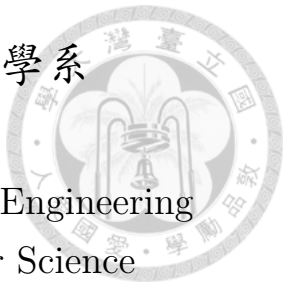


國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

資料缺乏下多樣且眞實的影像生成

Diverse and Fidelity Image Synthesis for Unsupervised Image-to-Image Translation with Limited Data

林揚昇

Yang-Sheng Lin

指導教授：許永眞博士

Advisor: Jane Yung-jen Hsu, Ph.D.

中華民國 112 年 8 月

August, 2023

doi:10.6342/NTU202303360

誌謝

非常感謝我的指導教授許永眞老師，對於我各種天馬行空的想法，總是用鼓勵的態度來支持我勇於嘗試，從不侷限我的研究創意和想法，並且在我研究遇到瓶頸時，也會幫我整理和分析現在的問題並指引我方向，最後在撰寫論文時，也與我分享了許多相關的經驗，讓我能順遂的完成這篇論文。

再來要感謝智淵學長，學長指出了我論文中缺乏的實驗，給我許多用詞、文法上的修改建議，並且還不厭其煩地教導我如何把研究投稿至國際會議，在修改論文的這段時間裡，從學長身上學習到非常多。

感謝我的口試委員老師，郭彥伶老師、鄭文皇老師、陳駿丞博士和楊智淵博士抽空來參與我的口試，並點出了許多我論文中需要補足的部分，有你們的建議，這篇論文才能更加完善。

謝謝敦捷、明毅和實驗室的夥伴，跟你們一起努力是讓我堅持下去的動力之一，並且在互相討論過程中也激發了我許多的靈感。

最後，感謝我的家人，和一路上幫助我的朋友們，有你們的支持和幫助，我才能順利完成我的學業。

doi:10.6342/NTU202303360

Abstract

Unsupervised Image-to-Image Translation (Unsupervised I2I) has emerged as a significant area of interest and has recently seen substantial advancements due to its wide range of applications and reduced data annotation requirements. However, in scenarios with limited data, ensuring training stability and generating diverse, realistic images remain critical research directions. To address these challenges, we propose two simple, plug-and-play methods: the Masked AutoEncoder Generative Adversarial Network (MAE-GAN) and the Style Embedding Adaptive Normalization (SEAN) block.

The MAE-GAN, a pre-training method for Unsupervised I2I tasks, integrates the architectures and strengths of both MAE and GAN. It also enhances learning style-specific information during pre-training, leading to stable training and improved image quality in downstream tasks. The SEAN block is a novel normalization block that leverages large-scale pre-trained feature extractors and self-learns the style feature space for each domain in each layer. Consequently, it allows for a choice between diversity and fidelity, enabling the generation of more diverse or realistic images.

Our method achieves substantial success on the less common and challenging concrete defect bridge dataset (CODEBRIM), demonstrating its real-world applicability. Additionally, our methods, trained on just 10% of the Animal Faces HQ dataset (AFHQ), achieve image quality on par with models trained on the full dataset, while also reaching greater image diversity, proving its real-world applicability and immense potential.

Keywords: *Unsupervised Image-to-Image Translation, Multiple Domain Image-to-Image Translation, Data-Efficient Generative Adversarial Network, Masked Autoencoder*

doi:10.6342/NTU202303360

摘要

非監督的圖像到圖像轉換（Unsupervised Image-to-Image Translation）因其廣泛的應用範疇與不需要標註的特性，已成爲圖像生成領域的研究重心之一並獲得了非常顯著的成果。然而，在資料有限的情況下，確保訓練的穩定性並產生多樣且眞實的圖像仍是很困難的研究問題。爲了解決這些挑戰，我們提出了兩種簡單且即插即用的方法：遮罩自動編碼器生成對抗網絡（MAE-GAN）和風格嵌入自適應歸一化塊（SEAN）。

MAE-GAN是一種用於非監督圖像到圖像（Unsupervised I2I）任務的預訓練方法，它融合了MAE和GAN的架構和優點，並且在預訓練期間能學習到不同領域的風格信息，從而使下游任務的訓練穩定性和圖像品質提高。SEAN塊是一種新的歸一化塊 (Normalization Block)，它利用了大規模的預訓練特徵提取器 (Large-scale Pre-trained Feature Extractor) ，並在模型的每一層中能各自學習每個不同領域的風格特徵空間。並且，它還能在多樣性和保眞度之間進行選擇，使得可以生成更多樣化或更眞實的圖像。

我們的方法在資料型態較少見且具有挑戰性的混凝土缺陷橋樑圖像數據集（CODEBRIM）上取得了非常好的成果，此外，我們的方法也使用 10% 動物臉部數據集（AFHQ）進行訓練，達到了與原本訓練在完整數據集上的模型相進的圖像品質，並且還能獲得更好的圖像多樣性，證明了其在現實世界中的應用性和巨大的潛力。

doi:10.6342/NTU202303360

Contents

doi:10.6342/NTU202303360

doi:10.6342/NTU202303360

List of Tables

doi:10.6342/NTU202303360

List of Figures

doi:10.6342/NTU202303360

Chapter 1

Introduction

In this chapter, we begin with introducing the background of the thesis. Then our motivation and the solution we proposed are given. Finally, we will list our contributions and outline our thesis.

