and societal impact of this research and its potential transfer to real-world manufacturing environments.

1.1 Background

Quality assurance is a critical component in modern manufacturing, incorporating the rigorous detection of surface irregularities such as cracks, pores, or burrs [15]. Historically, early automated solutions relied on classical computer vision. These algorithms depended on hand-crafted feature extraction using descriptors such as Scale-Invariant Feature Transform (SIFT) or Histogram of Oriented Gradients (HOG), requiring engineers to manually define defect parameters [16], [17]. Consequently, they lacked robustness against environmental variations, such as changes in lighting, complex object geometries, or the stochastic nature of metallic textures [15].

The arrival of Deep Learning (DL) via CNN overcame these limitations by automatically learning hierarchical feature representations directly from raw pixel data. However, the efficacy of CNN-based classifiers is strictly constrained by the availability of massive and balanced datasets [1]. To address this data deficit, traditional Data Augmentation techniques—often referred to as "Data Warping"—are commonly employed. While effective for increasing dataset size, these methods (e.g., rotation, flipping, or scaling) produce geometric transformations of existing images but fail to generate semantically new data or capture the complex visual distribution of rare anomalies [5].

This challenge drove to the adoption of generative models, a field pioneered by VAEs and GANs. VAEs were fundamentally important because they introduced a probabilistic approach to generate completely new data from a latent space [6], while GANs revolutionized the field through an adversarial training framework—where a Generator creates synthetic data and a Discriminator evaluates its authenticity—that produced exceptionally sharp and realistic images [8].

As generative techniques evolved, standard VAEs established a strong foundation by successfully modeling the elemental probabilistic distribution of the data. However, their initial formulation was *unconditional*, inherently lacking categorical control over the generation process. Additionally, a well-documented characteristic of these early VAEs was the tendency to produce blurred images, an effect stemming from their reliance on element-wise measures—such as the squared reconstruction error—and the inherent injection of noise [9].

Building upon this probabilistic framework, Conditional Variational Autoencoder (CVAE) were introduced as a natural extension. By integrating explicit conditional inputs, such as class labels, directly into the generative process, CVAE gained the ability to steer the latent variables. This architectural evolution provided the desired categorical control, allowing the models to handle complex one-to-many mappings—situations where a single input condition can correspond to multiple valid defect shapes and locations—more effectively [18], [19].

Despite their conditional capabilities, likelihood-based models like CVAE still exhibit poor generalization when trained on highly imbalanced datasets, as their learned distributions

inevitably skew toward the majority class [1], [5]. To overcome this statistical bias and successfully synthesize unpredictable anomalies in complex industrial domains, recent advancements have shifted toward specialized *adversarial architectures*, such as advanced GANs, explicitly designed to operate under strictly limited data regimes [19].

Throughout the development of these adversarial architectures, a well-documented phenomenon was the tendency of models to converge toward producing only a single type of output, losing sample diversity—a state known as *mode collapse* [20]. To stabilize the learning process and ensure synthesis diversity, researchers introduced *feature matching* techniques, which train the generator to match the statistical features of real data at intermediate network layers rather than relying on the discriminator's final output only [20].

Leveraging the foundation of these stabilization strategies, hybrid architectures like the CVAE-GAN framework emerged as a natural progression, combining the probabilistic encoding of VAEs with the adversarial learning of GANs [9]. By explicitly mapping real images to a latent space and applying pairwise feature matching, these frameworks establish reference points that effectively prevent mode collapse while preserving the fine structural details of the generated samples [9].

Similarly, models like the Contrastive Generative Adversarial Network (Con-GAN) have been specifically designed for surface defect generation, proving capable of synthesizing high-quality and varied defect images even under extremely limited data conditions (e.g., as few as 10 samples) [19].

Although Con-GAN excels at generating diverse surface defects under extremely limited data regimes through shared data augmentation [19], CVAE-GAN provides a superior theoretical framework for highly structured industrial anomalies like metallic cracks or casting defects. By mapping real images to a continuous latent space via an encoder, CVAE-GAN enables the synthesis of semantically new samples through attribute interpolation, while its pairwise feature matching and pixel-level reconstruction loss explicitly preserve the delicate physical topologies and structural details of the defects [9].

Despite the robust capabilities of hybrid adversarial architectures like the CVAE-GAN [9], synthesizing highly complex defects with high-frequency texture details—such as fine porosity and surface roughness—remains challenging. In these scenarios, GANs are notoriously difficult to train, often suffering from inherent adversarial bottlenecks and residual mode collapse risks [19].

These generative architectures are quantitatively evaluated in terms of their ability to capture true data distributions and avoid mode collapse by using specialized metrics: the Fréchet Inception Distance (FID) which has become the standard to measure the statistical realism and distributional fidelity of synthetic images compared to real data [21], and the Learned Perceptual Image Patch Similarity (LPIPS) metric, that utilized to assess perceptual variance and intra-class diversity, serving as a rigorous mathematical indicator of mode collapse mitigation [22].

To address these limitations, a major paradigm shift emerged with Denoising Diffusion Probabilistic Model (DDPM). Unlike GANs, DDPMs are likelihood-based models inspired by *nonequilibrium thermodynamics*, relying on a mathematically stable process consisting of two Markov chains: a fixed forward *diffusion* process that gradually adds Gaussian noise to an image until the signal is destroyed, and a parameterized reverse process that learns to iteratively denoise it to recover a high-fidelity data sample [11].

The neural backbone of this reverse process is typically a U-Net architecture [11]. Structurally, the U-Net builds upon a fully convolutional network foundation—featuring a contracting path to capture context and a symmetric expanding path—and incorporates skip connections to combine high-resolution features, thereby preserving fine-grained spatial details and enabling precise localization during synthesis [23].

While highly effective, standard DDPMs operate directly in the high-dimensional pixel space, making both training and inference computationally prohibitive [10]. This limitation was overcome by LDMs, the foundational architecture behind large-scale implementations like SD [10], [12]. LDMs explicitly separate the compressive and generative learning phases by utilizing a pre-trained autoencoder: an encoder compresses the raw image into a lower-dimensional, perceptually equivalent latent space where the diffusion process occurs efficiently, and a decoder reconstructs the final image back to pixel space [10].

To guide this generative process semantically, these models integrate Contrastive Language-Image Pre-training (CLIP)[12], [24]. By jointly training an image and a text encoder to maximize *cosine similarity*, CLIP aligns text and visual representations in a shared multimodal embedding space [24]. This allows textual prompts to be encoded into latent vectors that directly steer the U-Net's denoising steps via cross-attention layers [10], [12], [25], enabling the targeted synthesis of specific defect categories.

However, applying foundational SD to complex surfaces—such as those found in casting manufacturing—requires more than just semantic text guidance; it demands precise structural control over the image composition. To achieve this, the ControlNet architecture was introduced to provide explicit spatial conditioning to large pretrained diffusion models [12]. ControlNet preserves the generative capabilities of the original model by locking its parameters while creating a trainable copy of its encoding layers. These components are connected via "zero convolutions"—1x1 convolution layers initialized with zeros—ensuring that no harmful noise disrupts the robust pretrained backbone during finetuning [12].

By utilizing extracted foundational features—such as structural boundaries obtained via *Canny Edge Detection* [12], [26]—as localized input conditions, this architecture ensures that synthetic defects are generated with accurate physical topologies and spatial geometries that precisely match the original metallic pieces.

Finally, to adapt this massive, general-purpose model to the highly specific texturing of casting manufacturing, LoRA is employed [13]. Instead of updating all model parameters,

LoRA freezes the pre-trained weights and injects trainable rank decomposition matrices directly into the Transformer's cross-attention layers [13]. This enables highly efficient fine-tuning, allowing the model to effectively capture the domain-specific details of metallic anomalies without the need for massive computational infrastructure [13].

This synergistic combination—latent denoising via DDPMs and U-Net [11], [23], semantic guidance through CLIP [24], precise geometric conditioning with ControlNet and Canny edges [12], [26], and efficient textural adaptation via LoRA [13]—positions SD as a highly robust, state-of-the-art paradigm for synthetic defect generation.

1.2 Justification

Waiting for physical defects to occur naturally in order to manually acquire and annotate thousands of samples is a highly inefficient process that disrupts continuous production lines [15]. Therefore, transitioning from manual data acquisition to advanced synthetic data generation represents a crucial optimization of manufacturing resources.

The proposed solution leverages the efficiency of LDMs combined with LoRA, potentially lowering the barrier to entry in order to make this transition accessible in terms of both computational and economic resources. Historically, training foundational generative models from scratch required massive, financially prohibitive data center infrastructures [10]. However, LoRA fundamentally shifts this paradigm by drastically reducing the computational overhead required for domain-specific adaptation [13].

This architectural efficiency enables the synthesis of high-fidelity defects using accessible, consumer-grade hardware—such as an off-the-shelf NVIDIA RTX 3090Ti Graphics Processing Unit (GPU)—reducing the need for expensive compute clusters [10], [12], [13]. When compared to the recurrent labor and operational costs associated with extensive manual inspection procedures [2], [19], this localized synthetic approach becomes a highly affordable and sustainable alternative for broad industrial applications.

Ultimately, this approach directly advances the core principles of smart manufacturing. A central pillar of these modern intelligent manufacturing systems is the integration of the Internet of Things (IoT) and DL to digitally model and optimize physical processes [19].

By synthesizing a comprehensive digital distribution of manufacturing anomalies, this work shifts the paradigm from reactive data collection—waiting for rare physical failures to occur naturally to learn from them [15], [19]—to proactive synthetic generation. This integration enables systems to, *a priori*, train robust computer vision classifiers on expanded datasets [19], strengthening the role of digital transformation in solving inherent production challenges and achieving higher levels of industrial automation [15].

1.3 Problem

When developing a computer vision system to detect defects in manufactured parts, engineers face the necessity of performing data augmentation on their training data due to the extreme scarcity of balanced training data [5], [15], [19]. Although engineers might have access to a company's internal data, the defect class in datasets is typically highly imbalanced [1], [15]. However, manual intervention to increase data variability is highly complex and expensive [1], [15], [19].

This scarcity of real defect image samples represents a critical bottleneck in the development of automated defect detection systems [15], [19]. This lack of data is an inherent byproduct of modern manufacturing processes; because production lines are highly optimized to minimize physical errors, true structural defects manifest as rare statistical anomalies [1]. For DL classifiers, this extreme class imbalance leads to biased training, causing models to overfit to the defect-free majority class and ultimately fail to reliably identify critical surface flaws [5], [15], [19].

It is widely recognized in the literature that generating a labeled dataset is a highly expensive and time-consuming task [5], [15], [19]. To mitigate this issue, recent research has proposed various data augmentation strategies [5], [15], [19]. These approaches range from classical data warping—which artificially inflates datasets through geometric transformations such as rotating, flipping, and cropping [5], [19]—to "copy-paste" techniques that directly overlap defect samples onto defect-free images to balance the dataset [27].

Although these traditional data augmentation techniques exist, they often result in performance saturation regarding model generalization [5], [19]. Relying on simple geometric transformations fails to improve the underlying data distribution determined by higher-level features, ultimately falling short of capturing the complex visual variations of real surface anomalies [5], [19].

These transformations are effective in teaching the network translational invariances, ensuring the model recognizes a defect regardless of its position or rotation [5]. However, these techniques do not create semantically new synthetic instances [19]. In complex domains, bridging the gap between the training and test sets requires overcoming biases that are significantly more complex than simple translational and positional variances [5].

Regarding generative AI solutions for synthetic data generation in this domain, VAEs and GANs are two prominent options; however, both present significant limitations [9], [28]. While VAEs provide a probabilistic framework for generation, their inherent injection of noise and reliance on imperfect element-wise measures—such as the Mean Squared Error (MSE) loss—prevent them from accurately capturing high-frequency details, inevitably resulting in blurred image generation [7], [9].

On the other hand, standard GANs suffer from severe convergence instability and a well-documented phenomenon known as mode collapse [9], [19], [20], [28]. These adversarial

models are notoriously hard to train because early generated samples often differ significantly from real data [9], [20]. If the discriminator perfectly separates real from fake images too quickly during this early stage, it leads to an unstable training process where gradient descent suffers from vanishing gradients and fails to provide significant updates to the generator [8], [9].

Furthermore, even if training progresses, the generator frequently suffer from mode collapse, a state where it produces images restricted to a limited area of the real data distribution, thereby losing the ability to synthesize the diverse samples required for effective data augmentation [9], [19], [20].

To address these isolated vulnerabilities, hybrid architectures like the CVAE-GAN were initially proposed to integrate the probabilistic mapping of VAEs with the high-resolution generation of GANs [9]. However, while these models mitigate basic mode collapse and offer conditional control [9], they ultimately fail to overcome the inherent bottlenecks of adversarial training when applied to rigorous industrial demands [15], [19]. Specifically, synthesizing highly complex, high-frequency texture details—such as fine porosity and surface roughness—exposes residual adversarial instabilities [19]. As a result, the generator still struggles to accurately fit the true statistical distribution of complex metallic anomalies [15], [19].

Recent research concludes that the lack of high-fidelity samples and balanced datasets directly impacts critical metrics such as the *F1-score* [19]. Without adequate and high-quality data augmentation, classification performance decreases, translating into an unacceptable increase in false negatives (undetected defects) or false positives in production environments [2], [15], [19].

Additionally, industrial defects possess high-frequency texture details [19]. If a standard GAN generates low-quality images and suffers from Mode Collapse, the neural network fails to learn the true defect distribution [9], [19], [20]. This behavior is often caused by an overfitted discriminator that becomes unable to recognize subtle or irregular defects on real production lines [15], [19].

The generation of synthetic data is no longer simply an auxiliary tool, but an essential component of modern intelligent automation. Powered by the IoT, DL has become a developing trend in smart manufacturing and surface defect detection [15], [19]; however, overcoming dataset limitations is a critical prerequisite for deploying these methods in industrial environments [3], [19]. While classic data augmentation methods—such as Data Warping—reach a saturation point where adding more transformed data produces marginal returns in accuracy [5], [19], deep generative models successfully break this barrier by synthesizing highly diverse and realistic defect samples [19], [27].

Without the injection of high-quality synthetic data, industrial AI models hit an unacceptable performance constraint, risking the release of defective products to the market [2], [15], [19]. Consequently, overcoming this barrier requires abandoning unstable adversarial networks—which inherently suffer from convergence issues and mode collapse [9],

[19], [20]—in favor of mathematically stable paradigms.

DDPMs offer this stability through a rigorous likelihood-based framework [11]. Specifically, LDMs like SD, which perform the generative process within a compressed latent space reducing the computational costs, transition from theoretical research concepts to becoming the critical infrastructure for quality assurance in modern manufacturing [10].

To clearly define the scope within this context, this work focuses exclusively on the design, training, and evaluation of a Latent Diffusion generative architecture—leveraging SD conditioned with ControlNet and optimized via LoRA—to synthesize high-fidelity defect data and entirely mitigate the adversarial mode collapse inherent to standard GANs. Consequently, the compilation of a final, production-ready balanced dataset, as well as the subsequent training and deployment of an operational computer vision classifier for real-time defect detection, remains entirely outside the boundaries of this project.

1.4 Hypothesis

The implementation of a LDM, specifically SD, geometrically conditioned via ControlNet and texturally optimized through LoRA, enables a highly precise modeling of the statistical distribution of industrial defects on casting manufactured pieces. By relying on a mathematically stable denoising process, this approach outperforms hybrid adversarial architectures, such as the CVAE-GAN, by entirely avoiding inherent training instabilities and Mode Collapse, thereby successfully capturing complex, high-frequency textural details.

Furthermore, it addresses the critical class imbalance caused by the high refinement of modern manufacturing processes—which are strictly optimized to minimize physical errors—by providing a reliable and computationally efficient method to synthetically populate the "defect" class, mitigating the necessity to manually gather, preprocess, and annotate extensive real-world samples.

The success of this generative architecture will be quantitatively validated against a specific baseline: a custom, hybrid adversarial architecture CVAE-GAN trained under identical constraints. Operationally, success is defined by achieving a FID score strictly lower than the baseline—proving superior visual and distributional fidelity compared to real defect samples—and an LPIPS score significantly lower than the baseline, confirming robust intra-class diversity and the successful mitigation of Mode Collapse.

1.5 Objectives

1.5.1 General Objective

To design, implement, and fine-tune a pre-trained LDM—specifically SD enhanced with ControlNet and LoRA—for the generation of high-fidelity synthetic images of defective casting pieces, and to comparatively evaluate its performance against a custom designed and trained CVAE-GAN baseline model. Furthermore, to demonstrate that the diffusion-based architecture effectively mitigates mode collapse and provides a superior, mathematically stable solution to the severe class imbalance inherent to industrial datasets.

1.5.2 Specific Objectives

1. To acquire and preprocess an adequate industrial dataset of casting manufactured pieces, ensuring the standardization of images and the extraction of Canny edge maps required for the geometric-conditioning approach of ControlNet.
2. To design and train a custom class-conditioned CVAE-GAN architecture to serve as the formal adversarial baseline for comparative evaluation.
3. To implement and fine-tune a pre-trained LDM—specifically SD enhanced with ControlNet and LoRA—focusing on precise geometric conditioning and high-frequency textural adaptation.
4. To quantitatively evaluate the generative performance of both architectures by measuring visual realism and intra-class diversity using FID and LPIPS metrics.
5. To perform a comparative benchmark between the diffusion-based model and the adversarial baseline, demonstrating that the LDM effectively mitigates Mode Collapse and provides superior statistical modeling of industrial defects.

1.6 Scientific and Technological Contribution

Nowadays, while base generative models like VAEs and GANs are widely implemented, they frequently suffer from inherent limitations, such as severe mode collapse. This research contributes to the empirical demonstration that LDMs—specifically SD integrated with ControlNet and LoRA—provide a mathematically stable alternative that mitigates these issues. Furthermore, it proves that this diffusion-based approach excels in the high-frequency texture domain, accurately synthesizing the complex morphology of industrial casting defects.

Additionally, this work provides a validated methodology to transition from traditional geometric Data Warping to the generation of semantically new synthetic data. In doing so, this research establishes a scalable framework for targeted defect synthesis, effectively addressing the severe class imbalance inherent to industrial datasets.

Ultimately, by generating high-fidelity synthetic data, this approach facilitates the development of robust computer vision models for quality inspection. It allows the industry to overcome the critical data scarcity caused by strictly optimized manufacturing lines, providing the necessary volume of "defect" samples without the operational costs of extensive manual data collection.

2. STATE OF ART OR THE TECHNIQUE

The progressive evolution of deep generative models has fundamentally changed how researchers approach the synthetic data generation in complex visual domains, such as those found in industrial manufacturing. This chapter presents a critical review of the recent scientific literature, focusing on the advancement of generative architectures that have defined the state of the art in this field in recent years.

This chapter reviews current works that have proposed solutions to the problem of synthetic image generation across three main paradigms: adversarial approaches and hybrid architectures (specifically VAEs and GANs), the paradigm shift toward DDPMs, and the integration of explicit geometric conditioning and efficient textural adaptation techniques.

Furthermore, this chapter formally defines the research gap that motivates the proposed architecture based on diffusion models, and establishes the theoretical and empirical basis for the subsequent design, implementation, and evaluation of the proposed solution.

To provide a structured overview of the literature reviewed in this chapter, Table 2.1 summarizes each of the generative models discussed in the following sections, detailing their main methodology and publication year.

Table 2.1: Summary of state-of-the-art generative models in industrial defect augmentation.

Work / Paper	**Brief Summary**	**Year**
Defect-GAN [29]	A compositional layer-based architecture simulating defacement and restoration on normal surfaces to generate anomalies.	2021
Defect Transfer GAN [27]	An adversarial transfer framework copying learned defect characteristics onto healthy background textures.	2023
Con-GAN [19]	A contrastive GAN framework using hypersphere-based contrastive loss for few-shot metal defect synthesis.	2023
DG2GAN [30]	A CycleGAN-based architecture using Jensen-Shannon optimized discriminator to translate healthy images to defective ones.	2024
DCCVAE-ACWGAN-GP [31]	A hybrid DCCVAE and ACWGAN-GP model for controlled multi-category synthesis of ship coating defects.	2024
StyleGAN2-AFMS [32]	A frequency-domain dual-stream adversarial model modulating weld defect features via Amplitude-Frequency Modulation Synthesis.	2026
DDPM Baseline [33]	A systematic evaluation of standard DDPMs for in-distribution augmentation, showing robust anomaly capture.	2024
AnomalyDiffusion [34]	A diffusion model decoupling defect appearance from spatial location using spatial anomaly embeddings.	2024

Table 2.1 continued from previous page

Work / Paper	Brief Summary	Year
TF-IDG [35]	A training-free one-shot diffusion framework aligning features via Adaptive Anomaly Mask on normal backgrounds.	2025
GroundingAnomaly [36]	A diffusion-based approach incorporating per-pixel semantic maps for fine-grained spatial control.	2026
AnomalyControl [37]	A few-shot inpainting framework using Stable Diffusion and ControlNet guided by binary anomaly masks.	2024
stableIDG [38]	A Stable Diffusion adaptation using LoRA fine-tuning, personalized defect identifiers, and mask-guided learning.	2025

2.1 Adversarial Approaches and Hybrid Architectures (VAEs and GANs)

In recent years, literature has demonstrated that to address the extreme scarcity of balanced datasets in industrial environments—particularly the "defect" class in metallic parts, castings and manufactured surfaces—GANs and VAEs have evolved from general-purpose generative models into highly specialized domain architectures [15], [29].

2.1.1 Defect-GAN for Automated Inspection

Addressing the lack of anomalous samples in automated inspection, Zhang et al. [29] proposed *Defect-GAN*. This architecture pioneered the automated generation of synthetic defect images by mimicking the defacement and restoration processes on normal surface images. It employs a compositional layer-based architecture that provides flexible control over the categorical type and spatial location of the generated defects. By introducing stochastic variations into the defacement process, Defect-GAN successfully generates realistic samples across various image backgrounds, directly addressing the critical bottleneck of data scarcity in industrial defect detection [29].

2.1.2 Defect Transfer GAN for Data Augmentation

Generating entirely complex industrial backgrounds from scratch is a highly challenging task. To overcome this, Wang et al. [27] introduced the *Defect Transfer GAN*. This architecture focuses exclusively on diverse defect synthesis by extracting learned defect features and transferring them onto healthy—defect-free—images. This adversarial-based

"copy-paste" approach ensures that the newly synthesized samples maintain the highly optimized background of the textures of real manufactured parts while artificially populating the minority class with realistic defect variations, thus effectively balancing the dataset and improving the performance of downstream classifiers [27].

2.1.3 Con-GAN for Steel and Printed Circuit Board (PCB) surfaces

Focusing explicitly on metallic components and addressing extreme few-shot scenarios, Du et al.[19] developed Con-GAN. Since standard GANs struggle to synthesize the high-frequency texture details of metallic defects, Con-GAN uses a Shared Data Augmentation (SDA) module and Feature Attention Matching (FAM). Steel surfaces datasets and PCBs, the model learns to synthesize diverse anomalous samples using as few as 10 real defect images. The incorporation of a hypersphere-based contrastive loss prevents model collapse, and successfully tackles the data imbalance gap in critical metal manufacturing [19].

2.1.4 DG2GAN for Manufacturing Datasets

To address the severe imbalance between defective and non-defective data in practical production lines, Deng et al. [30] proposed *DG2GAN*. This architecture leverages cycle consistency loss to directly transform abundant defect-free manufacturing images into defective samples. To guarantee that the generated images exhibit sufficient diversity and high-quality textures, the authors integrated a Jensen-Shannon Divergence (DJS) optimized discriminator and a novel Defect Generation (DG)—DG2—adversarial loss. Evaluated on prominent industrial datasets like MVTec, this model successfully populates the defect class, raising downstream recognition precision significantly by artificially correcting the skewed class distribution [30].

2.1.5 DCCVAE-ACWGAN-GP for Ship Coating Defects

Addressing the severe data imbalance in the shipbuilding industry, Bu et al. [31] developed a hybrid architecture—*DCCVAE-ACWGAN-GP* to synthesize diverse coating defect images—cracking, blistering, and sagging—on metallic ship surfaces. By combining the probabilistic latent space modeling of a Deep Convolutional Conditional VAE (DCCVAE) with the adversarial learning mechanism of an Auxiliary Conditional Wasserstein GAN (ACWGAN) with Gradient Penalty (GP), this method generates controllable, multi-category defect datasets. This probabilistic-adversarial approach circumvents the mode collapse issues inherent to standard unsupervised GANs, effectively populating the defect class to train robust classifiers.

2.1.6 StyleGAN2-AFMS for Laser Welds

Pushing adversarial data augmentation into highly complex domains—such as precision electronics—, Ma et al. [32] addressed the severe shortage of class-balanced training datasets in sinusoidal wobble laser welds. They proposed *StyleGAN2-AFMS*, a novel generative model that handles the coexistence of strong periodic domain textures and local non-periodic defects in metallic joints. By employing a frequency-domain dual-stream mechanism—reconstructing global periodic structures via inverse Fourier transform while modulating non-periodic defect features (Attentive Fourier-Modulation Synthesis (AFMS))—the network successfully synthesizes high-fidelity critical metallic anomalies like fractures, and excessive penetration. Coupled with Adaptive Discriminator Augmentation (ADA), this model successfully mitigates the overfitting in few-shot metallic defect detection [32].

2.2 The Paradigm Shift Toward DDPMs

2.2.1 DDPMs for In-Distribution Augmentation

To establish as baseline for diffusion-based data augmentation in industrial domains, Capogrosso et al. [33] systematically evaluated standard DDPMs for generating surface defects. This work demonstrated that, compared with traditional techniques, diffusion models excel at generating in-distribution augmented images that capture the subtle visual features of industrial anomalies. By producing samples that accurately reflect the true distribution of the minority class, this approach effectively balances the datasets and significantly improves the performance of downstream defect detectors [33].

2.2.2 AnomalyDiffusion for Decoupled Anomaly Synthesis

To overcome the fact that generative models often destroy highly optimized, complex background textures of manufactured pieces when synthesizing defects, Hu et al. [34] proposed *AnomalyDiffusion*. This DDPM-based framework decouples the appearance of the anomaly from its spatial location. By utilizing spatial anomaly embeddings, the model learns the high-frequency stochastic details of industrial defects independently from the background. This allows the network to iteratively denoise and inject highly realistic anomalies into healthy parts, avoiding the corruption of the surrounding structural integrity, thereby effectively populating the defect class [34].

2.2.3 TF-IDG for Training-Free Defect Generation

Addressing the huge computational costs and data requirements of training standard diffusion models from zero, Xu et al. [35] presented the Training-Free Industrial Defect Generation (TF-IDG). This method tackles the extreme "one-shot" scenario—where only a single real anomaly sample is available per metallic defect type—by leveraging a feature alignment strategy and an Adaptive Anomaly Mask directly within a pre-trained diffusion space, TF-IDG extracts fine-grained characteristics of the single reference defect and seamlessly integrates them into the normal background images. This approach successfully bypasses the massive computational overhead due to retraining of DDPMs while continuously populating the defect class with high-fidelity, diverse synthetic industrial anomalies [35].

2.2.4 GroundingAnomaly for Spatial Control

Following the need for precise defect population in industrial datasets, Liu et al. [36] proposed *GroundingAnomaly*. While standard DDPMs offer better mode coverage over GANs, they lack fine-grained spatial control. To overcome this, the researchers integrated *per-pixel* semantic maps into the diffusion reverse process. This explicit mask-guided framework ensures that the generated anomalies are precisely aligned with predefined locations on the manufactured parts. By transforming readily available defect-free metallic and industrial images into accurately labeled defective samples, GroundingAnomaly provides a mathematically stable solution to artificially balance industrial datasets for downstream classification and segmentation [36].

2.3 Explicit Geometric Conditioning and Efficient Textural Adaptation

2.3.1 AnomalyControl via Inpainting and ControlNet

To address the difficulty of steering large general-purpose models to produce realistic defects for specific industrial products, *AnomalyControl* was proposed by Ali et al. [37] as a few-shot generation framework. Understanding that standard generation alters the entire image, this method casts defect generation strictly as an inpainting problem over nominal—defect-free—images. By deploying a pre-trained SD inpainting backbone alongside ControlNet, it ingests binary anomaly masks taken from real anomalous samples, ensuring that the generated defects perfectly adhere to the required spatial geometries without destroying the surrounding healthy textures. This approach was validated on the MVTec AD dataset, where it efficiently populates the defect class and significantly

improves the performance of downstream classifiers, demonstrating the critical role of explicit geometric conditioning in industrial defect synthesis [37].

2.3.2 stableIDG for Steel Surface Defects

Targeting the long-tailed distribution inherent to industrial manufacturing, Li et al. [38] developed *stableIDG*, a novel architecture specifically designed for generating complex metal surface defects—such as Inclusions and Foreign Object Embeddings on medium and heavy plates—. To achieve efficient textural adaptation without massive computational overhead, the model fine-tunes SD using LoRA embedding matrices. In addition, it incorporates personalized defect identifiers and a mask-guided learning mechanism. This geometric constraint forces the network's attention toward the critical defect regions during the denoising process, preventing malformation and synthesizing high-fidelity metallic defects that effectively rebalance data-scarce industrial datasets [38].

2.4 Research Gap

To address the limitations of standalone adversarial networks, recent literature has heavily explored hybrid architectures. For instance, advanced conditional variants, such as the DCCVAE combined with Wasserstein GANs, have been deployed to generate high-fidelity surface defects on metallic coatings, explicitly aiming to resolve model overfitting caused by severely unbalanced datasets [31]. While these VAE-GAN hybrids successfully mitigate basic instability by establishing reference points in a continuous latent space, they inherently rely on adversarial training. Consequently, they still face significant bottlenecks—such as mode collapse, convergence instability, and a lack of diversity—when strictly required to model the stochastic, high-frequency textures of complex manufacturing defects [19], [30].

To entirely mitigate these adversarial limitations, the field shifted toward DDPMs, which offer a mathematically stable, likelihood-based framework free from mode collapse. However, standard pixel-space DDPMs incur immense computational costs and slow inference times when scaling to high-resolution industrial images [35]. Consequently, LDMs—most notably SD—emerge as the superior paradigm for industrial data augmentation. By performing the iterative denoising process within a compressed, lower-dimensional latent space, Stable Diffusion drastically reduces computational requirement while preserving the ability to synthesize ultra-high-fidelity, high-frequency stochastic textures.

Despite the clear generative superiority of LDMs, adapting them to highly specific industrial domains—such as casting— requires strict, simultaneous control over both texture synthesis and geometric placement. Recent works have only tackled these conditioning requirements in isolation. On one hand, methods like stableIDG successfully utilize LoRA

for efficient textural fine-tuning [38]. However, because they rely on modifying the U-Net input channels to ingest masks rather than employing robust structural controllers, they exhibit limited control over the exact size, location, and quantity of the generated defects, occasionally failing to strictly respect the predefined geometry [38]. On the other hand, frameworks such as TF-IDG and AnomalyControl leverage ControlNet for precise spatial alignment, but they depend on training-free feature extraction models or standard inpainting techniques rather than direct, parameter-efficient weight fine-tuning [35], [37].

To the best of our knowledge, no existing approach in the literature seamlessly integrates Stable Diffusion concurrently with both LoRA and ControlNet for industrial defect generation for dataset augmentation. This unexplored combination presents a critical research gap. Uniting the parameter-efficient textural adaptation of LoRA with the rigorous, decoupled geometric conditioning of ControlNet could provide a scalable solution. Such an integrated framework would effectively populate the defect class with unparalleled spatial precision, strict adherence to the manufactured part's geometry, and high-fidelity metallic textures, entirely overcoming the data scarcity bottleneck in long-tailed precision manufacturing datasets.

3. THEORETICAL / CONCEPTUAL FRAMEWORK

While traditional Machine Learning (ML) depends on manual feature extraction, DL has revolutionized the computer vision field by learning autonomous hierarchical representations directly from raw data. Inside DL, the discriminative models have been standardized for classification tasks. However, for the objectives of this research—data synthesis—, it is necessary to get into the Deep Generative Models branch. But, it is imperative to first introduce the foundational architecture that is the backbone of these generative models: the CNNs

3.1 Deep Learning in Computer Vision

3.1.1 Convolutional Neural Networks CNNs

CNNs established the foundational architecture for modern generative models because they incorporate strong, inherently correct assumptions about the nature of visual data—such as images—, specifically the stationarity of visual statistics and the locality of pixel dependencies [4]. By utilizing an architecture based on 2D convolutions, CNNs achieve an immense learning capacity that can be easily controlled by varying their depth and breadth [4].

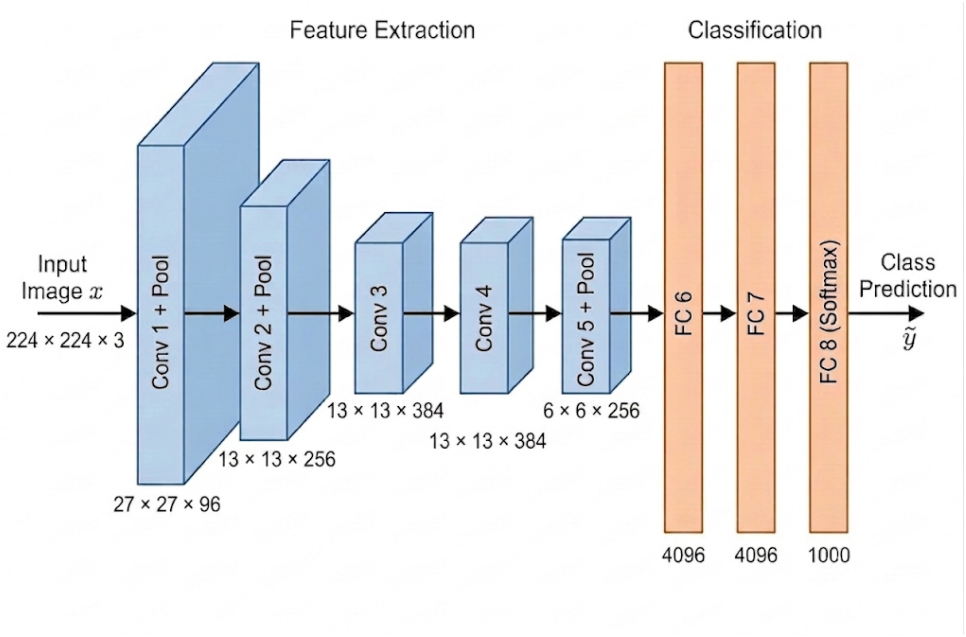


Figure 3.1: AlexNet CNN Generalized Architecture (generated using Nano Banana). Adapted from [4]

Unlike standard feedforward networks—such as Multilayer Perceptron (MLP)—, this localized connectivity requires significantly fewer parameters and connections, making them

more efficient and easier to train on high-resolution images [4]. Furthermore, the inherent, computationally efficient capability to understand and process complex spatial hierarchies—or topologies—solidified CNNs as the critical backbone for advanced visual synthesis

The Convolution Operation

In the context of DL applied to 2-dimensional signals, the convolution operation applies a set of trainable filters—commonly called kernels—over the input image. Each kernel is a small matrix of weights that slides across the input image, calculating the dot product between the kernel and the current region of the input image it is covering. This produces a feature map that captures visual patterns—edges, textures, and other complex structures—at different spatial locations [4].

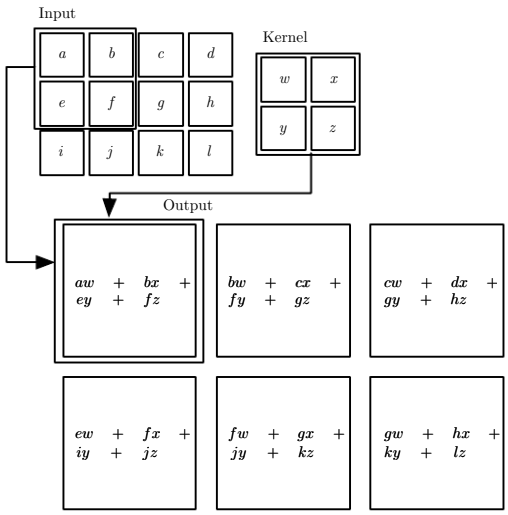


Figure 3.2: Convolution operation. Obtained from [39]

Goodfellow et al. [39] state that, given a two-dimensional input image I and a two-dimensional kernel K of size $m \times n$, the value of the resulting feature map S at spatial position (i, j) is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{3.1}$$

Where:

- I is the matrix of pixel intensities from the input image or previous layer.

- K is the weight matrix of the kernel, designed to detect specific spatial frequency patterns.
- The traversal of the kernel across the input image I is controlled by a stride and padding parameter that provides local translation invariance to the extracted feature map.

Non-Linear Activation

To model complex high-dimensional representations in addition to simple affine transformations, the resulting feature map S is passed through a non-linear activation function. The standard choice for convolutional layers is the Rectified Linear Unit (ReLU) activation function, mathematically defined as [4], [39]:

$$f(x) = \max(0, x) \tag{3.2}$$

This activation function introduces mathematical sparsity into the network by clamping all negative values to zero. This characteristic is critical since its derivative is exactly 1 for all positive inputs, effectively mitigating the vanishing gradient problem that deep architectures face during backpropagation [39].

Spatial Subsampling (Pooling)

To decrease the computational cost and progressively expand the receptive field of deeper kernels, a spatial subsampling operation—known as pooling—is applied. Max pooling is the most commonly chosen technique, since it captures the maximum value within a predefined sliding window—similar to the convolution operation—typically of size 2×2. This effectively downsamples the feature map's dimensions [4], [39]:

$$f_{pool}(R) = \max_{i,j \in R} x_{i,j} \tag{3.3}$$

Where:

- R is the region of the feature map covered by the pooling window.
- $x_{i,j}$ are the values within that region.

3.1.2 GoogLeNet and the Inception Module

To extract multi-scale topological features without incurring prohibitive computational costs, the Inception architecture utilizes parallel processing paths. Instead of a single filter, the module simultaneously computes convolutions using 1×1, 3×3, and 5×5 kernels, subsequently concatenating their output tensors. Crucially, it employs 1×1 convolutions as dimensionality reduction bottlenecks prior to the larger spatial operations. This mathematical configuration allows the network to capture both fine-grained anomalies and global structural defects within a single layer efficiently [40] (See Figure 3.3).

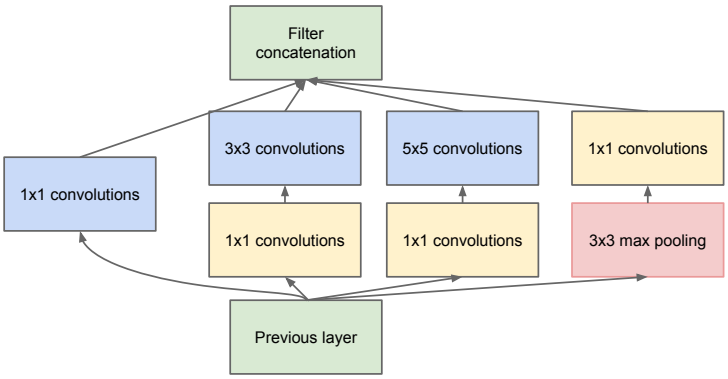


Figure 3.3: Inception Module. Obtained from [40]

3.1.3 Residual Network (ResNet)

As generative architectures increase in depth to model complex distributions, they inherently suffer from optimization difficulties, most notably the degradation problem, during training [41]. To guarantee stable optimization, the ResNet framework introduces identity shortcut connections [41]. Instead of expecting stacked non-linear layers to fit a direct mapping, it forces them to fit a residual mapping, formalized as [41]:

$$y = \mathcal{F}(x, \mathcal{W}_i) + x \tag{3.4}$$

This concept can be visualized in Figure 3.4, where the input x is directly added to the output of the convolutional block.