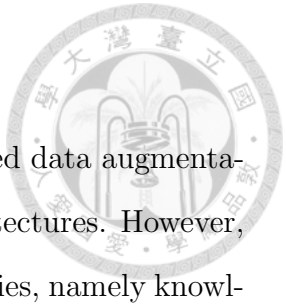
1.1 Background

In recent years, the field of image generation has made significant progress, with Unsupervised Image-to-Image Translation (Unsupervised I2I) emerging as a key area of interest. This is largely due to its wide-ranging applications and the reduced need for data annotation. However, in practical applications, collecting sufficient data to train Unsupervised I2I models is often challenging.

Furthermore, in situations with limited data, the diversity of generated images is often compromised to ensure the successful production of usable images. Therefore, ensuring the stability of generative model training and producing realistic and diverse images with limited data has become a critical research topic.

doi:10.6342/NTU202303360

1.2 Motivation



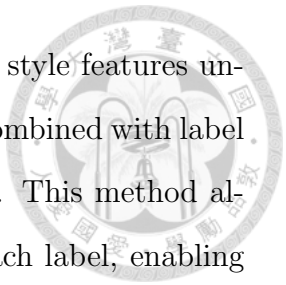
To address these challenges, recent works [1, 2] have employed data augmentation techniques that can be easily applied to different model architectures. However, in the same task of data-efficient generation, the other two categories, namely knowledge sharing and model architecture, have not been extensively explored, or they may have more restrictions, making them less applicable in practical scenarios. Interestingly, Previous research [3] suggests that these three methods can complement each other. Motivated by this, our work is dedicated to creating a simple and universal method focusing on knowledge sharing and model architecture. We hope our method can ensure stable training, generate realistic images, and maintain image diversity even in situations with limited data.

1.3 Proposed Method

We first propose Masked AutoEncoder Generative Adversarial Network (MAE-GAN), a novel pre-training method for Unsupervised Image-to-Image (I2I) tasks. It integrates the Masked Autoencoder (MAE) architecture with Generative Adversarial Networks (GANs), leveraging the adversarial nature of GANs to address the blurred image generation often associated with MAE and simultaneously trains the generator and discriminator within the GAN framework. During pre-training, an additional style input is incorporated, enabling the generator to repair masks and acquire style-related information. This method can be easily integrated with various Unsupervised I2I models and training methodologies, enhancing learning style-specific information during the pre-training phase and improving the quality of images generated in downstream tasks.

Another method we proposed is the Style Embedding Adaptive Normalization (SEAN) block, which leverages large-scale pre-trained feature extractor to enhance the diversity of images. It takes as input several style features obtained from style

doi:10.6342/NTU202303360



images processed through a pre-trained feature extractor. These style features undergo a simple dimensionality reduction, are averaged, and then combined with label embedding to generate the required style code for the generator. This method allows the model to self-learn the distribution of style codes for each label, enabling the acquisition of the desired style code directly from the label without needing a style image referencing. By adjusting the standard deviation of the distribution, a balance can be achieved between the diversity and fidelity of the generated images. Ultimately, this significantly enhances the diversity of generated images when data is limited.

1.4 Result

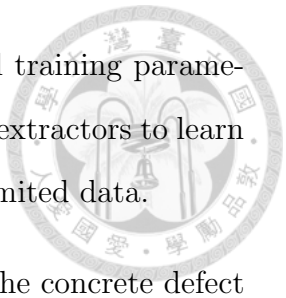
Our experiments primarily focus on a less-explored COncrete DEfect BRidge IMage (CODEBRIM) dataset, demonstrating the feasibility of our method in less common real-world data environments. Additionally, on the widely used Animal Faces-HQ (AFHQ) dataset, our methods can generate high-quality images comparable to the baseline produced with 100% of the data while only using 10% of the data. This indicates a significant improvement in data efficiency. Our methods also yield competitive results compared to state-of-the-art approaches. Notably, our methods produce higher image diversity, a critical aspect in image generation tasks.

1.5 Contribution

Our work contributes to the field of Data-Efficient Unsupervised Image-to-Image Translation as follows:

- We've developed a new framework that combines MAE and GAN for pre-training I2I's Generator and Discriminator. This approach stabilizes GAN training and enhances image quality.

doi:10.6342/NTU202303360



- We have designed a new normalization block with minimal training parameters. This block can leverage large-scale pre-trained feature extractors to learn style codes, thereby enhancing image diversity even with limited data.
- Our method effectively handles less-explored datasets like the concrete defect bridge, demonstrating its real-world applicability. It also competes well with state-of-the-art approaches on the popular AFHQ dataset, highlighting its broad generalizability.