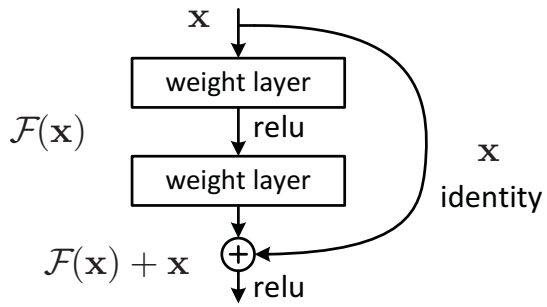


Figure 3.4: Residual Learning Block. Obtained from [41]

3.1.4 U-Net Architecture

While standard CNNs excel at feature extraction, they often struggle to preserve the spatial information critical for pixel-perfect reconstruction tasks. The U-Net architecture achieves this through a symmetrical encoder-decoder structure linked by spatial skip connections [23].

Unlike ResNet's element-wise addition, U-Net explicitly concatenates high-resolution feature maps from the encoder to the corresponding decoder layers [23]. This matrix concatenation reintroduces the exact spatial details lost during the downsampling process—latent compression—enabling the decoder to keep the strictly necessary high-frequency details to reconstruct sharp geometric structures.

Mathematically, the input to a decoder layer l is defined as:

$$X_{dec}^{(l)} = \text{Concat}\left(\mathcal{U}(X_{dec}^{(l-1)}), X_{enc}^{(L-l)}\right) \tag{3.5}$$

where:

- $\mathcal{U}(\cdot)$ denotes the spatial upsampling operation (such as a transposed convolution).
- $X_{enc}^{(L-l)}$ represents the symmetrically corresponding high-resolution feature map from the encoder.

By passing this high-frequency topological data directly across the network bottleneck, the architecture preserves fine-grained spatial coordinates throughout the generative process. Figure 3.5 illustrates the U-Net topology, demonstrating how the encoder and decoder are linked to achieve precise localization.

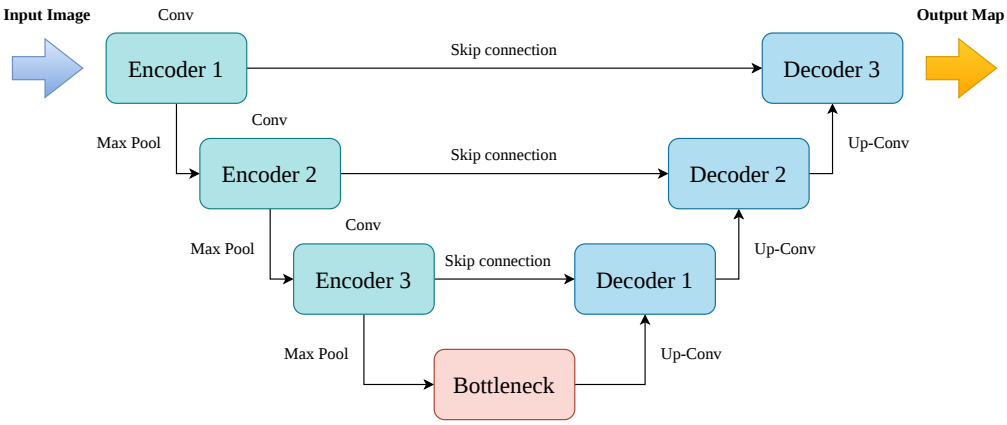


Figure 3.5: U-Net Basic Architecture. Adapted from [23].

3.2 Deep Generative Models

3.2.1 Variational Autoencoders (VAEs)

Standard deterministic Autoencoders (AEs) are designed to compress high-dimensional data into lower-dimensional latent vectors—bottleneck-shaped architectures—through an encoder, and subsequently reconstruct the original data via a decoder (see Figure 3.6). While effective for data compression, these models inherently memorize the training manifold as discrete points in the latent space. This results in a meaningless, non-continuous interpolation between samples, making them unsuitable for generating high-fidelity images [39].

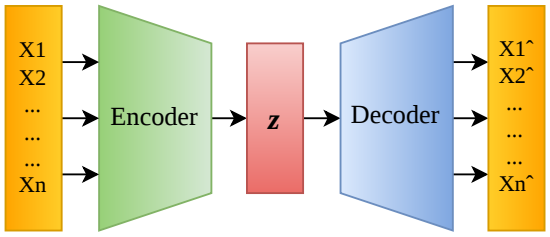


Figure 3.6: Autoencoder Architecture. Adapted from [42]

This limitation is addressed by VAEs, introduced by Kingma and Welling [6]. By defining a probabilistic generative process rather than a deterministic mapping, the VAE network, parameterized by ϕ, maps the high-dimensional image x to a localized probability distribution within the continuous latent space z. Instead of resulting in a strict coordinate

vector, the encoder approximation $q_\phi(z|x)$ produces two different vectors: a mean vector μ and a variance vector σ^2. Together, these vectors define a multivariate Gaussian distribution of the latent representation of the input image x [6]:

$$q_\phi(z|x) = \mathcal{N}(z; \mu, \sigma^2 I) \tag{3.6}$$

The Reparameterization Trick

To enable backpropagation through the stochastic sampling process, this Gaussian distribution must be differentiable with respect to the parameters μ and σ^2. To achieve this, the VAE architecture employs the reparameterization trick [6]. A random noise variable ϵ is sampled from a standard normal distribution $\mathcal{N}(0, I)$. The latent representation z is computed as a deterministic function as follows [6], [39]:

$$z = \mu + \sigma \odot \epsilon \tag{3.7}$$

where $\odot$ denotes the element-wise product. This formulation isolates the stochasticity in ϵ, preserving the differentiability required for gradient-based optimization of μ and σ^2 during training [39] (see Figure 3.7).

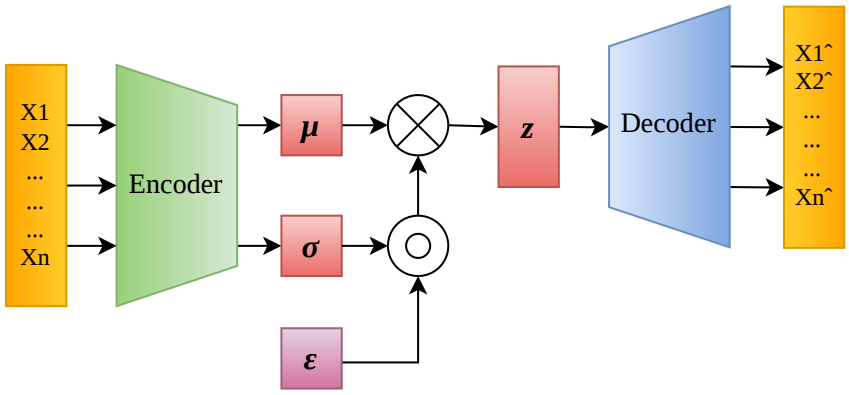


Figure 3.7: Variational Autoencoder Architecture. Adapted from [43]

The Objective Function

The optimization objective of the VAE cannot be solved analytically. Therefore, the model is trained by minimizing the Evidence Lower Bound (ELBO), which combines a spatial reconstruction loss—measuring the fidelity of the generated image compared to the

original input—with a statistical regularization term—the Kullback-Leibler Divergence (KLD)—that encourages the learned latent distribution to approximate a prior standard normal distribution [6]:

$$\mathcal{L}_{VAE}(\theta, \phi; x) = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + D_{KL}(q_\phi(z|x) \parallel p(z)) \tag{3.8}$$

Where:

- The first term is the expected negative log-likelihood of the data that evaluates the reconstruction fidelity of the generated image $p_\theta(x|z)$ compared to the original input x [6]. This term ensures that the generated morphology accurately matches the input sample constraint—typically measured by a pixel-wise loss such as MSE.
- The second term utilizes the KLD to force the learned latent distribution towards a standard normal distribution $p(z) = \mathcal{N}(0, I)$ [6]. This statistical constraint ensures a continuous latent space that is strictly necessary for generating coherent structural interpolations by sampling coordinates within the existing data distribution [39].

3.2.2 Generative Adversarial Networks GANs

While VAEs optimize an explicit lower-bound on the data likelihood, GANs, introduced by Goodfellow et al. [44], address the generative problem through implicit density estimation. The architecture avoids Markov chains and unrolled approximate inference networks, formulating the training process as a zero-sum game between two parameterized neural networks: a Generator G and a Discriminator D (see Figure 3.8).

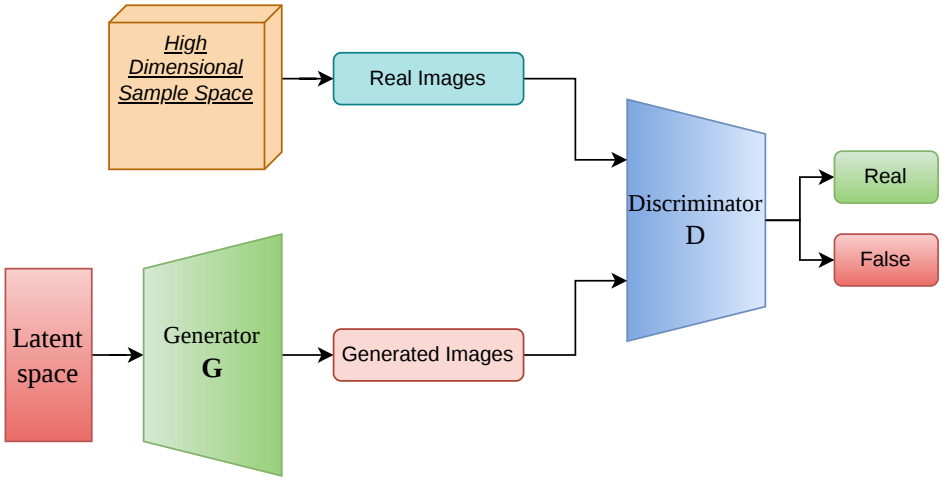


Figure 3.8: Generative Adversarial Network Architecture. Adapted from [44]

The Adversarial Framework

Here, $p_{data}(x)$ represents the true data distribution over the real spatial data x. The Generator $G(z;\theta_g)$, parameterized by θ_g, is tasked with mapping a latent noise vector z, sampled from a simple prior distribution $p_z(z)$—such as a standard normal distribution—to the high-dimensional data space, producing synthetic samples $G(z)$.

In contrast, the Discriminator $D(x;\theta_d)$, parameterized by θ_d, acts as a binary classifier that outputs a single scalar $D(x) \in [1]$. This represents the probability that the evaluated input x came from the real empirical data distribution p_{data} rather than the synthetic distribution p_g produced by the Generator [44].

To understand the training dynamics between the Generator and the Discriminator, it is helpful to visualize how their respective probability distributions evolve over time. As illustrated in Figure 3.9, this adversarial process aims to drive the model toward a Nash equilibrium [44].

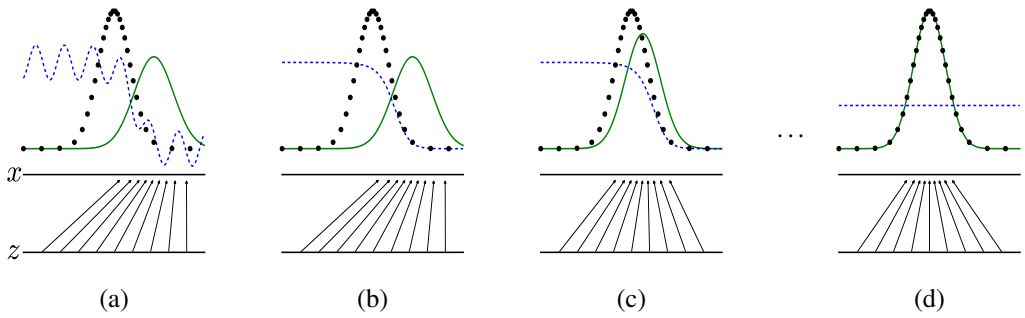


Figure 3.9: Adversarial Training Dynamics of a GAN. Obtained from [44]

- **Black dotted line (p_{data}):** Underlying data generating distribution of the real data [44].
- **Green solid line (p_g):** Distribution of the synthetic data created by the Generator (G) [44].
- **Blue dashed line ($D(x)$):** Discriminative distribution (D), indicating the probability that a sample x comes from the real data [44].
- **Upward arrows ($z \to x$):** Mapping function $x = G(z)$, showing how the uniform noise z is projected into the data space x, imposing the non-uniform distribution p_g on the transformed samples [44].
- **(a) Near Convergence:** The adversarial pair is considered near convergence. The generated distribution p_g is already similar to p_{data}, and the discriminator D is a partially accurate classifier [44].

- **(b) Training the Discriminator:** In the inner loop of the algorithm, the discriminator is trained to distinguish real samples from fake ones, converging to its optimal shape [44].
- **(c) Training the Generator:** After an update to G, the gradient of D has guided $G(z)$ to flow toward regions that are more likely to be classified as real data, which is observed by the shifting arrows [44].
- **(d) Nash Equilibrium:** After several steps of training, the generator perfectly models the real distribution ($p_g = p_{data}$). The discriminator is unable to differentiate between the two distributions, resulting in a constant value of $D(x) = \frac{1}{2}$ (a flat blue line) [44].

The Min-Max Objective Function

The training process is performed simultaneously for both networks in a strictly adversarial manner. The Discriminator is optimized to maximize the probability of correctly classifying both real and fake—generated—samples. At the same time, the Generator is optimized to minimize the probability that the Discriminator correctly labels its synthetic outputs—it effectively tries to maximize the Discriminator's error rate [8], [44].

This adversarial training process can be mathematically formalized as a min-max optimization problem [44]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.9}$$

where:

- The first term $\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)]$ represents the expected log-likelihood of the Discriminator correctly classifying real samples x. The Discriminator aims to maximize this term by assigning high probabilities to real data [44].
- The second term $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$ represents the expected log-likelihood of the Discriminator correctly classifying generated samples $G(z)$ as fake. The Generator aims to minimize this term by producing synthetic samples that the Discriminator incorrectly classifies as real [44].

From Theoretical Equilibrium to Real Instability: Mode Collapse

In a theoretical global equilibrium, this adversarial game converges to a Nash equilibrium—the Generator perfectly reproduces the target data distribution ($p_g = p_{data}$), and the Discriminator is left guessing randomly, outputting $D(x) = 0.5$ for all inputs [44]. However,

in real optimization scenarios, unconstrained adversarial networks often suffer from training instabilities [8]. The most critical issue is mode collapse, where the Generator learns to produce a small subset of the data distribution that consistently fools the Discriminator and subsequently maps multiple latent vectors to this identical geometric modality [8]. This results in a lack of structural diversity of the target distribution.

3.2.3 Deep Convolutional GAN (DCGAN)

The original GAN architecture proposed by Goodfellow et al. [44] relied on MLPs—fully connected networks—that were not designed to capture spatial hierarchies. This strictly limited its capacity to generate high-fidelity topological data and frequently led to training instabilities, such as mode collapse. To address this, Radford et al. [45] proposed the DCGAN.

The core of this innovation is the replacement of deterministic spatial pooling functions and fully connected layers with fully convolutional networks [45]. The Discriminator uses standard strided convolutions to downsample the input images and extract hierarchical features, while the Generator uses transposed convolutions—also known as fractionally-strided convolutions—to upsample a randomized latent vector z into a high-dimensional spatial matrix, or image [45] (see Figure 3.10).

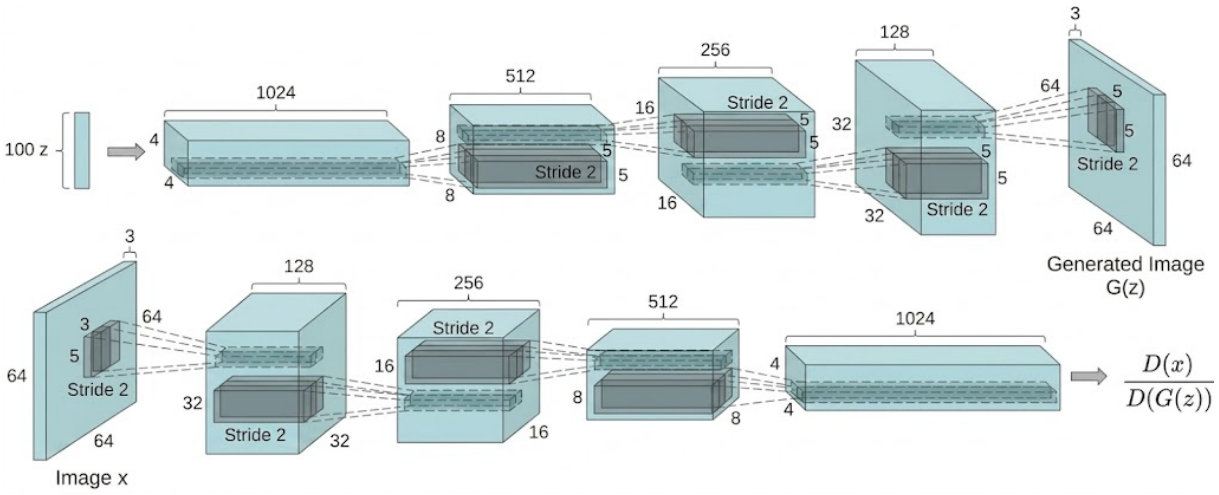


Figure 3.10: Architecture of a DCGAN (generated with Nano Banana). Adapted from [45]

Mathematically, while a standard convolution reduces spatial dimensions by computing a dot product between the kernel and the input, a transposed convolution reverses this spatial mapping. If the former can be defined as $Y = CX$, where C is the kernel matrix, the transposed convolution can be defined as $X = C^T Y$ [39], [45].

3.2.4 Conditional Generation and Hybridization (CVAE-GAN)

To address the limitations of isolated generative architectures, Bao et al. [9] proposed the CVAE-GAN architecture. This hybrid model combines the probabilistic latent space modeling of the VAE with the sharp adversarial learning mechanism of the DCGAN. To control the specific morphology of generated samples, this framework consists of four networks: an Encoder (E), a Generator (G)—acting as the Decoder of the VAE—, a Discriminator, and a Classifier [9].

The AI-generated figure 3.11 illustrates the overall structure of the CVAE-GAN model.

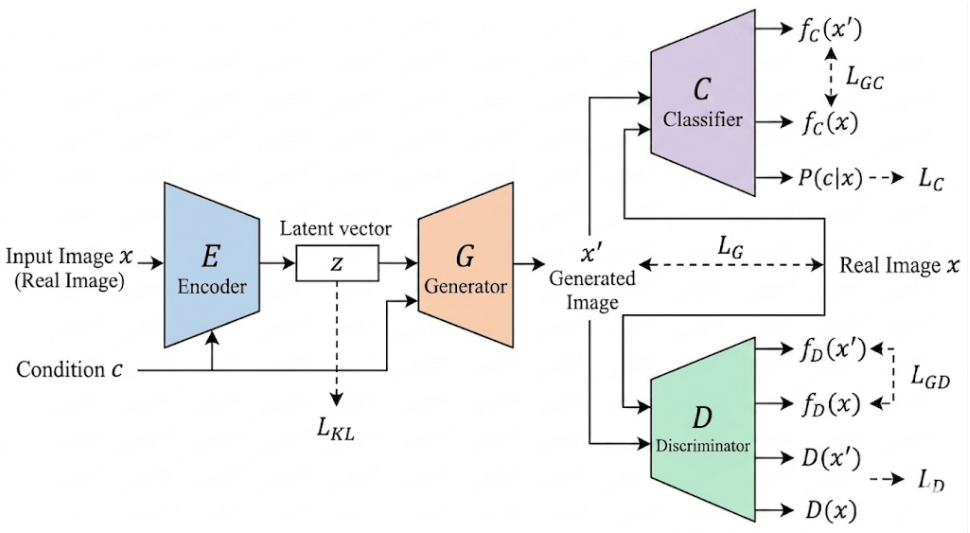


Figure 3.11: CVAE-GAN Architecture (generated with Nano Banana). Adapted from [9]

While the original CVAE-GAN utilizes a separate Classifier network, recent literature has evolved to integrate the classification task directly into the discriminator's architecture. This technique, known as the Auxiliary Classifier GAN (Auxiliary Classifier GAN (AC-GAN)), was introduced by Odena et al. [46] and further stabilized by Kang et al. [47]. Instead of a standalone classifier, the ACGAN Discriminator outputs both the probability of the sample being real or fake and the probability of the sample belonging to a specific class [46]. This modification allows the model to be trained efficiently end-to-end, eliminating the need for a separate classifier network while significantly improving overall generation performance and structural diversity [47].