1.6 Outline of this Thesis

The rest of the paper is organized as follows: Section 2 provides a literature review, Section 3 describes our proposed methodology, Section 4 presents our experiments and results, and Section 5 concludes the paper and discusses future work.

doi:10.6342/NTU202303360

Chapter 2

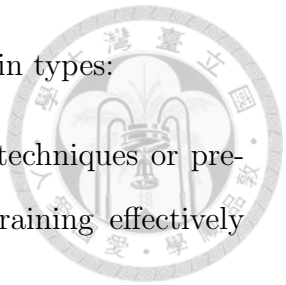
Literature Review

In this chapter, we initially explore the concept of Data-Efficient Generation, with a specific focus on one of its implementation strategies - the Pre-trained Generative Adversarial Network. Subsequently, in Section 2.2, we explore diverse types of Image-to-Image Translation techniques, including but not limited to Unsupervised Image-to-Image Translation and Data-Efficient Image-to-Image Translation, which aligns with the objectives of our study. Section 2.3 provides a succinct overview of the Masked-AutoEncoder utilized during our pre-training process and discusses previous work related to its integration with generative models. Lastly, we introduce the concept of latent space embedding, a critical component in understanding our proposed methodology.

2.1 Data-Efficient Generation

Generative Models have made significant advancements in image synthesis, which can be attributed to the emergence of large-scale datasets. However, this progress also necessitates increased computational resources and time. In scenarios with limited training data, stabilizing the training process and generating realistic images have become increasingly crucial research objectives. Consequently, a novel class of generative tasks, referred to as Data-Efficient Generation, has emerged.

doi:10.6342/NTU202303360



Previous research [3] has categorized this field into three main types:

1. Knowledge Sharing: This approach uses transfer learning techniques or pre-trained methods to reduce data demand and speed up training effectively [4, 5, 6, 7, 8].

2. Model Architecture: This approach involves designing specific model architectures to avoid overfitting [9, 10, 11].

3. Data Augmentation and Regularization: This approach involves transforming input data to augment the original dataset or enabling the model to produce consistent outputs when given augmented inputs, thereby allowing the model to learn some prior knowledge [12, 13, 14, 15].

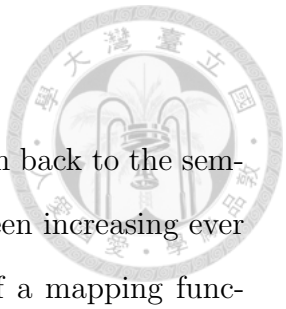
2.1.1 Pre-trained Generative Adversarial Networks

Previous studies [4, 6] have shown that the knowledge obtained from pre-trained generative networks is intimately related to the source data distribution. Transfer learning proves more effective when the source and target domains exhibit greater similarity. Therefore, a few papers [8, 7] have utilized the same dataset for both pre-training and fine-tuning by altering and combining model components to achieve the desired pre-training effect.

Although previous work [4] demonstrates that GANs can benefit from pre-trained models, particularly when dealing with limited data, the competitive nature of GANs, which consists of two mutually adversarial models, raises questions about the optimal approach to pre-train GANs. Several studies [4, 6, 7] indicate that to effectively employ a pre-trained GAN network, it is crucial to pre-train both the generator and the discriminator. Otherwise, pre-training results might even be inferior to those obtained from training from scratch.

doi:10.6342/NTU202303360

2.2 Image-to-Image Translation



The term "Image-to-Image Translation (I2I)" traces its origin back to the seminal work of Isola et al., titled 'pix2pix' [16], and its usage has been increasing ever since. Initially, this paper drafted I2I as the learning process of a mapping function from an input image to a corresponding output image. However, the advent of Unsupervised I2I [17] has contributed to a more expansive definition [18], encapsulating I2I as a process of transferring images from a source domain to a target domain while retaining the inherent content representations. Over recent years, I2I has garnered increasing interest and witnessed considerable advancements, primarily due to its wide-ranging applicability across several image generation challenges, including but not limited to image synthesis [16, 19, 20], style transfer [17, 21, 22], image restoration [23], and super-resolution [24].

2.2.1 Unsupervised Image-to-Image Translation

In practical applications, obtaining paired and content-aligned image pairs is often challenging. This issue has steered the bulk of I2I research towards Unsupervised I2I [17], the objective of which is to infer a transformation from the source domain to the target domain without necessitating paired data. Given the nebulous nature of this definition, the translated outcomes can potentially be ambiguous without supplemental restrictions. To mitigate this ambiguity, extant I2I methodologies incorporate constraints to maintain image content based on semantic features [25, 26].