Conditional Injection and The ACGAN

In this architecture, the Encoder E models a conditional posterior distribution $q(z|x,c)$ and the Generator G synthesizes the spatial data conditioned on the label as $G(z,c)$ [9]. The semantic consistency is enforced by the Auxiliary Classifier C that acts as an

independent network trained to predict the correct class label c of an input image x [9], [46]. The classification loss is defined as the categorical cross-entropy:

$$\mathcal{L}_C = -\mathbb{E}_{x,c}[\log P(C(x) = c)] - \mathbb{E}_{z,c}[\log P(C(G(z, c)) = c)] \tag{3.10}$$

Feature Matching Loss

The critical mathematical bridge uniting the VAE and GAN frameworks is the augmentation of the standard pixel-wise reconstruction penalty with a feature matching loss [9]. Unlike comparing only the raw pixel values, both the real image x and the synthetic reconstruction $G(z, c)$ are passed through the convolutional layers of the Discriminator D. Here, $f(x)$ denotes the activations of these intermediate layers. The Generator is then optimized to minimize the distance between these abstract spatial representations (L_2 Euclidean distance) [9]:

$$\mathcal{L}_{FM} = \mathbb{E}_{x,z\sim q(z|x,c)} \left[||f(x) - f(G(z, c))||_2^2\right] \tag{3.11}$$

The Unified Objective Function

The whole four-network architecture undergoes an asymmetric training process optimized via a joint objective function [9]. The Encoder E minimizes the KLD ($\mathcal{L}_{KL}$), the Discriminator D maximizes the conditional adversarial log-likelihood ($\mathcal{L}_{GAN}$), the Classifier C minimizes the categorical cross-entropy ($\mathcal{L}_C$), and the Generator G minimizes both the feature matching distance ($\mathcal{L}_{FM}$) and the classification error ($\mathcal{L}_C$) [9]. The overall objective function can be formalized as follows:

$$\mathcal{L}_{CVAE-GAN} = \lambda_1\mathcal{L}_{KL}(E) + \lambda_2\mathcal{L}_{FM}(G) + \lambda_3\mathcal{L}_{GAN}(G, D) + \lambda_4\mathcal{L}_C(G, C) \tag{3.12}$$

where $\lambda_1, \lambda_2, \lambda_3$, and λ_4 are scaling hyperparameters necessary to balance the importance of the gradients flowing from each distinct network [9]. In particular, balancing λ_2—the topological reconstruction fidelity—against λ_3—the adversarial sharpness—is strictly required. In this unified optimization scheme, the representations extracted from D and C act as regularizers for G, preventing mode collapse while ensuring that the generated samples adhere to the real morphological conditions [9].

3.3 Denoising Diffusion Probabilistic Models (DDPMs)

To address the fundamental limitations of adversarial training—such as mode collapse—DDPMs were introduced by Ho et al. [11]. These models were inspired by non-equilibrium thermodynamics and replace adversarial training with a mathematically stable and tractable likelihood-based framework: a fixed forward process that systematically destroys the data distribution by adding Gaussian noise and a learned reverse process that restores it [11].

3.3.1 Forward Diffusion Process (Markov Chains)

The forward diffusion process, denoted as q, is a fixed Markov chain that gradually adds Gaussian noise to an initial real image $x_0 \sim q(x_0)$ over T discrete time steps [11]. At each step t, the transition probability solely depends on the previous state x_{t-1}, mathematically defined as [11]:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{3.13}$$

where $\beta_t \in (0,1)$ is the noise schedule that controls the amount of variance added at each step t. Consequently, the total approximate posterior $q(x_{1:T}|x_0)$ after T steps is the product of these conditional transitions [11]:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{3.14}$$

A crucial property of this Markovian approach is that the distribution of any intermediate state x_t can be directly sampled from the original image x_0, without the need to compute the intermediate steps iteratively [11]. By defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the forward process can be expressed in closed form as [11]:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I) \tag{3.15}$$

This translates to the following sampling procedure:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{3.16}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is the standard Gaussian noise tensor [11]. As $T \to \infty$ and $\bar{\alpha}_T \to 0$, the distribution of x_T converges to a standard isotropic Gaussian distribution $\mathcal{N}(0, I)$, ensuring that the signal is completely destroyed [11] (see Figure 3.12. AI-generated illustration of the deterministic corruption of Gaussian noise).

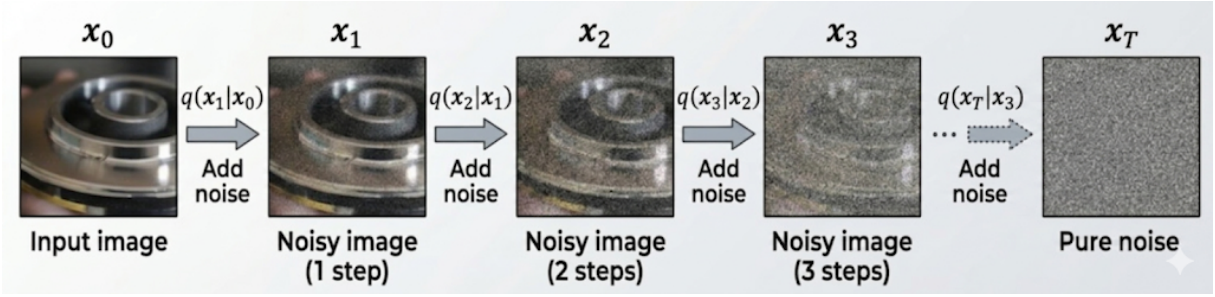


Figure 3.12: Forward Diffusion Process (generated with Nano Banana). Adapted from [11]

3.3.2 Reverse Denoising Process

While the forward process q is deterministic and strictly defined by the variance schedule, the reverse process requires inference of the intractable posterior distribution $q(x_{t-1}|x_t)$. DDPMs approximate this distribution by deploying a neural network parameterization, defining a learned Markov chain p_θ [11]:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{3.17}$$

Rather than predicting the original image x_0 directly, Ho et al. [11] demonstrated that the architecture converges more robustly and achieves better sample quality when tasked exclusively with predicting the specific noise tensor ϵ injected during the forward process. By conditioning the tractable forward process posterior on the initial state x_0, the parameterized mean μ_θ can be redefined as a deterministic function of the network's noise prediction $\epsilon_\theta(x_t, t)$ [11]:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) \tag{3.18}$$

This formulates the generative sampling process: at any arbitrary noisy state x_t, a CNN-based architecture—commonly a U-Net—estimates the residual noise [11]. Lastly, this reparameterized equation dictates how to subtract this estimation to iteratively obtain the denoised state x_{t-1} [11].

The AI-generated figure 3.13 illustrates the reverse denoising process, where the network learns to iteratively remove noise from a pure Gaussian sample to reconstruct a high-fidelity image.

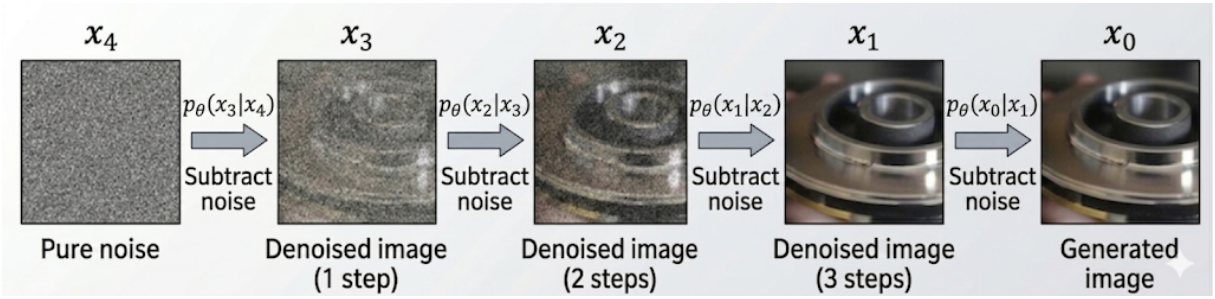


Figure 3.13: Reverse Denoising Process (generated with Nano Banana). Adapted from [11]

3.3.3 Diffusion Objective Function (Variational Lower Bound)

Training a DDPM involves optimizing the log-likelihood of the training data. Since direct optimization of the marginal likelihood is analytically intractable, the model optimizes a Variational Lower Bound (VLB), conceptually similar to the objective of a VAE [11].

According to Ho et al. [11], the complex VLB objective can be significantly simplified. Since the forward process q is deterministic and the reverse process p_θ is parameterized to predict the noise tensor ϵ, the probabilistic objective reduces to minimizing the MSE between the true noise ϵ and the network's prediction ϵ_θ [11]. This is mathematically defined as follows:

$$\mathcal{L}_{simple} = \mathbb{E}_{t,x_0,\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[||\epsilon - \epsilon_\theta(x_t, t)||_2^2\right] \tag{3.19}$$

where x_t is the noisy state computed from x_0 and ϵ using the closed-form forward sampling process. By minimizing this unweighted L_2 loss, the U-Net learns to reverse the Markov chain by uniformly sampling the time steps t to obtain the global denoising gradients [11]. Furthermore, discarding the weighting of the original variational bound naturally down-weights the loss terms corresponding to small t, allowing the network to focus on the more difficult denoising tasks at larger time steps, which ultimately leads to better sample quality [11].

3.4 Latent Diffusion Models (LDMs) and Stable Diffusion (SD)

Standard DDPMs successfully generate high-fidelity images, processing both the forward and reverse Markov chains directly in the high-dimensional pixel space. However, this approach is computationally expensive and causes heavily bottlenecked training and in-

ference [10]. To address the dimensionality constraint, Rombach et al. [10] proposed the LDMs, which perform the diffusion process in a highly compressed latent space.

3.4.1 Perceptual Image Compression (The VAE connection)

Instead of applying noise to raw pixels, LDMs utilize a pre-trained VAE backbone. An Encoder $\mathcal{E}$ compresses the input image x into a dense, continuous latent representation $z = \mathcal{E}(x)$ [10]. After the diffusion process, a Decoder $\mathcal{D}$ reconstructs the image from the latent space back to the pixel space $\tilde{x} = \mathcal{D}(z)$ [10].

3.4.2 Latent Space Denoising via U-Net

By isolating the diffusion process to the compressed latent space $\mathcal{Z}$, the U-Net architecture ϵ_θ operates on significantly smaller spatial dimensions, increasing the computational efficiency while still capturing the topological fidelity [10]. The simplified training objective is defined as:

$$\mathcal{L}_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))||_2^2\right] \tag{3.20}$$

3.4.3 Cross-Attention Mechanism and Semantic Conditioning

A crucial characteristic of the SD architecture is the integration of a robust conditioning mechanism, which steers the explicit morphology of the generated samples [10]. The condition y is pre-processed by a domain-specific encoder $\tau_\theta(y)$——most notably, the CLIP model [24] for semantic text-to-image mapping——and injected directly into the U-Net's intermediate layers through a cross-attention mechanism [10].

The attention operation relies on mapping the flattened latent features into three distinct representations: queries Q, keys K, and values V. The attention output is computed as a weighted sum of the values, where the weights are determined by the similarity between the queries and keys [10], [25]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3.21}$$

where d is the dimensionality of the key vectors. This mechanism allows the model to mathematically fuse the latent spatial features with the semantic information at multiple levels of the architecture. With this, LDMs guarantee that the reverse denoising pro-

cess is explicitly guided by designated class features or text prompts, providing a highly controllable generative framework [10].

The AI-generated figure 3.14 illustrates the stable diffusion architecture, where all the elements—Encoder, U-Net, cross-attention, and Decoder—are integrated to perform the latent diffusion process [10].

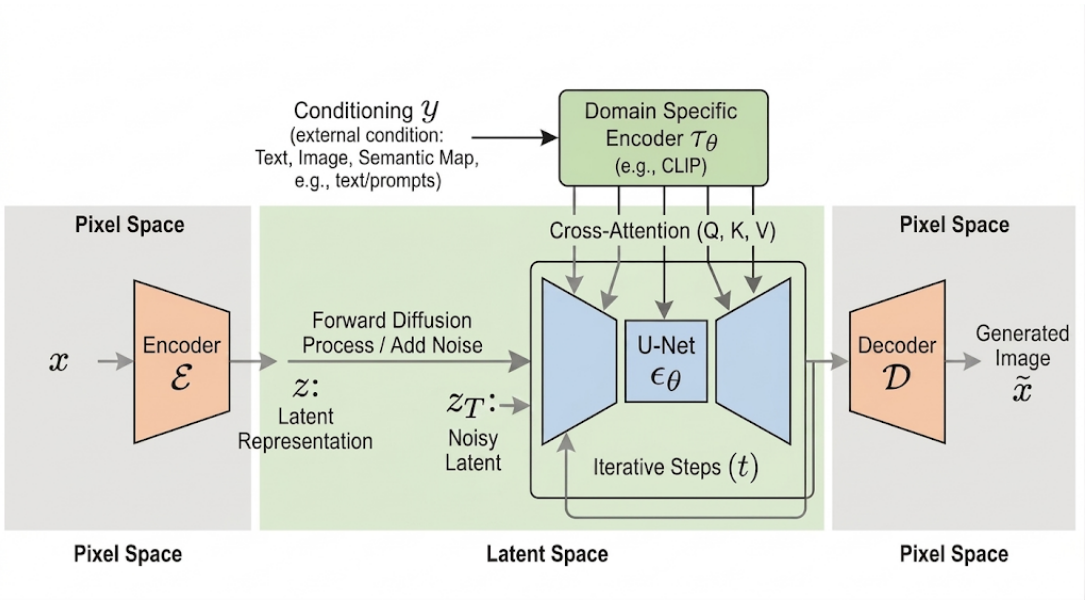


Figure 3.14: Stable Diffusion Architecture (generated with Nano Banana). Adapted from [10]

3.5 Advanced Spatial and Textural Conditioning

Base LDMs architectures natively support semantic guidance through cross-attention; however, synthesizing explicit spatial structures—such as manufacturing defects—requires architectural improvements. To enforce deterministic spatial controls and adapt the pre-trained model to highly specific high-frequency details without catastrophic forgetting, the base LDM architecture must be augmented with additional techniques.

3.5.1 ControlNet Architecture

To achieve pixel-level control over the generated output, Zhang et al. [12] proposed the ControlNet architecture. This framework extends conditioning capabilities by explicitly freezing the pre-trained diffusion weights (Θ_{locked}) and enabling a trainable copy of the same architecture ($\Theta_{trainable}$) to learn task-dependent spatial conditions c_f (e.g., edge maps) [12].

The trainable outputs are injected into the base model using "zero convolutions" ($\mathcal{Z}$)—layers initialized with zero weights and biases to preserve the original generative distribution at the onset of training [12]. The augmented output y_c for a neural block $\mathcal{F}$ with input x is defined as follows:

$$y_c = \mathcal{F}(x; \Theta_{locked}) + \mathcal{Z}_{out}(\mathcal{F}_c(x + \mathcal{Z}_{in}(c_f); \Theta_{trainable})) \tag{3.22}$$

Minimizing the standard LDM objective with this vector allows the framework to simultaneously follow the semantic prompts and conform to rigid spatial morphologies.

The AI-generated figure 3.15 illustrates the ControlNet architecture, where the trainable branch learns to extract spatial conditions and injects them into the frozen base model through zero convolutions [12].

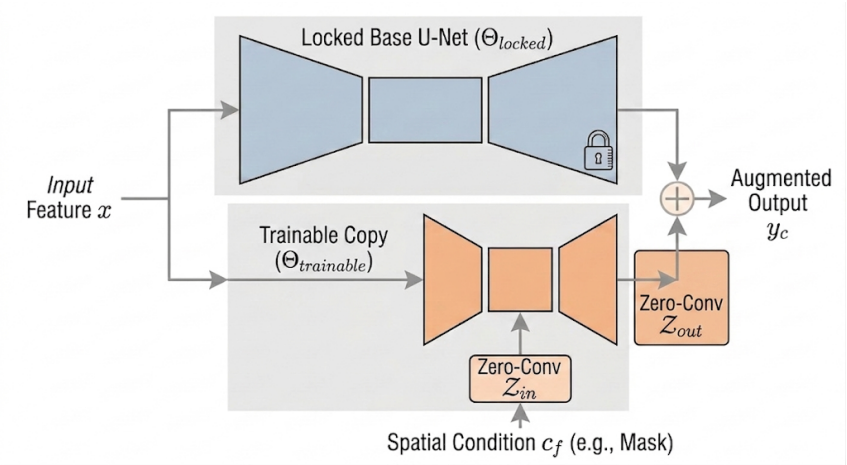


Figure 3.15: ControlNet Architecture (generated with Nano Banana). Adapted from [12]

3.5.2 Low-Rank Adaptation (LoRA)

Adapting the pre-trained feature distribution of a base LDM to the specific high-frequency details of the target morphology requires a full model fine-tuning, which is computationally prohibitive. To tackle this issue, Hu et al. [13] introduced LoRA, a Parameter-Efficient Fine-Tuning (PEFT) technique.

LoRA proposes that the weight updates during domain adaptation possess a low intrinsic rank [13]. Instead of updating the full pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA freezes W_0 and injects a trainable low-rank decomposition matrix $\Delta W = BA$ where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with $r \ll \min(d, k)$ [13]. The adapted weight matrix is then defined as follows:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{3.23}$$

Injected predominantly into the cross-attention layers of the U-Net, matrix A is initialized with a random Gaussian distribution, while B is initialized with zeros to preserve the property $\Delta W = 0$ at initialization [13]. This allows the model to efficiently learn the textural distributions using a minimal fraction of the trainable parameters.

3.6 Quantitative Evaluation Metrics

To objectively evaluate the fidelity, diversity, and perceptual quality of the generated images, traditional pixel-wise metrics—such as MSE—are insufficient, as they often fail to correlate with human visual perception [22]. Therefore, more sophisticated metrics that capture the underlying feature distributions and perceptual similarities are required.

3.6.1 Fréchet Inception Distance (FID)

The FID is a widely used standard for evaluating the overall quality and diversity of generated image distributions. Instead of comparing images at the pixel level, FID compares the statistics of deep feature representations extracted from a pre-trained Inception-v3 classifier [21].

By utilizing the coding layer of the Inception model, FID assumes that the extracted vision-relevant features of both the real and generated distributions follow a multidimensional Gaussian distribution [21]. The metric then calculates the Fréchet distance—also known as the Wasserstein-2 distance—between these two distributions, which are characterized by their mean and covariance matrices [21]:

$$\text{FID} = ||\mu_r - \mu_g||_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}\right) \tag{3.24}$$

where $||\cdot||_2^2$ denotes the squared Euclidean distance—or squared L_2 norm—between the mean feature vectors μ_r and μ_g of the real and generated distributions, respectively. $\text{Tr}(\cdot)$ denotes the trace of the matrix. A lower FID score indicates that the generated distribution more closely matches the real distribution, implying a higher quality and diversity of the generated images [21].

3.6.2 Learned Perceptual Image Patch Similarity (LPIPS)

While FID evaluates the macroscopic distribution similarity, the LPIPS metric quantifies the perceptual distance between two individual images. Introduced by Zhang et al. [22], LPIPS addresses the limitations of traditional pixel-wise metrics by aligning the mathematical distance with human perceptual judgments.

LPIPS leverages the internal activations of pre-trained CNNs—such as AlexNet, VGG, or SqueezeNet [22]. For a given pair of images—a reference patch x and a generated patch x_0—feature maps are extracted from multiple layers of the network and unit-normalized in the channel dimension, denoted as $\hat{y}^l$ and $\hat{y}^l_0$ for layer l [22]. The distance is then computed as a weighted sum of the squared L_2 distances between these normalized feature maps across all layers [22]:

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} ||w_l \odot (\hat{y}^l_{hw} - \hat{y}^l_{0hw})||_2^2 \tag{3.25}$$

where $\odot$ denotes element-wise multiplication and w_l represents a layer-specific scaling vector. By mathematically computing the distance in a hierarchical deep feature space rather than a flattened pixel space, LPIPS robustly captures nuanced structural and textural similarities. A lower LPIPS score signifies a higher perceptual resemblance between the synthesized topology and the reference condition, effectively mirroring human visual perception [22].

4. METHODICAL DEVELOPMENT

This chapter details the experimental methodology and implementation strategies employed to validate the proposed hypothesis in this work. Building upon the theoretical and mathematical foundations established in the previous chapter, the following sections provide a comprehensive and replicable framework of the generative benchmark.

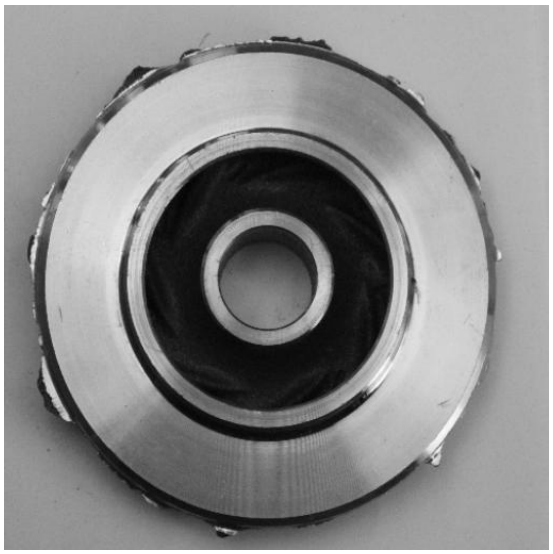
4.1 Dataset Preparation and Preprocessing

4.1.1 Data Acquisition

The experimental validation of the proposed generative frameworks relies on the *casting product image data for quality inspection* dataset [14], which focuses on submersible pump impeller manufactured by casting. The data consist of high-resolution Red, Green, Blue (RGB), top-down industrial inspection images of the impeller, which consist of a circular metallic structure (see Figure 4.1). The dataset is divided into two classes:

- **ok_front:** Representing flawless impeller samples that passes quality control. These images serve as the reference distribution for the non-defective class (see Figure 4.1a).
- **def_front:** Representing rejected impeller samples that exhibit blow holes, pinholes, burr, shrinkage defects, mould material defects, pouring metal defects, metallurgical defects among other manufacturing anomalies. These images serve as the reference distribution for the defective class (see Figure 4.1b).

(a) "ok-front" Impeller Sample



(b) "def-front" Impeller Sample

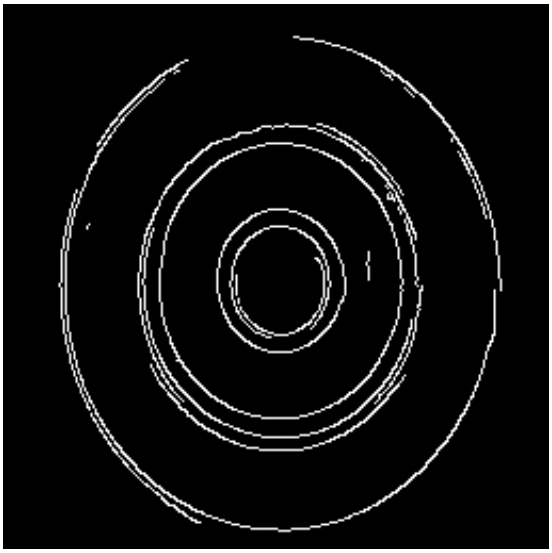
Figure 4.1: Representative samples of casting product dataset. Obtained from [14].

4.1.2 Pipeline Transformations

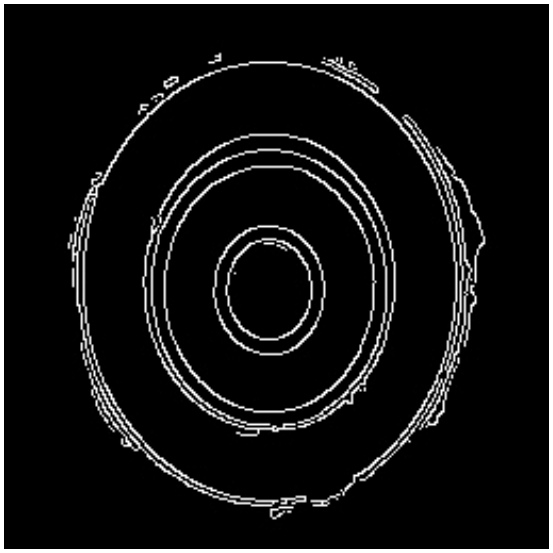
To satisfy the distinct numerical and structural input requirements of the adversarial and diffusion architectures, two specialized preprocessing pipelines were implemented, enforcing spatial standardization while diverging in their normalization strategies.

1. **Adversarial Standardization:** Input images are scaled using a sequential transformation optimized via `torchvision.transforms`, applying a resize operation that enforces a fixed spatial resolution of 256×256 pixels. Subsequently, to properly align the empirical pixel distribution with the generator's analytical constraints, a normalization transform scales the tensors using a mean of 0.5 and a standard deviation of 0.5 across all RGB channels, mapping the pixel intensities from the original $[0, 1]$ range to a symmetrical $[-1, 1]$ domain to match a Hyperbolic Tangent (Tanh) activation function.

2. **Geometric Conditioning Extraction:** For the LDM pipeline, a secondary transformation flow was developed to extract rigid structural boundaries. This path applies the Canny edge detection algorithm directly to the nominal images. The resulting edge maps are converted to tensors and strictly maintained within a $[0, 1]$ range, bypassing the $[-1, 1]$ scaling to serve as raw spatial priors for the ControlNet conditioning layers (see Figure 4.2).



(a) "ok-front" Canny Edge Map



(b) "def-front" Canny Edge Map

Figure 4.2: Canny edge maps of casting product dataset.

3. **Multimodal Tokenization:** To enable semantic steering inside the diffusion U-Net, descriptive textual prompts are processed via the pretrained CLIP tokenizer. These prompts utilize a specialized identifier template, formalized as "photo of SKS_PART, top-down industrial inspection, circular machined component, flat grey

ferrous material, [condition]", where the placeholder is dynamically substituted by class-specific descriptors (e.g., "flawless smooth surface, QC passed" for nominal components or "severe porosity defect, surface blowholes and cavities, QC failed" for anomalies). Truncation and maximum padding are automatically enforced to output a static shape of 77×768 tokens, ensuring exact structural alignment with the text encoder's cross-attention projection matrices.

4.2 Hardware and Software Environment

4.2.1 Computational Infrastructure

The computational intensive execution of the training processes and the synthesis of data were performed using a local high-performance workstation optimized for DL workflows. The primary acceleration hardware utilized was an NVIDIA® GeForce™ RTX 5080. This consumer-grade GPU was critical for managing the high-dimensional tensor operations and parallel processing demands inherent to both the adversarial and diffusion architectures.

The main specifications of the computational infrastructure are summarized in Table 4.1:

Processor:	Intel® Core™ i9-14900K CPU @ 3.20GHz
RAM Memory:	64 GB DDR5
GPU:	NVIDIA® GeForce™ RTX 5080
GPU Memory:	16 GB GDDR7
Operating System:	Windows 11 Pro
System Architecture:	x86-64
GPU Architecture:	Blackwell
Driver Version:	595.79
CUDA Version:	13.2

Table 4.1: Computational Infrastructure Specifications

4.2.2 Software Stack

The development, training, and tracking infrastructure were built entirely upon a specialized Python-based open-source ecosystem. The core tensor operations, automatic differentiation graphs and low-level architectural abstractions were handled by `PyTorch`.

The software dependencies are structured as follows:

- **Adversarial Baseline Framework:** Implemented using native PyTorch modules

(`torch.nn`), utilizing `torchvision` transforms for standard tensor operations and mathematical weight initializations.

- **Diffusion Pipeline Ecosystem:** Leveraging the Hugging Face ecosystem, the `diffusers` library was utilized to initialize the pretrained SD v1.5 foundational components, the DDPMScheduler noise mechanics, and the pre-trained Canny ControlNet layers. The textual tokenization and prompt encoding hidden states were extracted using the Hugging Face `transformers` library.
- **Parameter-Efficient Adapters:** The injection of LoRA matrices into the U-Net's attention projection pathways was managed exclusively via the PEFT library, wrapping the target layers without disrupting the frozen weights of the base model.
- **Experiment Tracking and Logging:** Hyperparameter documentation, step-wise training losses, epoch averages, and temporary validation artifacts were managed using `MLflow`. This setup enabled monitoring of the optimization curves and provided systematic storage for the synthesized validation checkpoints.

4.3 Baseline Model Implementation AC-CVAE-GAN

4.3.1 Architectural Setup

The adversarial baseline architecture is designed to enforce categorical control over the generated image topologies. This framework combines a continuous variational latent space with an Auxiliary Classifier (AC). While traditional CVAE-GAN formulations are composed of four independent networks [9], the custom architecture developed in this work optimizes computational efficiency by integrating the AC mechanism from the ACGAN framework [46] (see Figure 4.3). The resulting unified model—denoted as AC-CVAE-GAN—can be categorized into four logical sub-networks:

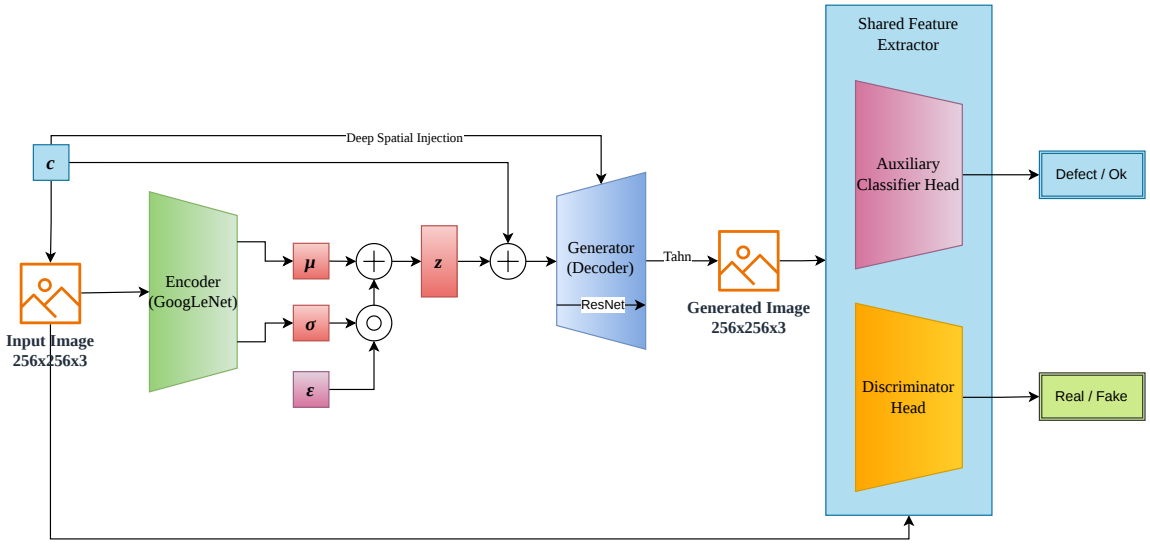


Figure 4.3: Baseline Architecture: AC-CVAE-GAN.

Encoder

The encoder network utilizes a truncated, non-pretrained GoogLeNet backbone. The structural hierarchy is sliced to isolate the primary feature extractor, completely discarding the final classification layers. This design extracts a 1024-dimensional feature vector from the standardized 256×256 image.

This vector is passed to a dense linear layer that compresses the representations down to 512 dimensions under a ReLU activation function. Finally, two parallel linear layers project these representations into the mean (μ) and logarithm of variance ($\log \sigma^2$) vectors of the 128-dimensional latent space.

In the figure 4.4, the baseline encoder architecture is detailed.

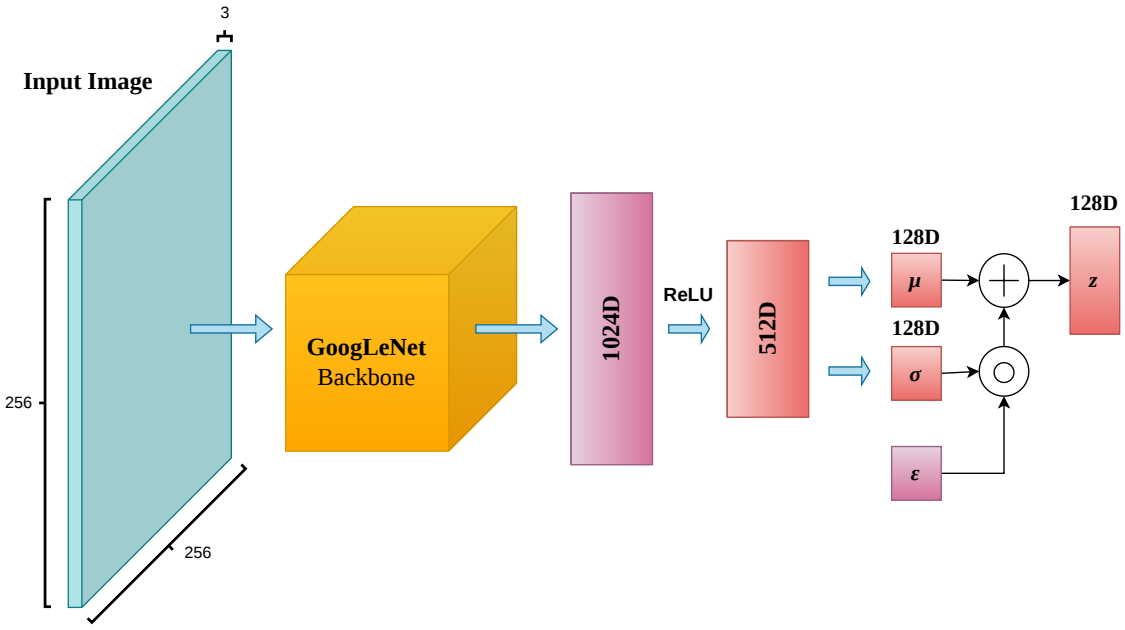


Figure 4.4: Baseline Encoder Architecture.

Decoder (Generator)

The decoder instantiates a deep spatial conditioning mechanism. The 2-element one-hot categorical label is projected via dense embedding layer to a 64-dimensional continuous vector and concatenated with the 128-dimensional latent sample, forming a 192-dimensional input vector. Two sequential dense layers map this vector from 192 to 1024, and subsequently to 8,192 features, which are unflattened into a spatial feature map of $512 \times 4 \times 4$.

To prevent the network from losing class conditioning in deeper layers, the original label (c) is spatially expanded to a $2 \times 4 \times 4$ tensor and concatenated along the channel axis. The upsampling trajectory consists of six transposed convolutional layers. Custom ResNet block modules are interspersed after the first five upsampling operations to redefine high-frequency casting textures.

The final layer maps directly to a Tanh activation function without normalization.

In the figure 4.5, the baseline decoder architecture is detailed.

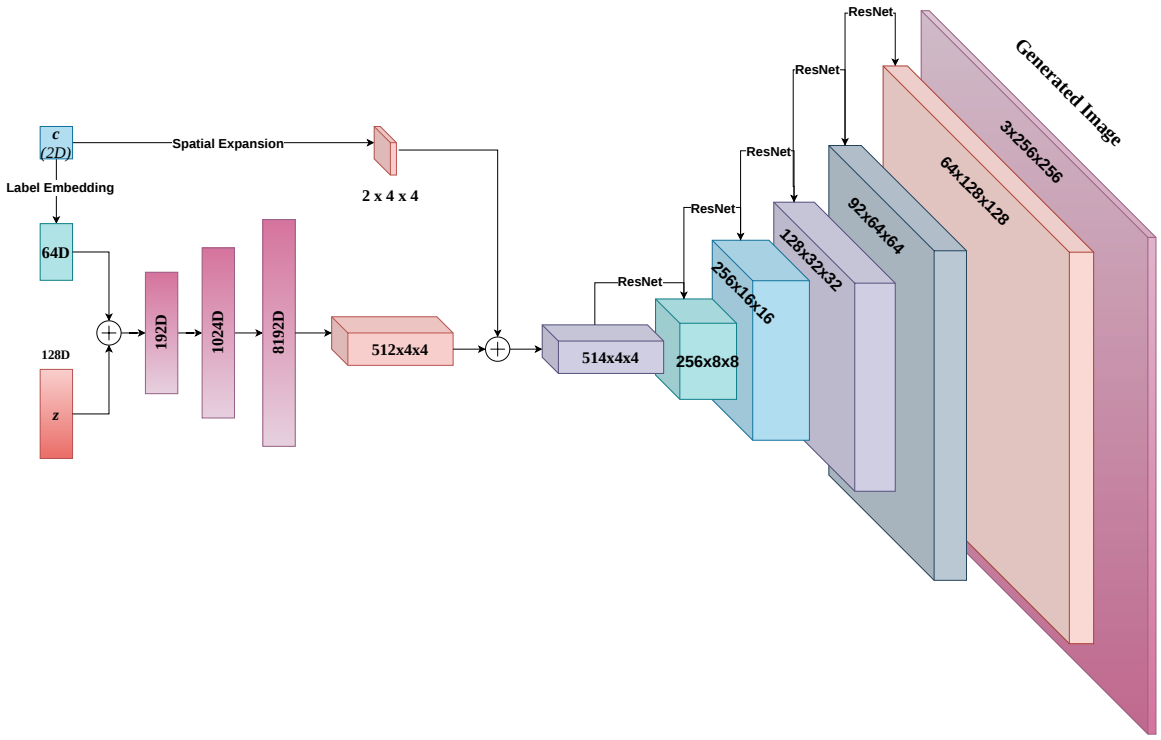


Figure 4.5: Baseline Decoder Architecture.

Discriminator

Acting as the adversarial evaluator, the discriminator uses a five-layer convolutional feature extractor. Layers 1 through 5 implement a 4×4 kernel, a stride of 2 and a padding of 1, scaling channels sequentially from 3 up to 1024, LeakyReLU functions with a negative slope of 0.2 and 2D batch normalization are applied after each convolutional layer.

The specific discriminative branch culminates in a convolutional layer that outputs a binary real/fake classification, producing a single scalar via global pooling.

The figure 4.6 shows the baseline discriminator architecture.

Auxiliary Classifier

Rather than operating as an isolated network, this classifier shares the deep convolutional feature extractor of the Discriminator. From the final 1024-dimensional feature map, it diverges into a parallel convolutional branch that outputs a 2-element logit vector mapping to the categorical dataset labels, effectively guiding the generator towards producing class-correct samples—defective or good parts.

The figure 4.6 shows the baseline auxiliary classifier architecture.