2.2.2 Data-Efficient Image-to-Image Translation

While numerous existing I2I models can produce realistic and high-quality images, they typically require extensive training data. The quality of the generated images tends to degrade in scenarios where the data is limited. Consequently, the

doi:10.6342/NTU202303360

concept of Data-Efficient I2I emerged. An earlier model, FUNIT [9], utilized an additional style encoder and trained by taking the mean of multiple style codes as the input to AdaIN [22]. This approach aimed to construct a useful latent space so that during inference, only a few target style images are required to obtain the necessary style code to complete the translation in unseen domains. Considering the success of Data Augmentation in the realm of Data-Efficient Generation [12, 13], ReMix [1] chose to generate new intermediate samples by interpolating training data at the feature level. It also proposed a novel content loss based on the perceptual relationship between samples, achieving the effect of data augmentation.

Recent advancements, particularly in large-scale diffusion models, have generated realistic and diversified images. Consequently, many researchers have shifted their focus from GANs to diffusion models. State-of-the-art works such as PITI [27] and pix2pix-zero [28] exemplify this trend. However, given the vast parameter space of diffusion models, these methods primarily manipulate the latent space, altering the input text prompt or style code to control the content of the generated images. They refrain from additional training on the original diffusion model, which consequently restricts the generated downstream tasks within the generating scope of the original diffusion model. This limitation manifests in the inability to generate images for rare or unseen data. Therefore, training a Data-Efficient I2I Diffusion Model from scratch remains an intriguing and valuable research problem.

2.3 Masked AutoEncoder

The concept of a Denoising Autoencoder [29], introduced in 2008, was designed to facilitate robust representation learning, even in the presence of partial input corruption. Recent studies [30, 31] have successfully applied masked autoencoding techniques in the spatial dimension, where random pixel grids are masked, resulting in notable improvements. These advancements have subsequently triggered a wave

doi:10.6342/NTU202303360

of exploratory research centered around the capabilities of masked autoencoders.

2.3.1 Masked AutoEncoder with Generation Model

Masked Autoencoders (MAEs) have achieved considerable success in semi-supervised learning (SSL) tasks, particularly those related to classification. As a result, a significant number of subsequent studies have concentrated their efforts on the encoder component of the MAE, with minimal attention given to the decoder due to its typical omission in downstream tasks. However, a recent shift has occurred. Several recent works [32, 33] have aimed to enhance the pre-training effectiveness of the MAE encoder by integrating the structure of Generative Networks with the existing MAE architecture. This merging unexpectedly unveiled that the MAE's decoder also yields promising results in downstream generative tasks, such as inpainting.

2.4 Latent Space Embedding

In the domain of image generation tasks, the utilization of an auxiliary style code as input is occasionally mandated to exert more precise control over the stylistic content of the generated images. The advent of AdaIN [22] set the stage, employing a pre-trained VGG as a style encoder to extract the style code. This code was then fed into the generator through a purpose-designed normalization block. This method was later adopted by the highly influential styleGAN [34] in the GAN domain, albeit with an added twist: using a latent decoder instead of a style encoder. Essentially, this involves leveraging a randomly initialized latent code in tandem with style (label) information to produce the style code, a strategy that has yielded notable successes within the GAN sphere. This progress has consequently instigated numerous further investigations, including the Image2styleGAN series [35, 36].

More recently, in image-to-image (I2I) tasks, numerous networks, exemplified

doi:10.6342/NTU202303360

by psp [37] and starGANv2 [38], have employed both latent decoder and style encoder for style code generation. These works have demonstrated promising results, yet the efficacy of these techniques with limited training data remains a topic open for exploration.

doi:10.6342/NTU202303360

Chapter 3

Methodology

In this chapter, we will first define the problem at hand and enumerate all the notations used throughout this thesis. Following that, we will explicate the application of Masked Autoencoders (MAEs) in pre-training for Unsupervised Image-to-Image (I2I) translation tasks. We will also present a newly conceived normalization block designed to leverage the capabilities of a pre-trained feature extractor to enhance the diversity of generated images. Figure 3.1 shows the overview architecture of the proposed method.

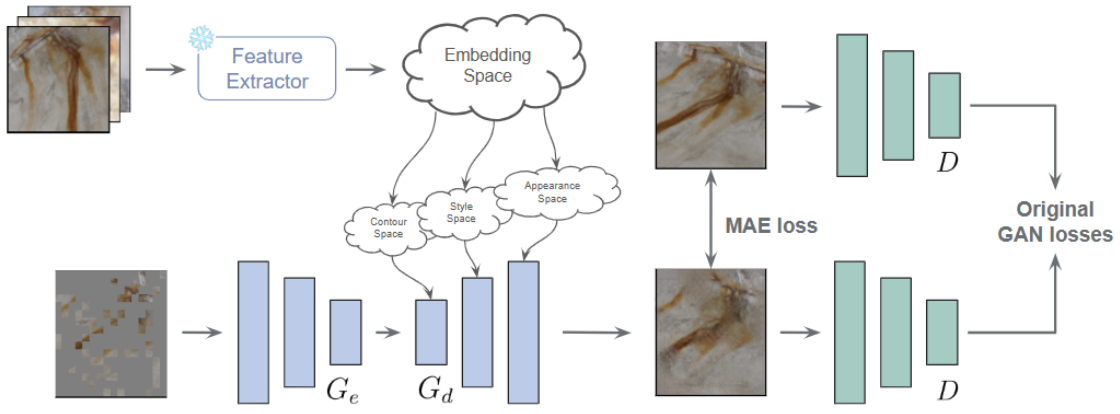


Figure 3.1: Overview of proposed architecture.

doi:10.6342/NTU202303360

3.1 Problem Definition

The objective of Unsupervised I2I is to transform images from the source domain to the target domain without necessitating paired data. However, the task of learning the generative task of domain transformation is inherently challenging, often requiring a substantial amount of data to ensure training stability. Under such circumstances, the fidelity and diversity of the generated images often serve as trade-off points, a situation that becomes even more pronounced in cases of data scarcity.

This thesis aims to address the aforementioned issues, namely: i) promoting the stable training of a viable generative model, and ii) guaranteeing the diversity of the images generated, particularly when confronted with data-limited situations. We aim to propose a pre-training approach for unsupervised I2I translation alongside a novel architectural design for style code insertion.

3.1.1 Notations

Before introducing our proposed method, we present a list of symbols used in this work:

Data Representation

- x: input image
- s : style code
- c : style label
- y : generated image with the content of source image x_s and the style of target image x_t
- X : dataset containing different style datasets $X = \{X_1, X_2, ..., X_N\}$, for each dataset $X_i \in X$, there exists a corresponding ground truth label $c_i \in C$

doi:10.6342/NTU202303360

- C : label set $C = \{c_1, c_2, ..., c_N\}$

Network

- G_e : encoder component in generator
- G_d : decoder component in generator
- D : discriminator
- E : style code generator

Loss

- $\mathcal{L}_{adv}$: adversarial loss
- $\mathcal{L}_{cls}$: classification loss
- $\mathcal{L}_{rec}$: reconstruction loss
- $\mathcal{L}_{style}$: style loss in the original architecture

Loss Weight

- λ_{cls} : weight of the classification loss
- λ_{rec} : weight of the reconstruction loss
- λ_{style} : weight of the style loss

3.2 MAE-GAN for I2I Pre-training

Image generation tasks are generally more challenging compared to other tasks, and achieving stable training is a consistent area of research. This is especially the case when data are limited. Existing approaches include techniques such as i) data augmentation and regularization, ii) specially designed model architectures, and iii) knowledge-sharing strategies like pre-training or transfer learning.

doi:10.6342/NTU202303360

Extensive research has been conducted into data augmentation and regularization, as evidenced by numerous previous papers. However, model architectures are often specific to certain tasks or datasets and thus are difficult to use more broadly. Consequently, we directed our research toward knowledge-sharing. By integrating Generative Adversarial Networks (GANs) with the Masked Autoencoder (MAE) architecture, which has achieved great results in the Self-Supervised Learning (SSL) domain, we propose a novel pre-training method for Unsupervised Image-to-Image (I2I) tasks. This approach provides the model with a better starting point for training, ensures stability during the training process, and yields higher-quality images.

Figure 3.2 shows our MAE-GAN for I2I model. In MAE-GAN pre-training, we employ the structure of a Masked Autoencoder (MAE) for pre-training the multi-domain image-to-image translation model. During pre-training, an additional style input is inserted with the intention for the generator to not only learn to repair masks but also to acquire style-related information beyond lower-level attributes such as structure or edges.

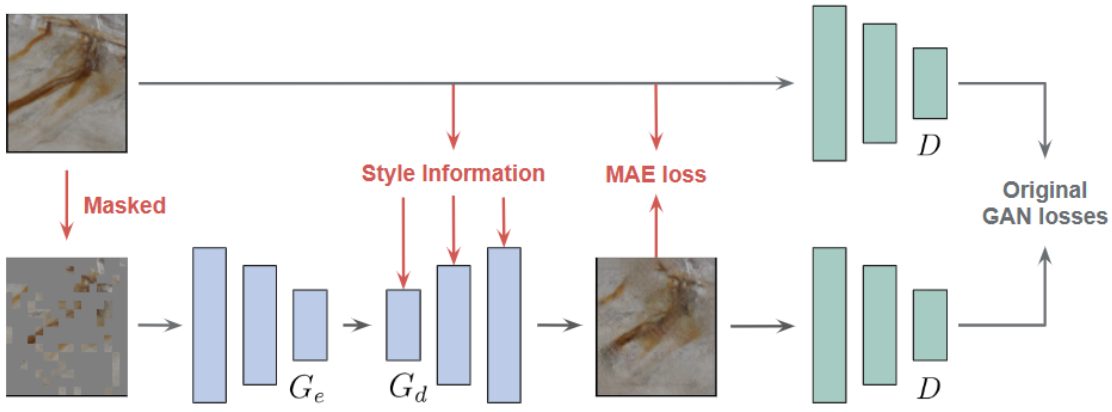


Figure 3.2: Overview of proposed architecture.

doi:10.6342/NTU202303360

3.2.1 Pre-training Pipeline

Our objective is to pre-train a generator G, a discriminator D, and a style code generator E (if E exists in the downstream task). During the pre-training phase, G is trained to transform a masked image x_m into an unmasked output image $\hat{x}$, which is governed by a style code s. This style code s is either a direct derivative of the class label c or a product of E in the form of $E(c) \rightarrow s$, facilitating the pre-training of G and E. Due to GAN models' adversarial nature, we integrate an existing Masked Autoencoder (MAE) architecture with the GAN framework. Both the real input image x and the unmasked image $\hat{x}$ are fed to D for the computation of $\mathcal{L}gan$ and $\mathcal{L}cls$ for $x, \hat{x}$, thereby enabling the pre-training of D. We find that these supplementary losses will improve the quality of images generated and bolster the learning of style-specific information.