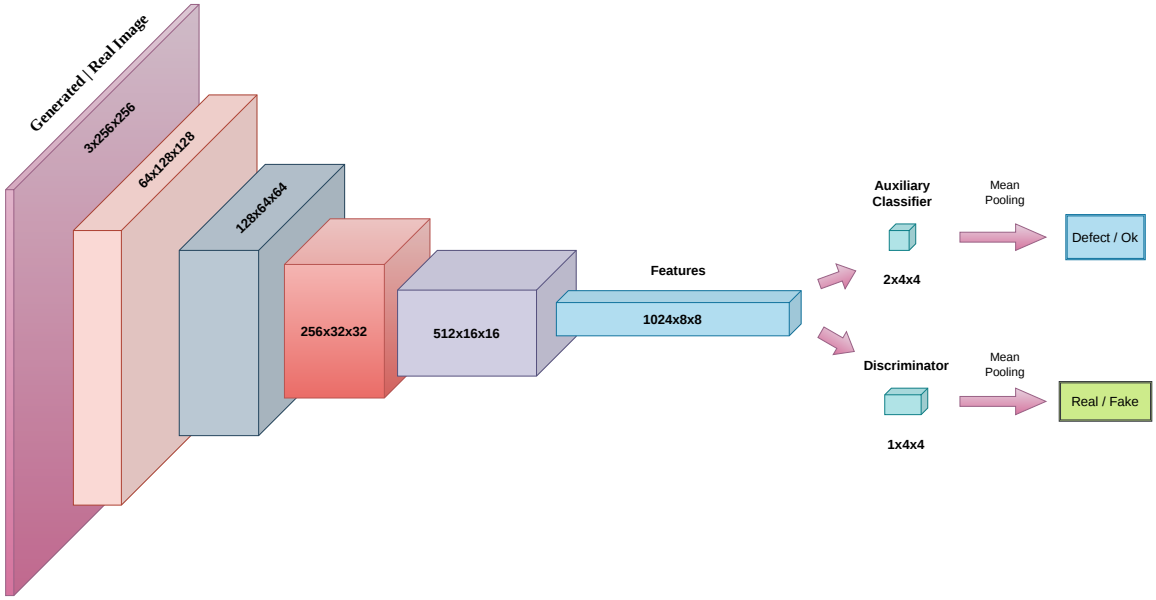


Figure 4.6: Baseline Discriminator and Auxiliary Classifier Architecture.

4.3.2 Training Protocol

The training of the AC-CVAE-GAN Baseline framework consists of a highly stabilized joint optimization process, designed to balance the variational inference of the Encoder with the adversarial min-max game of the Generator and Discriminator. To prevent early stage instability, the training is explicitly divided into a Variational Warmup Phase and the Adversarial Equilibrium Phase.

The training algorithm is summarized in Algorithm1.

Objective Function

The global optimization relies on a composite loss function. An important architectural choice of this implementation is the replacement of a standard generator adversarial loss with a Feature Matching criterion to mitigate mode collapse.

For the Discriminator and Classifier (D, C), the objective is to correctly classify real images and predict their corresponding labels. To prevent the Discriminator from becoming overconfident, one-side label smoothing is applied (real labels $\sim U(0.9, 1.0)$, fake labels $\sim U(0.0, 0.1)$), along with decaying instance noise added to the input images. The loss formulation is defined as:

$$\mathcal{L}_{D,C} = \frac{1}{2}\left(\mathcal{L}^{real}_{Adv\_D} + \mathcal{L}^{fake}_{Adv\_D}\right) + \lambda_{aux}\mathcal{L}_{Aux\_D} \tag{4.1}$$

Where $\mathcal{L}_{Adv\_D}$ calculates the binary classification error, and $\mathcal{L}_{Aux\_D}$ evaluates the categorical cross-entropy of the defect classification. The auxiliary weight is maintained at $\lambda_{aux} = 1.0$.

For the Encoder and Generator (E, G), the composite loss relies on strictly structural reconstruction, latent regularization, feature distribution matching, and auxiliary classification. The loss is defined as:

$$\mathcal{L}_{E,G} = \frac{1}{16}\mathcal{L}_{Recon} + \beta_{anneal}\mathcal{L}_{KL} + \gamma\mathcal{L}_{Feat} + \lambda_{aux}\mathcal{L}_{Aux\_G} \tag{4.2}$$

Where $\mathcal{L}_{Recon}$ is the spatial MSE ($\mathcal{L}_2$ distance) scaled down by a factor of 16 to compensate for the 256×256 high-resolution pixel increment. $\mathcal{L}_{KL}$ represents the KLD. Crucially, the standard adversarial loss is replaced by $\mathcal{L}_{Feat}$, a Feature Matching loss computed via the Mean Absolute Error (MAE) ($\mathcal{L}_1$ norm) between the intermediate feature maps of the Discriminator for real and generated images, weighted by γ.

Loss Optimization

The optimization is performed using the Adam optimizer with alternating gradient updates, following a phased training loop. To maintain a stable min-max equilibrium, the learning rates are paired with momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.9999$, alongside an explicit L_2 weight decay penalty across all subnetworks to prevent overfitting.

- **Phase 1: CVAE Warmup.** During the initial warmup epochs, the Discriminator's gradient updates are disabled. The Encoder and Generator are trained exclusively on $\mathcal{L}_{Recon}$ and $\mathcal{L}_{KL}$ to establish a mathematically consistent latent space and elemental structural synthesis before the adversarial dynamics begin.
- **Phase 2: GAN Enabling.** Once the warmup finishes, the adversarial feature matching ($\gamma\mathcal{L}_{Feat}$) and auxiliary losses are enabled.
- **KL Annealing.** The weight of the KL divergence is dynamically scaled using a linear annealing schedule to avoid posterior collapse. The weight starts at 0 and gradually increases to a target value β along the warmup phase ($\beta_{anneal} = \beta \cdot min(1.0, epoch/warmup\_epochs)$)

During the GAN phase, the Generator is evaluated and its losses are computed with a frozen Discriminator. Furthermore, the Discriminator's weights are unfrozen, it evaluates

noisy versions of real and generated samples, computes $\mathcal{L}_{D,C}$, and performs gradient back-propagation to update its weights. This controlled process prevents vanishing gradients and ensures the reliable generation of synthetic defect images.

Training Algorithm

Algorithm 1 AC-CVAE-GAN Training Protocol

Require: Training dataset $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$
Require: Hyperparameters: Total epochs E, Warmup epochs W, Batch size B
Require: Opt. params: Learning rates α_G, α_D, Weight decay λ_{L2}, Loss weights $\beta, \gamma, \lambda_{aux}$
Ensure: Optimized network parameters $\theta^*_{E,G}$ and θ^*_D

Initialize $\theta_E, \theta_G, \theta_D$ with $\mathcal{N}(0, 0.02)$
Initialize Adam Optimizers with decoupled L_2 penalty (λ_{L2})
for $e = 1$ **to** E **do**
  $\beta_{anneal} \leftarrow \beta \cdot \min(1.0, \frac{e}{W})$ ▷ KL divergence annealing
  $\sigma_{noise} \leftarrow 0.1 \cdot \max(0, 1.0 - \frac{e}{E \cdot 0.5})$ ▷ Decaying instance noise
  for each mini-batch $(x, c) \sim \mathcal{D}$ **do**
    Encode label c into one-hot format
    — **Phase 1: Encoder/Generator Update** —
    $\mu, \sigma \leftarrow E(x)$ ▷ Blind/Unconditional feature extraction
    Sample latent vector $z = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
    Generate synthesized image $\tilde{x} \leftarrow G(z, c)$ ▷ Conditional synthesis
    Compute Reconstruction Loss: $\mathcal{L}_{Recon} \leftarrow \text{MSE}(\tilde{x}, x)$
    Compute KL Divergence: $\mathcal{L}_{KL} \leftarrow D_{KL}(\mathcal{N}(\mu, \sigma^2) \parallel \mathcal{N}(0, I))$
    Initialize $\mathcal{L}_{Feat} \leftarrow 0$, $\mathcal{L}_{Aux\_G} \leftarrow 0$
    if $e \geq W$ **then** ▷ Adversarial Phase Active
      Extract intermediate features: $f_{real} \leftarrow D_{feat}(x)$, $f_{fake} \leftarrow D_{feat}(\tilde{x})$
      Compute Feature Matching Loss: $\mathcal{L}_{Feat} \leftarrow L_1(f_{fake}, f_{real})$
      Compute Generator Auxiliary Loss: $\mathcal{L}_{Aux\_G} \leftarrow \text{CrossEntropy}(D_{cls}(\tilde{x}), c)$
    end if
    $\mathcal{L}_{E,G} \leftarrow \frac{1}{16}\mathcal{L}_{Recon} + \beta_{anneal}\mathcal{L}_{KL} + \gamma\mathcal{L}_{Feat} + \lambda_{aux}\mathcal{L}_{Aux\_G}$
    Update parameters: $\theta_{E,G} \leftarrow \theta_{E,G} - \alpha_G \nabla_{\theta_{E,G}} \mathcal{L}_{E,G}$
    — **Phase 2: Discriminator Update** —
    if $e \geq W$ **then**
      Apply instance noise: $x_{noisy} \leftarrow x + \mathcal{N}(0, \sigma^2_{noise})$, $\tilde{x}_{noisy} \leftarrow \tilde{x} + \mathcal{N}(0, \sigma^2_{noise})$
      Assign smoothed labels: $y_{real} \sim \mathcal{U}(0.9, 1.0)$, $y_{fake} \sim \mathcal{U}(0.0, 0.1)$
      Evaluate real samples: $p_{real\_rf}, p_{real\_cls} \leftarrow D(x_{noisy})$
      Evaluate fake samples: $p_{fake\_rf}, p_{fake\_cls} \leftarrow D(\tilde{x}_{noisy})$
      Compute Adversarial Loss: $\mathcal{L}_{Adv\_D} \leftarrow \frac{1}{2}\left[\text{BCE}(p_{real\_rf}, y_{real}) + \text{BCE}(p_{fake\_rf}, y_{fake})\right]$
      Compute Auxiliary Loss: $\mathcal{L}_{Aux\_D} \leftarrow \frac{1}{2}\left[\text{CE}(p_{real\_cls}, c) + \text{CE}(p_{fake\_cls}, c)\right]$
      $\mathcal{L}_{D,C} \leftarrow \mathcal{L}_{Adv\_D} + \lambda_{aux}\mathcal{L}_{Aux\_D}$
      Update parameters: $\theta_D \leftarrow \theta_D - \alpha_D \nabla_{\theta_D} \mathcal{L}_{D,C}$
    end if
  end for
end for
return $\theta_E, \theta_G, \theta_D$

4.4 Diffusion Model Implementation: SD-LoRA-ControlNet

Unlike the adversarial approach described in the previous section, the implementation of the LDM proposed in this work leverages a pre-trained foundational architecture. This section outlines the implementation and fine-tuning and optimization of the SD v1.5 framework, augmented with LoRA and structural conditioning via ControlNet.

4.4.1 Architectural Adaptation and Dual Conditioning

This approach leverages a pre-trained Autoencoder (VAE) and a U-Net backbone, adapting their vast prior knowledge to the specific domain of the industrial casting defects.

To manage computational constraints and ensure high-fidelity synthesis, the architectural pipeline is divided into a frozen feature extraction backbone and a trainable conditioning mechanism. The input RGB images ($256 \times 256 \times 3$) are first compressed by the pretrained *AutoencoderKL* into a lower-dimensional latent space. Both, the VAE and the U-Net layers are frozen during training to prevent *catastrophic forgetting* of the general image structures.

To synthesize structurally accurate industrial parts, the generation process requires strict control over both, the surface texture and the geometric shape. To achieve this, a Dual Conditioning approach is implemented:

1. **Semantic Textural Conditioning (Texture):** The morphological characteristics of the casting components (e.g., severe porosity or burrs) are captured using a frozen CLIP text encoder. These natural language prompts are transformed into dense continuous embeddings injected into the U-Net via cross-attention layers.
2. **Geometric Spatial Conditioning (Shape):** In industrial quality control, the macroscopic geometry of a manufactured part remains constant. To prevent the diffusion model from modifying the circular shape or the structural boundaries of the component, a ControlNet adapter is integrated into the pipeline Before the generation, a Canny Edge detection map is extracted from the original casting image. The ControlNet process this edge map and injects spatial residuals into the U-Net's decoder.

This dual mechanism ensures that the semantic prompts solely dictate the generation of the defects, while ControlNet rigidly preserves the macroscopic geometry of the inspected parts. The figure 4.8 illustrates the complete pipeline of the fine-tuning process.

4.4.2 Parameter-Efficient Fine-Tuning (PEFT) via LoRA

The adaptation of foundational diffusion models to specific industrial domains presents a severe computational challenge. Fully fine-tuning the U-Net backbone of SD v1.5 requires updating approximately 861 million parameters. Demanding computational requirements and prohibitive Video Random Access Memory (VRAM) allocation are not the sole challenges; this approach can also lead to catastrophic forgetting, where the model forgets its general image generation capabilities [13].

To circumvent these challenges, this work implements PEFT via LoRA. The mathematical foundation of LoRA relies on the hypothesis that the parameter updates (ΔW) during domain adaptation occupy a low-rank subspace with a significantly lower intrinsic dimension than the original weight matrix (W_0) [13].

For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$ within the U-Net's attention layers, the full gradient update is bypassed. Instead, the update matrix ΔW is decomposed into the product of two low-rank matrices, $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ [13]:

$$W = W_0 + \Delta W = W_0 + \frac{\alpha_{\text{LoRA}}}{r} BA \tag{4.3}$$

where the rank r is tightly constrained such that $r \ll \min(d, k)$ [13]. During the forward pass, the input x is multiplied in parallel by the frozen pre-trained weights and the low-rank adapters; consequently, the modified activation is given by [13]:

$$h = W_0 x + \Delta W x = W_0 x + \frac{\alpha_{\text{LoRA}}}{r} BAx \tag{4.4}$$

In this implementation, the rank is set to $r = 8$ and the scaling factor to $\alpha_{\text{LoRA}} = 16$. The matrices are initialized utilizing a random Gaussian distribution for A and zeros for B, ensuring that $\Delta W = 0$ initially to preserve the prior knowledge of the model [13].

LoRA adapters are injected into the cross-attention layers of the U-Net backbone—specifically into the *query*, *key*, *value*, and *output* projection matrices [13]. By restricting backpropagation exclusively to these low-rank adapters—while keeping the VAE, CLIP, ControlNet, and the rest of the U-Net strictly frozen—the number of trainable parameters is drastically reduced from 861,115,332 to 1,594,368. This results in an optimization of only 0.1852% of the entire network, enabling the localized high-frequency texture synthesis of severe casting defects on standard consumer GPU hardware.

The figure 4.7 illustrates the LoRA architecture used in this work.

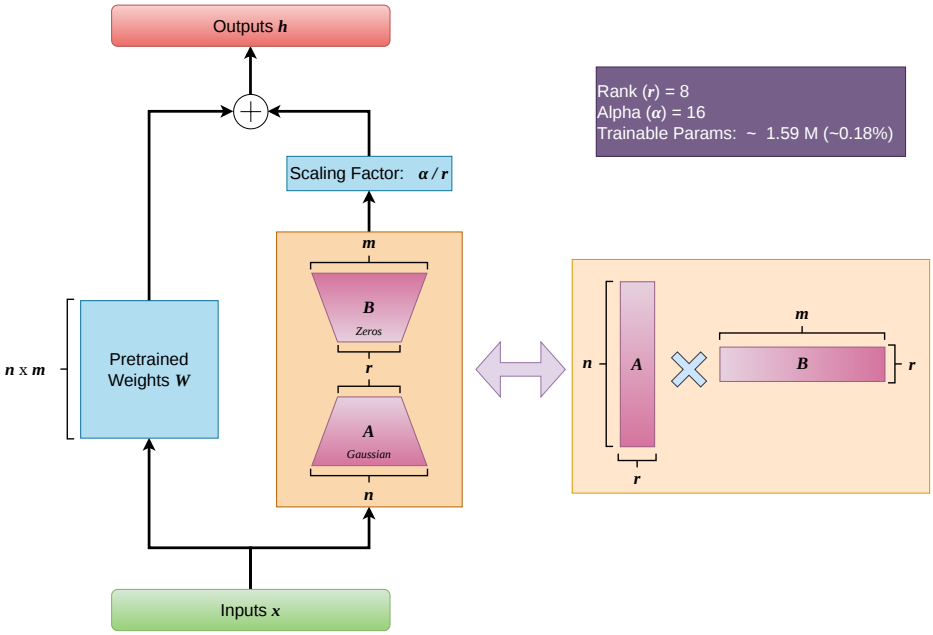


Figure 4.7: Low-Rank Adaptation (LoRA) Architecture.

4.4.3 Training Protocol

The training loop for the LoRA-adapted LDM was designed to optimize the injected attention matrices over $E = 50$ epochs. The optimization relies on a DDPM scheduler to control the forward noise injection—diffusion process—and guide the reverse denoising predictions. The protocol strictly adheres to the following computational sequence for each mini-batch of 8 images:

1. Latent Encoding and Forward Diffusion

The input RGB images ($256 \times 256 \times 3$ normalized to $[-1, 1]$) are compressed by the frozen VAE's Encoder into latent representations. These latents are multiplied by the VAE's intrinsic scaling factor to ensure a unit variance distribution, a mathematical prerequisite for the diffusion process. Subsequently, a noise tensor $\epsilon \sim \mathcal{N}(0, I)$ is sampled and added to the scaled latents based on a randomly selected timestep t producing the noisy latent z_t. In parallel, the descriptive text prompts are tokenized and processed by the frozen CLIP text Encoder to generate semantic embeddings (c_{text}).

2. Dual Conditioned Forward Pass

The forward pass is divided to accommodate the dual conditioning strategy. First, the frozen ControlNet receives the noisy latents z_t, the timestep t, the semantic text embeddings c_{text}, and the corresponding Canny edge map (c_{canny} normalized to $[0, 1]$). ControlNet processes the spatial features and outputs a set of intermediate residuals.

The U-Net evaluates the exact same noisy latents z_t, timestep t and the semantic embeddings c_{text} immediately after. However, the internal activations are dynamically altered by the trainable LoRA matrices in its cross-attention layers and the spatial residuals injected from the ControlNet. The U-Net is guided by these combined conditions to yield a prediction of the original noise tensor, denoted as $\epsilon_\theta(z_t, t, c_{text}, c_{canny})$.

3. Loss Computational and Gradient Update

The objective function is defined as the MSE between the Gaussian noise ϵ added during the forward diffusion process and the predicted noise ϵ_θ generated by the U-Net:

$$\mathcal{L}_{LDM} = \mathbb{E}_{z_0,\epsilon,t,c_{text},c_{canny}} \left[\|\epsilon - \epsilon_\theta(z_t, t, c_{text}, c_{canny})\|_2^2 \right] \tag{4.5}$$

The calculated loss is backpropagated through the parameter-efficient LoRA matrices only. The optimization is carried out with the AdamW optimizer with a decoupled weight decay mechanism and static learning rate of 1×10^{-4}.

Periodic Validation Synthesis

To visually validate the synthesis quality and ensure the model was not overfitting the noise distribution at the expense of structural accuracy, a validation pipeline was executed every 5 epochs. The training gradients were temporarily detached, and the updated U-Net was coupled with the frozen VAE Decoder. Using a static geometric Canny edge map and fixed defect prompt, the Reverse Markov Chain denoising process was initiated for 20 inference steps with a Classifier-Free Guidance (CFG) scale of 7.5. This allowed continuous visual verification of the texture generation against the structural boundaries.

In the figure 4.8 can be observed the training and validation pipelines.

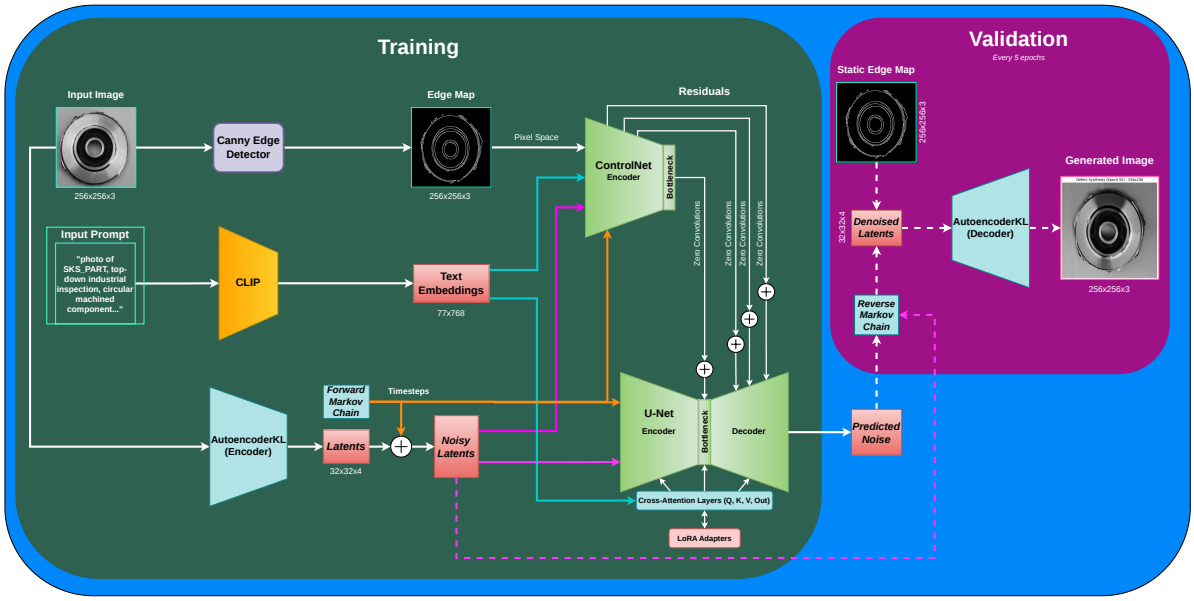


Figure 4.8: Diffusion Model Architecture During the Training and Validation Stages.