3.2.2 Training Objectives

Given an image $x \in X$ and its domain $c \in C$, and the style code s is either a direct derivative of the class label c or a product of E in the form of $E(c) \rightarrow s$, and we will mask input image x to get the masked input x_m, then we train our framework as the following objectives.

Reconstruction Loss. During pre-training, the generator G accepts the masked image x_m and s as input and learns to generate an unmasked image $\hat{x} = G(x_m, s)$ through a reconstruction loss function, defined as follows:

$$\mathcal{L}rec = \mathbb{E}x_m, x, c[\| \ x - G(x_m, s) \ \|_1], \tag{3.1}$$

the reconstruction loss function is similar to the one used in MAE [30], but with an additional style code s input. The intention here is to enable the generator G to learn not only high-level information from the unmasking process but also to understand how to generate images belonging to the style domain defined by the

doi:10.6342/NTU202303360

style label s, thereby assisting in downstream task learning.

Adversarial Loss. The adversarial loss is subject to modifications depending on the type of adversarial loss used in different downstream tasks. Without loss of generality, we provide a definition of the adversarial loss as follows:

$$\mathcal{L}adv = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{x_m,c}[\log(1 - D_{src}(G(x_m, s)))], \tag{3.2}$$

D_{src} is one of the output heads of D, used to determine if the input image is real or fake. In this process, G is trained to generate unmasked images that are indistinguishable from real images, improving the quality of the generated image details, unlike the original MAE, which tends to produce blurred images.

Domain Classification Loss. The classification loss may change depending on the classification loss type used in various downstream tasks. Without loss of generality, we define the classification loss as follows:

$$\mathcal{L}_{cls} = \mathbb{E}x, c[-\log D_{cls}(c|x)] + \mathbb{E}_{x_m,c}[-\log D_{cls}(c|G(x_m, s))], \tag{3.3}$$

D_{cls} is another output head of D, designed to identify the style label domain of the generated images. G is trained to generate unmasked images indistinguishable from images within the style label domain. The aim here is to ensure that, even if G cannot generate an image $\hat{x}$ identical to the original x, it should at least generate images similar to those within the style label domain, assisting in learning for downstream tasks.

Style Loss. Style loss refers to potential losses related to the style code generator E in downstream tasks. For instance, the Style Reconstruction Loss and Style Diversification Loss used in StarGANv2 [38]. These losses can be integrated into the training objective during the MAE-GAN pre-training phase to further optimize the pre-training of E. This component can be omitted if the downstream task does not

doi:10.6342/NTU202303360

require a similar loss or E.

Full Objective. Our full objective functions to optimize G, E and D can be summarized as follows

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}^{D}\mathcal{L}_{cls}, \tag{3.4}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}^{G}\mathcal{L}_{cls} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{style}\mathcal{L}_{style}, \tag{3.5}$$

where λ_{cls}, λ_{rec}, and λ_{style} are hyper-parameters that control the relative importance of different losses, respectively, compared to the adversarial loss. If you don't have style loss, you can directly set the value of λ_{style} to 0.In our Experiments, we use $\lambda_{cls}^{D} = 3$, $\lambda_{cls}^{G} = 1$, $\lambda_{rec} = 10$ and $\lambda_{style} = 0$ in all of our pre-training experiments.

3.2.3 Masked Strategies

While masking the image, we adopt several suggestions and practices from previous MAE-related work. i) We used larger mask sizes as suggested in [30, 31] to prevent the generator from learning to fill in masked pixels based solely on neighboring unmasked pixels. ii) We utilized the shifted mask technique as implemented in [39], which involves randomly shifting the mask by a grid of pixels. This approach helps prevent the generator from learning fixed grid-shaped patterns, allowing for a dynamic mask ratio. Finally, iii) we incorporated a learnable position mask, i.e., we transformed the mask values from their original zeros into learnable values, unique for each pixel. This technique protects the generator from being influenced by the mask values while preserving the original intention of the corrupt input. These three strategies aim to enhance the effectiveness of the MAE-GAN pre-training process.

doi:10.6342/NTU202303360

3.3 Style Embedding Adaptive Normalization

To increase diversity in the generated images, traditional style codes typically employ latent decoders or style encoders. These are typically produced by inputting a style label with noise or a style image. Techniques like the Style Diversification Loss in StarGANv2 [38] are employed to ensure the diversity of the generated style codes. However, when there is a lack of data, adding another set of parameters to train can lead to instability during training, and even poorer performance than without these techniques. To address this, we propose a novel block, which achieves satisfactory results with the training of relatively few parameters and still performs well when data is scarce.