4.4.4 Training Algorithm

Algorithm 2 SD-LoRA-ControlNet Training Protocol

Require: Training dataset $\mathcal{D} = \{(x^{(i)}, c_{canny}^{(i)}, c_{text}^{(i)})\}_{i=1}^{N}$
Require: Base Models (Frozen): VAE Encoder E_{VAE}, VAE Decoder D_{VAE}, CLIP E_{text}, ControlNet C_{net}
Require: Base Model (Trainable): U-Net ϵ_θ
Require: Hyperparameters: Total epochs $E = 50$, Batch size $B = 8$, Learning rate $\eta = 1 \times 10^{-4}$
Require: DDPM Scheduler with noise timestep T, VAE scaling factor s_{VAE}
Require: Validation frequency $V_{freq} = 5$, Validation steps $S_{val} = 20$, CFG scale $\omega = 7.5$

Initialization:
Freeze parameters of $E_{VAE}, D_{VAE}, E_{text}, C_{net}$
Inject LoRA matrices (A, B) with rank $r = 8$ into ϵ_θ cross-attention layers (`to_q, to_k, to_v, to_out.0`)
Initialize AdamW optimizer with η for LoRA parameters only
for *epoch* $= 1, 2, \ldots, E$ **do**
    for mini-batch $(x, c_{canny}, c_{text}) \subset \mathcal{D}$ **do**
        // 1. Latent Encoding
        $z_0 \leftarrow E_{VAE}(x) \times s_{VAE}$ ▷ Compress and scale RGB image
        $e_{text} \leftarrow E_{text}(c_{text})$ ▷ Extract CLIP text embeddings
        // 2. Forward Diffusion (Noise Injection)
        Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
        Sample timestep $t \sim \text{Uniform}(1, T)$
        $z_t \leftarrow \text{AddNoise}(z_0, \epsilon, t)$ ▷ Apply DDPM forward Markov chain
        // 3. Dual Conditioned Forward Pass
        $R_{down}, R_{mid} \leftarrow C_{net}(z_t, t, e_{text}, c_{canny})$ ▷ Extract geometric residuals
        $\hat{\epsilon} \leftarrow \epsilon_\theta(z_t, t, e_{text}, R_{down}, R_{mid})$ ▷ Predict noise via LoRA-adapted U-Net
        // 4. Loss Computation and Optimization
        $\mathcal{L}_{MSE} \leftarrow \frac{1}{B} \sum_{j=1}^{B} ||\epsilon^{(j)} - \hat{\epsilon}^{(j)}||_2^2$
        $\nabla \mathcal{L}_{MSE} \leftarrow$ Backpropagate exclusively through LoRA matrices
        Update LoRA weights using AdamW optimizer
    end for
    // 5. Periodic Validation Synthesis
    if *epoch* $(\text{mod } V_{freq}) == 0$ **or** *epoch* $== E$ **then**
        Load fixed validation variables: c_{canny}^{val}, c_{text}^{val}, $z_T \sim \mathcal{N}(0, \mathbf{I})$
        for $t = T, T-1, \ldots, 1$ in S_{val} steps **do** ▷ Reverse Markov Chain
            $\hat{\epsilon} \leftarrow \epsilon_\theta(z_t, t, e_{text}^{val}, C_{net}(z_t, t, e_{text}^{val}, c_{canny}^{val}))$
            Apply Classifier-Free Guidance with scale $\omega = 7.5$
            $z_{t-1} \leftarrow \text{DenoiseStep}(z_t, \hat{\epsilon}, t)$
        end for
        $x_{gen} \leftarrow D_{VAE}(z_0 / s_{VAE})$ ▷ Decode to RGB space
    end if
end for
return Optimized LoRA weights for ϵ_θ

4.5 Evaluation Metrics and Validation Protocol

The validation framework for assessing the generative models (AC-CVAE-GAN and SD-LoRA-ControlNet) was designed to ensure a robust and reproducible evaluation, relying on the established metrics FID and LPIPS. This assessment measured both the distributional fidelity and the perceptual high-frequency realism of the generated casting defects against the real industrial distribution. The validation pipeline was executed under the following protocol:

1. **Data Standardization:** To establish the real distribution, the images from the original casting dataset (`ok_front` and `def_front`) were resized from their original 512×512 resolution to 256×256 pixels serving as the statistical ground truth.
2. **Mass Generation Strategy:** To ensure a statistically robust evaluation and avoid rank-deficient covariance matrices during FID calculation, a mass generation target was set to $\mathcal{N} = 1000$ synthetic images per class for both the CVAE-GAN and SD architectures.
3. **Iterative Generation:** Since the number of available real images (and their derived Canny maps) was lower than the generation target, an iterative cycling strategy was adopted. The generation process cycled through the available real reference images; however, for every repetition a unique stochastic seed was used to generate a unique synthetic instance resulting in unique high-frequency textures ensuring statistical diversity.
4. **Metric Computation:** The evaluations were performed using the `torchmetrics` library. The FID was calculated extracting the 2048-dimensional feature vectors from the final pooling layer of a pre-trained Inception-V3 network. On the other hand, the LPIPS was calculated utilizing the activations of the VGG16 network to evaluate the textural coherence of the high-frequency details.

4.6 Code and Data Availability

To ensure the reproducibility of the experiments presented in this thesis, the complete implementation, including the generative architectures and data pipelines, has been made open-source. The source code, pre-trained weights, and detailed instructions are hosted in the official GitHub repositories, which can be accessed at:

- `https://github.com/StevoAngel/CVAE-GAN-For-Defect-Synthesis`
- `https://github.com/StevoAngel/Stable-Diffusion-For-Defect-Synthesis`

5. RESULTS AND DISCUSSION

5.1 Results

5.1.1 Quantitative Evaluation

This section presents the quantitative results of the AC-CVAE-GAN and the LDM (SD-LoRA-ControlNet) architectures. The models were evaluated using the FID to measure the distributional fidelity and the LPIPS to evaluate the structural and perceptual coherence. This evaluation was conducted on a generated sample size of 1,000 images per class (`ok_front` and `def_front`), compared against the real dataset to ensure statistical stability.

The quantitative results are summarized in table 5.1.

Table 5.1: Quantitative performance metrics of the generative architectures.

Architecture	**Generation Class**	**FID** (↓)	**LPIPS** (↓)
AC-CVAE-GAN	OK	209.36	0.4619
AC-CVAE-GAN	Defect	180.72	0.4671
SD-LoRA-ControlNet	OK	67.65	0.2590
SD-LoRA-ControlNet	Defect	96.17	0.4322

AC-CVAE-GAN Performance

The adversarial model obtained an FID score of 209.36 for the nominal class and 180.72 for the defect class. Quantitatively, the architecture achieved a lower (better) distributional distance when generating defective parts compared to nominal (flawless) geometries. In terms of perceptual similarity, the LPIPS scores remained consistent across both generative tasks, scoring 0.4619 for nominal and 0.4671 for the defect class.

SD-LoRA-ControlNet Performance

The LDM architecture achieved a contrasting numerical behaviour. The flawless class obtained an FID score of 67.65 alongside the lowest LPIPS score in the experiment (0.2590). For the defective class, the FID metric achieved a slight degradation to 96.17, while the LPIPS score experienced a more significant degradation to 0.4322.

Overall, the diffusion-based model outperformed the adversarial model registering strictly lower FID scores across both classes, establishing a new quantitative threshold for the dataset.

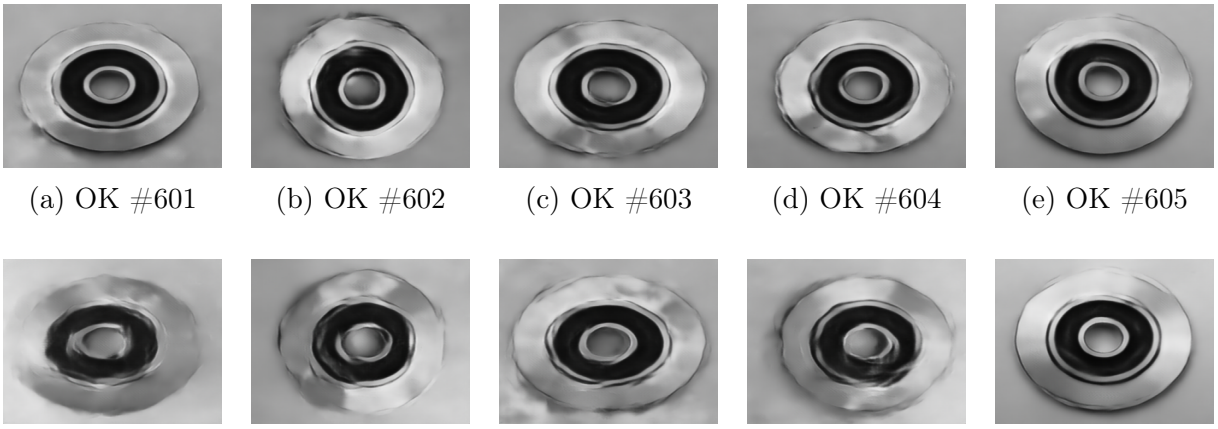
5.1.2 Qualitative Generation Outcomes

This section presents a visual inspection of the synthesized images from both generative architectures.

AC-CVAE-GAN Visual Assessment

Visual inspection of the AC-CVAE-GAN synthesized images reveals significant limitations in synthesizing high-frequency details. For the defect-conditioned class, the model struggled to generate distinct defect artifacts—such as porosity. The majority of these images presented severe blurriness, geometric deformations, and in general, a lack of structural definition.

On the other hand, images belonging the nominal class (OK) showed a notable improvement in visual quality and definition. However, blur artifacts and malformations remained present. Furthermore, visual analysis revealed that an apparent randomness or class entanglement, since several defect-conditioned images lacked anomalies, while some nominal images presented malformations. Despite this mixing, the overall trend indicated that defect-class generation suffered from lower visual quality compared to the nominal class (see Figure 5.1).



(a) OK #601 (b) OK #602 (c) OK #603 (d) OK #604 (e) OK #605

(f) Defect #601 (g) Defect #602 (h) Defect #603 (i) Defect #604 (j) Defect #605

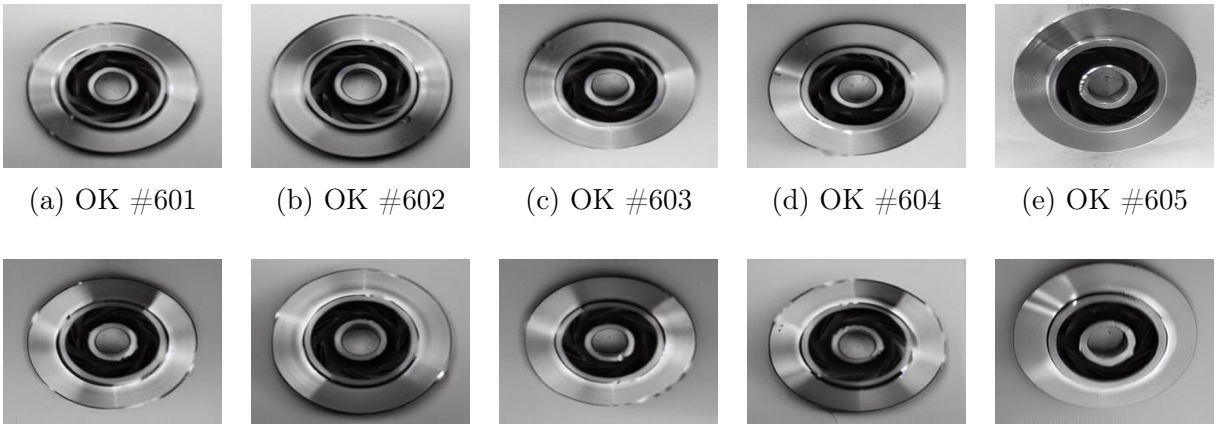
Figure 5.1: Comparative of synthetic samples generated by the AC-CVAE-GAN architecture.

Note that the top row (5.1a to 5.1e) displays the synthetic samples, while the bottom row (5.1f to 5.1j) shows the synthetic defect samples. The images have their corresponding number of the sample in between the 1000 generated samples, starting from the 601st sample. In this case, the synthetic samples are the 601st to the 605th sample.

SD-LoRA-ControlNet Visual Assessment

The diffusion-based architecture showed a vastly superior generation performance. The visual difference compared to the adversarial architecture baseline is substantial, with SD-LoRA-ControlNet producing highly defined, photorealistic, high-frequency details. When successfully synthesized, the defect artifacts—such as porosity—closely match the appearance of real industrial defects.

However, the visual inspection assessment identified two primary generation inconsistencies. First, a degree of class mixing was observed, since some images conditioned as normal contained visible defects, while certain defect-conditioned images were generated completely flawless, a fact suggesting an incomplete semantic understanding of the defect conditions by the model. Second, spatial localization of artifacts were evident. High-frequency defect textures occasionally appeared outside the expected physical boundaries—appearing in the background or near the edges of the piece—indicating the need for improved spatial conditioning mechanisms (see Figure 5.2).



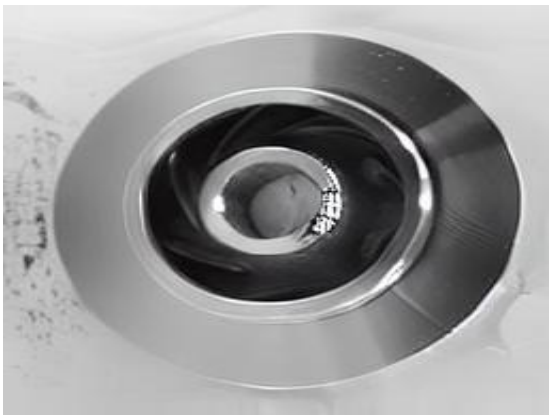
(a) OK #601 (b) OK #602 (c) OK #603 (d) OK #604 (e) OK #605

(f) Defect #601 (g) Defect #602 (h) Defect #603 (i) Defect #604 (j) Defect #605

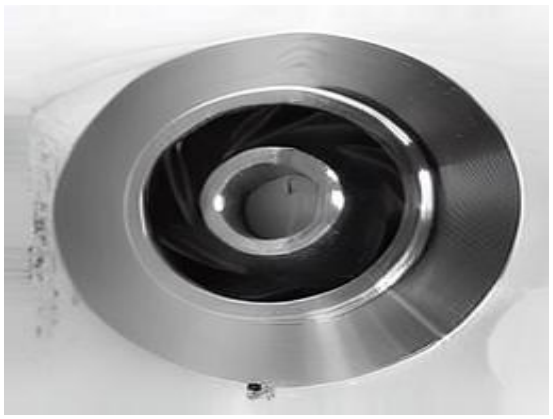
Figure 5.2: Comparative matrix of synthetic samples generated by the SD-LoRA-ControlNet architecture.

Note that the top row (5.2a to 5.2e) displays the synthetic samples, while the bottom row (5.2f to 5.2j) shows the synthetic defect samples. The images have their corresponding number of the sample in between the 1000 generated samples, starting from the 601st sample. In this case, the synthetic samples are the 601st to the 605th sample.

The image 5.3 illustrates the difference between an OK and Defect sample generated using the same seed. It is observable that despite the class condition, both images share almost the same high-frequency details as if both were defect-conditioned images. Furthermore, some of the artifacts are present out of the boundaries of the samples.



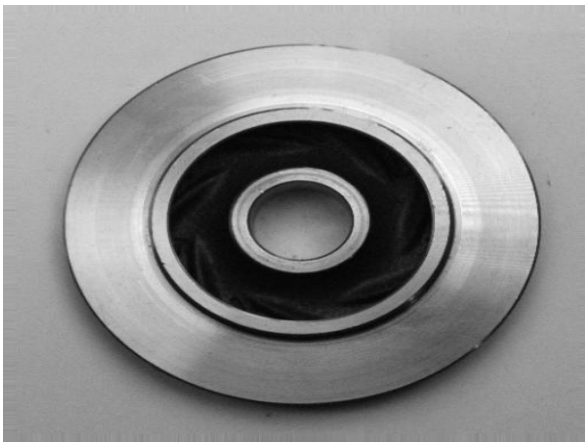
(a) OK Sample



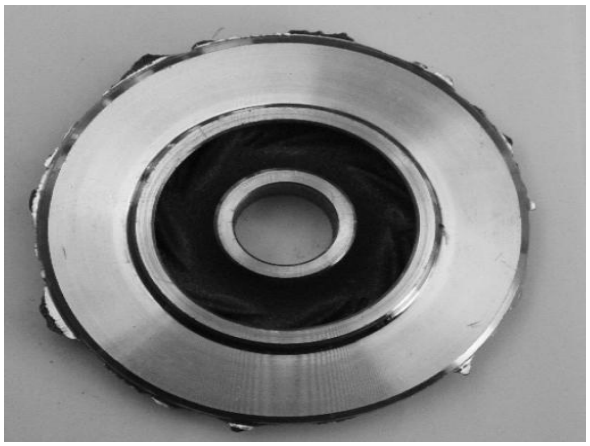
(b) Defect Sample

Figure 5.3: Image generated for the SD-LoRA-ControlNet model using the same seed for both classes (OK & Defect)

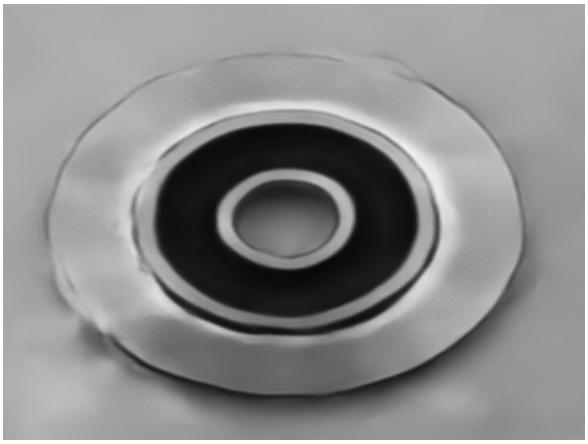
In the figure 5.4 the difference between the real samples and the synthetic samples from both models is presented.



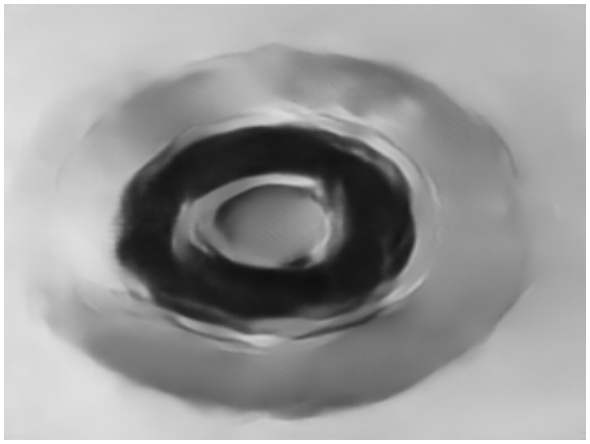
(a) Real OK Sample



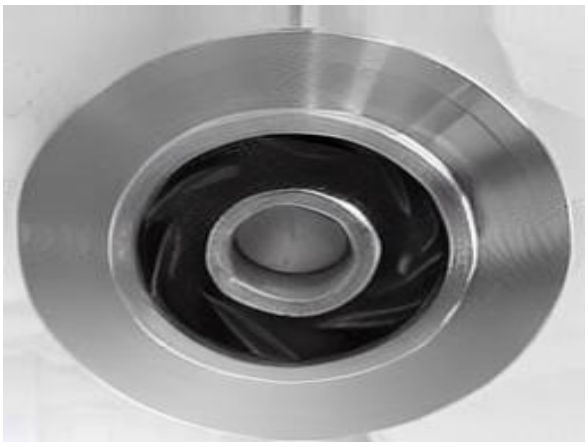
(b) Real Defect Sample



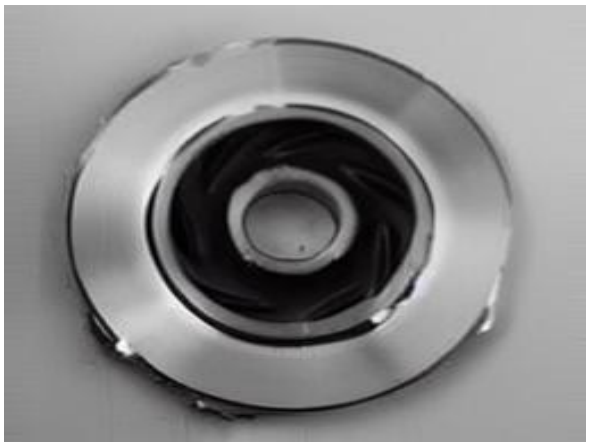
(c) AC-CVAE-GAN OK Sample



(d) AC-CVAE-GAN Defect Sample



(e) SD-LoRA-ControlNet OK Sample



(f) SD-LoRA-ControlNet Defect Sample

Figure 5.4: Comparative visualization of samples from the original dataset and synthetic samples generated by the AC-CVAE-GAN and SD-LoRA-ControlNet architectures.

5.2 Discussion

5.2.1 AC-CVAE-GAN: Latent Entanglement and Metric Contradiction

The quantitative metrics and qualitative results for the AC-CVAE-GAN present a complex contradiction. The FID score suggested the model was performing accurately at generating defects (180.72), theoretically outperforming the generation of nominal parts (209.36). However, qualitative visual inspection revealed severe blurriness—a known limitation of VAE reconstruction objectives [7]—and a failure to synthesize accurate defect structures. This behavior suggests a discriminator dominance coupled with latent space entanglement.

During the adversarial training from scratch, the high-frequency features of the defect class dominated the gradients of the discriminator network. As described by Kang et al. in their analysis of ACGAN instability [47], early-training collapse often occurs when the network concentrates on classifying categories rather than discriminating authenticity, which breaks the adversarial balance. Consequently, the generator collapsed towards generating high-frequency noise to trivially trick the discriminator [47]. Because the Inception-v3 model used to compute the FID compares feature distribution moments (mean and covariance) [21], it processed this structural noise as statistically similar to the naturally high variance of real defects. This paradoxically led to a lower—seemingly better—FID score, despite the lack of anatomical realism.

Perceptually, the model failed to learn the underlying topology of the defects. Furthermore, the observed class randomness—where nominal parts presented malformations and vice versa—indicates that the variational autoencoder's KLD constraint forced the latent distributions of both classes to overlap significantly. This phenomenon, known as posterior collapse, occurs when the optimization trivially minimizes the divergence by making the latent code completely uninformative and pushing it towards the standard normal prior [7], [48]. This resulted in an entangled latent space where the OK and Defect conditional labels lost their steering control over the generation process [48].

5.2.2 SD-LoRA-ControlNet: Spatial Dislocation and Concept Bleeding

The LDM-based architecture displayed exceptional qualitative superiority, effectively bridging the photorealistic gap required for industrial defect synthesis. The sharp definition of pores, burrs, and other high-frequency artifacts aligns with the low FID and LPIPS scores. However, class entanglement—where defects inadvertently appear in nominally conditioned samples—remained present for this model as well. Furthermore, a spatial

dislocation of these artifacts is notable, as they frequently appear outside the physical boundaries of the cast samples. These inconsistencies expose the inherent limitations of relying exclusively on global textual conditioning and basic geometric edge guidance.

The presence of identical high-frequency defects in nominal samples sharing the same initial stochastic seed suggests that the LoRA weights overfitted the specific defect concepts—such as "porosity", "burrs", and "cavities"—to the base latent noise. This phenomenon, which can be described as "concept bleeding", causes the U-Net cross-attention layers to prioritize the learned structural noise distributions over the nominal text constraints. As noted in recent literature regarding diffusion models in industrial settings, text-to-image architectures inherently struggle to interpret strict quantitative or negative constraints, often prioritizing the general learned "concept" of a defect over strict prompt adherence [38].

Additionally, the spatial dislocation of flaws occurs because the ControlNet Canny maps enforce only geometric edge boundaries rather than semantic segmentation masks. While the model successfully learned to generate highly accurate porosity, it lacks the explicit spatial constraints necessary to place these textural artifacts strictly within the designated metallic regions of the casting parts. Traditional diffusion frameworks inherently lack the capability to associate generated features with fine-grained spatial boundaries [38]. As highlighted by recent studies on ControlNet-based anomaly generation, overcoming this textural spilling effect strictly requires robust foreground segmentation masks to confine the generation process exclusively to the valid object topology [37].

5.2.3 Computational Efficiency and Industrial Scalability

Beyond the generative quality, the deployment of synthetic data into an industrial pipeline demands a rigorous evaluation of the computational efficiency and architectural flexibility. Comparing the AC-CVAE-GAN and the SD-LoRA-ControlNet frameworks reveals distinct industrial trade-offs regarding training speed, latent manipulation, and resolution scalability.

Despite the metric limitations in defect synthesis, the custom AC-CVAE-GAN demonstrated significant computational advantages. Training this model from scratch for 1,000 epochs required approximately 4 hours in the environment described in Section 4.2. The lightweight nature of the model facilitates rapid prototyping. Furthermore, a key advantage of the custom VAE backbone is the continuous and explicitly structured nature of its latent space. This architecture allows for explicit latent arithmetic, theoretically enabling the precise interpolation of the defect severity over a nominal part. While the model presented in this work struggles with high-frequency clarity, its underlying morphology offers a level of deterministic mathematical control over the defect spectrum that diffusion models inherently lack.

On the other hand, the SD-LoRA-ControlNet pipeline, despite utilizing PEFT, represents

a substantially higher computational cost. Achieving an acceptable convergence required only 50 epochs; however, this brief training phase consumed approximately 3.5 hours due to the massive parameter count of the foundational U-Net and the CLIP text encoder. The LDM latent space is also highly non-linear and entangled, making continuous defect arithmetic less intuitive. Nevertheless, this framework provides a crucial operational advantage: zero-shot resolution scaling. As demonstrated by Rombach et al. [10], the fully convolutional architecture of latent diffusion models allows them to be evaluated beyond their training dimensions. The diffusion architecture demonstrated the capability to be trained on 256×256 inputs while successfully inferring and synthesizing high-fidelity images at larger resolutions (e.g., 320×320), often yielding improved perceptual quality without requiring architectural modifications (see Figure 5.5).

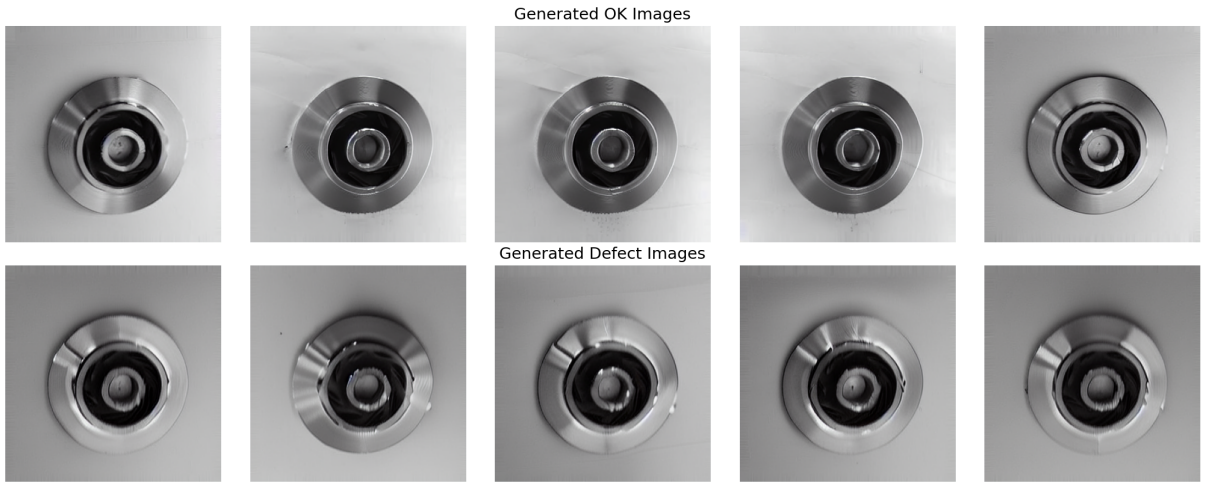


Figure 5.5: 320x320 Diffusion-Generated Images

Ultimately, the choice between the AC-CVAE-GAN and the SD-LoRA-ControlNet depends on the industrial case. The first option provides a high efficiency and a mathematically manipulable latent space, suited for continuous defect interpolation. The second option offers computationally expensive but structurally photorealistic synthesis with a robust resolution scalability.

6. CONCLUSIONS

6.1 Conclusions

This research evaluated the efficacy of two distinct generative architectures—a custom AC-CVAE-GAN and a LDM (SD-LoRA-ControlNet)—for the synthesis of high-fidelity industrial casting defects in metallic submersible pump impellers. This analysis demonstrated that the SD-LoRA-ControlNet model architecture establishes a superior standard photorealistic, high-frequency anomaly generation. The success of this model relies on the architectural decoupling of structural geometry and stochastic textures. By leveraging ControlNet to strictly enforce the rigid, metallic boundaries of the impellers, and applying LoRA to inject localized defect patterns, the model successfully generated defects without compromising the structural integrity of the casting parts.

However, the analysis also revealed that the AC-CVAE-GAN remains a highly relevant and competitive architecture for industrial applications, particularly when the computational efficiency and latent determinism are important factors. While the adversarial network struggle to resolve high-frequency clarity when trained on binary class distribution, its lightweight architecture enables exceptionally fast training convergence, requiring only a fraction of the resources needed for a diffusion model of comparable resolution. Furthermore, the encoder-decoder structure of the VAE provides an explicit, structured latent space, offering a level of deterministic control over defect characteristics that is difficult to achieve in the non-linear latent manifold of

Ultimately, the development of these synthetic data generator models addresses a critical bottleneck in modern industrial manufacturing. Collecting physical data for rare, severe defects on a production line is prohibitively expensive and time-consuming. The generative pipelines validated in this work, provide a scalable, software-driven solution to this challenge. By enabling the mass generation of highly accurate, defective datasets, these models facilitate the robust training of automated visual inspection systems directly contributing to more resilient, automated, and cost-effective industrial manufacturing environments.

6.2 Future Work

While the current research establishes a robust foundation for industrial defect synthesis, several directions remain for optimizing both the computational efficiency and the deterministic accuracy of these generative pipelines. Future iterations of this work should focus on the following areas:

- **Dataset Granularity and Multi-Class Conditioning**: The visual blurriness and latent collapse observed in the AC-CVAE-GAN can be partially attributed to the binary nature of the dataset (`ok_front` and `def_front`). Grouping diverse,

chaotic anomalies into a single macro-class forces the VAE to average the incompatible high-frequency details. Future work should involve segmenting and labeling the dataset into specific defect typologies (e.g., pores, cracks, misruns) to enable fine-grained conditional synthesis. By conditioning the AC-CVAE-GAN on granular sub-classes, the latent space could achieve a higher disentanglement potentially improving its visual generation quality to compete with SD-based models while retaining its computational speed.

- **Semantic Masking for Spatial Constraints**: To resolve the spatial localization of failures (texture bleeding) observed in the SD-LoRA-ControlNet model, future implementations should include semantic segmentation masks alongside Canny edge maps. While edge detection preserves the structural boundaries of the casting part, binary semantic masks could provide additional guidance to the U-Net to preserve the texture of the defect localized within the desired region preventing artifacts from appearing in the image background.

- **Text-Encoder(CLIP) Fine-Tuning**: To mitigate the class entanglement observed in the SD-LoRA-ControlNet model—where the nominal parts occasionally present defects—future work should explore fine-tuning the foundational CLIP text encoder in parallel with the U-Net LoRA weights. Improving the text-to-image semantic alignment would ensure strict adherence to the generation prompts, virtually eliminating stochastic class mismatches.

- **Explainable Artificial Intelligence (XAI) for Qualitative Evaluation**: Finally, the subjective nature of qualitative visual assessments should be replaced by mathematically interpretable metrics. Future validation protocols should incorporate XAI techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) to audit the decision-making process of the models and detect potential biases in the adversarial discriminator. Similarly, extracting Diffusion Attentive Attribution Maps (DAAM) from the cross-attention layers of the U-Net would provide objective mathematical evidence of semantic grounding, ensuring that the network accurately links the textural concept of a defect to the exact spatial pixels it modifies.

REFERENCES

[1] P. Bergmann et al., "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. DOI: 10.1109/CVPR.2019.00982

[2] J. E. See, "Visual Inspection Reliability for Precision Manufactured Parts," *Human Factors*, vol. 57, 2015. DOI: 10.1177/0018720815602389

[3] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, vol. 54, no. 2, 2022. DOI: 10.1145/3439950

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, 2017. DOI: 10.1145/3065386

[5] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, 2019. DOI: 10.1186/s40537-019-0197-0

[6] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6114

[7] S. Zhao, J. Song, and S. Ermon, "Towards Deeper Understanding of Variational Autoencoding Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. [Online]. Available: https://arxiv.org/abs/1702.08658

[8] I. Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Networks," *arXiv preprint arXiv:1701.00160*, 2016. [Online]. Available: https://arxiv.org/abs/1701.00160

[9] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. DOI: 10.1109/ICCV.2017.299 [Online]. Available: https://arxiv.org/abs/1703.10155

[10] R. Rombach, 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 (CVPR)*, 2022, pp. 10 684–10 695. [Online]. Available: https://arxiv.org/abs/2112.10752

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6840–6851. [Online]. Available: https://arxiv.org/abs/2006.11239

[12] L. Zhang, 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 (ICCV)*, 2023, pp. 3813–3824. DOI: 10.1109/ICCV51070.2023.00355 [Online]. Available: https://arxiv.org/abs/2302.05543

[13] E. J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2106.09685

[14] R. Dabhi, *Casting Product Image Data for Quality Inspection*, Kaggle Dataset, Accessed: 2026-04-04, 2020. [Online]. Available: https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product

[15] P. M. Bhatt et al., "Image-Based Surface Defect Detection Using Deep Learning: A Review," *Journal of Computing and Information Science in Engineering*, vol. 21, 2021. DOI: 10.1115/1.4049535

[16] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, 2004. DOI: 10.1023/B:VISI.0000029664.99615.94

[17] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. DOI: 10.1109/CVPR.2005.177

[18] K. Sohn, X. Yan, and H. Lee, "Learning Structured Output Representation using Deep Conditional Generative Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[19] Z. Du, L. Gao, and X. Li, "A New Contrastive GAN With Data Augmentation for Surface Defect Recognition Under Limited Data," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, 2023. DOI: 10.1109/TIM.2022.3232649

[20] T. Salimans, I. Goodfellow, et al., "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. [Online]. Available: https://arxiv.org/abs/1606.03498

[21] M. Heusel, 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 (NeurIPS)*, vol. 30, 2017. [Online]. Available: https://arxiv.org/abs/1706.08500

[22] R. Zhang, 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 (CVPR)*, 2018, pp. 586–595. DOI: 10.1109/CVPR.2018.00068 [Online]. Available: https://arxiv.org/abs/1801.03924

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[24] A. Radford, J. W. Kim, et al., "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[25] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[26] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, 1986. DOI: 10.1109/TPAMI.1986.4767851

[27] R. Wang, S. Hoppe, E. Monari, and M. F. Huber, "Defect Transfer GAN: Diverse Defect Synthesis for Data Augmentation," *arXiv preprint arXiv:2302.08366*, 2023. [Online]. Available: https://arxiv.org/abs/2302.08366

[28] M. Razghandi, H. Zhou, M. Erol-Kantarci, and D. Turgut, "Variational Autoencoder Generative Adversarial Network for Synthetic Data Generation in Smart Home," *IEEE Access*, vol. 10, pp. 24 522–24 536, 2022. DOI: 10.1109/ACCESS.2022.3155304

[29] G. Zhang, K. Cui, T.-Y. Hung, and S. Lu, "Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[30] F. Deng et al., "DG2GAN: Improving Defect Recognition Performance with Generated Defect Image Sample," *Scientific Reports*, vol. 14, no. 1, 2024. DOI: 10.1038/s41598-024-64716-y

[31] H. Bu, T. Yang, C. Hu, X. Zhu, Z. Ge, and H. Zhou, "An Image Classification Method of Unbalanced Ship Coating Defects Based on DCCVAE-ACWGAN-GP," *Coatings*, vol. 14, no. 3, 2024. DOI: 10.3390/coatings14030288

[32] G. Ma, J. Zhang, and J. Jiang, "Few-Shot Surface Defect Detection in Sinusoidal Wobble Laser Welds Using StyleGAN2-AFMS Augmentation and YOLO11n-WAFE Detector," *Automation*, vol. 7, no. 2, p. 38, 2026. DOI: 10.3390/automation7020038

[33] L. Capogrosso et al., "Diffusion-based Image Generation for In-distribution Data Augmentation in Surface Defect Detection," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2024, pp. 409–416.

[34] T. Hu et al., "AnomalyDiffusion: Few-Shot Anomaly Image Generation with Diffusion Model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 8526–8534.

[35] R. Xu, Y.-T. Chiu, T.-I. Chen, O. Chew, Y.-Y. Chuang, and W.-H. Cheng, "Training-Free Industrial Defect Generation with Diffusion Models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[36] Y. Liu et al., "GroundingAnomaly: Spatially-Grounded Diffusion for Few-Shot Anomaly Synthesis," *arXiv preprint arXiv:2604.08301*, 2026. [Online]. Available: https://arxiv.org/abs/2604.08301

[37] M. Ali, N. Fioraio, S. Salti, and L. Di Stefano, "AnomalyControl: Few-Shot Anomaly Generation by ControlNet Inpainting," *IEEE Access*, vol. 12, pp. 192 903–192 914, 2024. DOI: 10.1109/ACCESS.2024.3520002

[38] H. Li, Y. Liu, C. Liu, H. Pang, and K. Xu, "A Few-Shot Steel Surface Defect Generation Method Based on Diffusion Models," *Sensors*, vol. 25, no. 10, 2025. DOI: 10.3390/s25103038

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[40] C. Szegedy, W. Liu, Y. Jia, et al., "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: https://arxiv.org/abs/1409.4842

[41] K. He, 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.

[42] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006, ISSN: 0036-8075. DOI: 10.1126/science.1127647 [Online]. Available: https://www.science.org/doi/10.1126/science.1127647

[43] C. Doersch, "Tutorial on Variational Autoencoders," *arXiv preprint arXiv:1606.05908*, 2016. [Online]. Available: https://arxiv.org/abs/1606.05908

[44] I. J. Goodfellow et al., "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[45] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. [Online]. Available: https://arxiv.org/abs/1511.06434

[46] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis With Auxiliary Classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. [Online]. Available: https://arxiv.org/abs/1610.09585

[47] M. Kang et al., "Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [Online]. Available: https://arxiv.org/abs/2111.01118

[48] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging Inference Networks and Posterior Collapse in Variational Autoencoders," *arXiv preprint arXiv:1901.05534*, Jan. 2019. [Online]. Available: https://arxiv.org/abs/1901.05534