Figure 3.3 shows the proposed Style Embedding Adaptive Normalization (SEAN) block. SEAN block is a normalization block to augment the diversity of the generated images. This block takes as input several style features, which are obtained from style images processed through a pre-trained feature extractor. These style features are subjected to a straightforward dimensionality reduction, averaged, and then combined with the label embedding to ultimately generate the required style code for the generator. Importantly, this method can self-learn the distribution of style codes for each label, and through the aggregation of multiple style codes during training, it ensures that the style code space is continuous. This allows acquiring the desired style code directly from the label without needing a style image post-training. Moreover, by adjusting the standard deviation of the distribution, a balance can be struck between the diversity and fidelity of the generated images. This process effectively completes the style translation task in Image-to-Image Translation.

doi:10.6342/NTU202303360

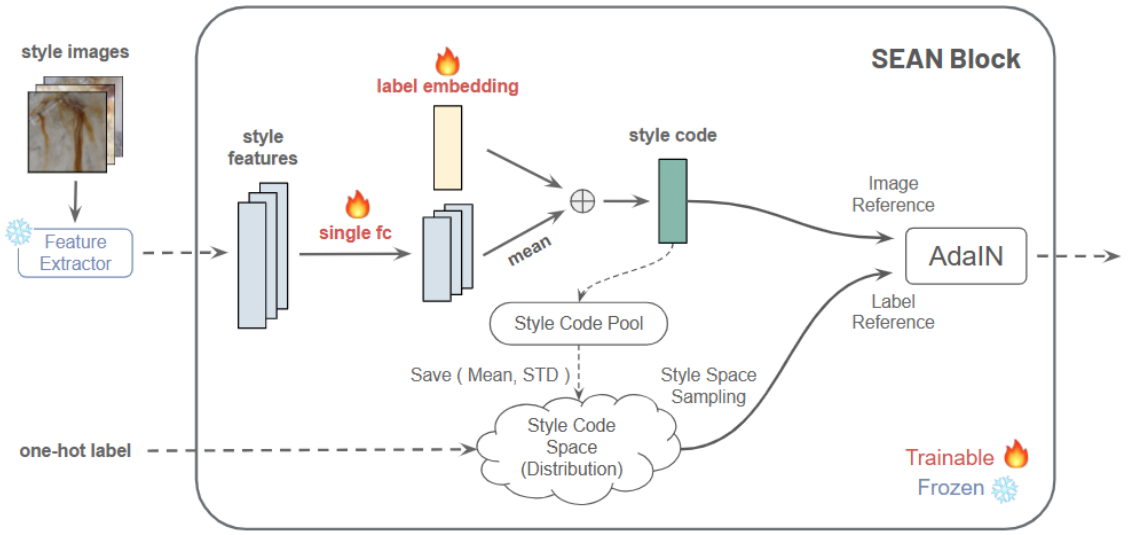


Figure 3.3: The architecture of proposed SEAN block.

3.3.1 Pre-trained Feature Extractor as Style Code Generator

In our pursuit of training a model component with a minimal number of parameters that can nonetheless generate a broad diversity of style codes, the overall style code extraction process can be formalized as follows:

$$s_i = mean(\sigma(\mathbf{W}_i E(\tilde{x}))) + \rho_i(c), \tag{3.6}$$

where $\tilde{x}$ represents a collection of augmented style images x' that are derived by applying style-preserving augmentations to several input style images x.

$$\tilde{x} = [x'_0, \cdots, x'_n], \tag{3.7}$$

for pre-trained feature extractors E, specifically, we utilize the Vision Transformer (ViT) [40], a widely recognized and commonly used tool in this domain. We input augmented style images $\tilde{x}$ and retrieve extracted style features from the ViT's cls-

doi:10.6342/NTU202303360

token output. These style features are subsequently passed into our custom-designed Style Embedding Adaptive Normalization (SEAN) blocks.

Following the insights provided by StyleGAN [34], the requirement for style codes differs across layers, each correlating to distinct features in the generated image. Consequently, the weights of the SEAN blocks are layer-specific. Identical style features inputted into different layers' SEAN blocks are first dimensionally reduced via a single fully connected layer W_i to a hidden dimension. Following a straightforward non-linear activation layer σ, these style codes will be aggregated, resulting in the foundational style code.

The objective of the dimensionality reduction procedure is to allow different layers to extract features that are important to each layer from the comprehensive style features. As for style code aggregation, the first aspect is to achieve an effect similar to data augmentation, and the second aspect is to allow the SEAN block to learn that each style code space is continuous through the input of multiple style code aggregations. Therefore, it can be guaranteed that any style code sampled from this style code space can generate a real image corresponding to the label.

Furthermore, although the feature extraction capabilities of the pre-trained feature extractor are strong, the dimension reduction process may inevitably lose some image features. Therefore, we added a style label embedding ϕ_i to supplement the style information missing from the style feature by inputting different style labels c to obtain the style label embedding. These two components are then added to obtain a better style code s_i.

Finally, after the AdaIN operation is applied to s_i, it is fed to the generator G. The AdaIN operation is defined as:

$$\text{AdaIN}(x_i, s_i) = s_\sigma \frac{x_i - \mu(x_i)}{\sigma(x_i)} + s_\mu, \tag{3.8}$$

doi:10.6342/NTU202303360

where each feature map x_i is normalized separately, and then scaled and biased using the corresponding s_σ and s_μ vector transformed from style code s_i via a single fully connected layer.

While our entire block and pipeline may appear complicated, excluding the parameters inherently involved in the AdaIN operation, the truly trainable parameters remain limited to the dimension reduction W_i and the style label embedding ϕ_i—both of which are merely low-dimensional fully connected layers. Consequently, our methodology can be efficiently trained and stable even under data-limited conditions.

3.3.2 Style Code Space

Although the previously mentioned SEAN block can obtain high-quality style codes to complete Image-to-Image (I2I) tasks, it relies on a style image as input to acquire the style code during inference. Consequently, the style of the generated output is constrained by the quantity and variety of the style images. How can we address this issue? We propose a method to improve the SEAN block, enabling it to obtain diverse and high-quality style codes during inference using only the style label.

Our approach is straightforward and does not require additional training parameters. The style code pool retains the style codes generated during training, and every few epochs, it is used to calculate the mean and standard deviation of the style code space. As we ensure the code space's continuity by leveraging the style code's mean during the SEAN block's training phase, we can preserve the means and standard deviations of these styles along with the model weights. Consequently, we can randomly sample a style code from this distribution during the inference stage, resulting in high-quality and diverse style codes.

Furthermore, the balance between fidelity and diversity can be adjusted by varying the multiples of the standard deviation. One could select a standard devia-

doi:10.6342/NTU202303360

tion multiple of 1 or 2 for stable output quality. Conversely, to increase diversity in the generated images, one might choose a larger multiple of the standard deviation, though the quality of the style code might be less stable. This flexibility allows us to meet the objectives of various downstream tasks.

doi:10.6342/NTU202303360

Chapter 4

Experiments

In this chapter, we first introduce the datasets employed in our study and the corresponding baseline models. We then outline the evaluation metrics chosen from a broad selection of generative metrics and provide a rationale for our selections. Subsequently, we analyze the effects of individual components within the two methods we proposed. Finally, we draw comparisons between our results and those of previous research. This comparative analysis allows us to understand the advantages and disadvantages of our methods in a broader context.

4.1 Datasets

In this research, our target datasets are those with insufficient data. Given the absence of a dataset specifically dedicated to the Data-Efficient Multiple Domain Unsupervised Image-to-Image Translation (I2I) task, we chose to employ standard datasets for our study. For training purposes, we randomly selected 10% of the data from these datasets.

Our experiments mainly utilized the Concrete Defect Bridge Image (CODEBRIM) Dataset [41]. This dataset possesses distinctive characteristics, including: i) The dataset consists of rare images, with no similar large-scale datasets available; ii) The dataset has imbalanced quantities of images across different categories; iii)

doi:10.6342/NTU202303360

the image sizes and resolutions within the dataset are diverse. These attributes make the CODEBRIM dataset closely resemble the conditions of real-world data collections.

Given that images of defect patterns are sparse and exhibit irregularities, evaluating the quality of synthesized images is not straightforward. Consequently, we use the Animal Faces-HQ (AFHQ) Dataset for visual demonstrations of our results. Additionally, the AFHQ dataset is widely employed in Multiple Domain Unsupervised Image-to-Image (I2I) tasks, facilitating comparison with state-of-the-art methods and providing deeper insights into the performance of our approach.

4.1.1 COncrete DEfect BRidge IMage

The Concrete Defect Bridge Image Dataset (CODEBRIM) is a multi-target, multi-class dataset designed for concrete defect classification. It consists of six mutually non-exclusive classes: crack, spallation, exposed reinforcement bar, efflorescence (calcium leaching), corrosion (stains), and normal surface samples. The number of defects for these classes ranges from 833 to 2507, as represented in a selection of images from the dataset shown in Figure 4.1. In our experimental setup, all images are randomly cropped to a resolution of 128x128 pixels for the training phase. We preserve the original class distribution and randomly select 10% of the original training set to serve as the training data. The same test data set is used for the evaluation phase.

4.1.2 Animal Faces-HQ

The Animal Faces-HQ (AFHQ) is a dataset of animal faces, specifically tailored for the task of Multiple Domain Unsupervised Image-to-Image Translation. It was first introduced in StarGAN-v2 [38]. Comprising 15,000 high-quality images of 512 x 512 resolution, this dataset is represented with a selection of images in Figure 4.2. The dataset includes three domains - cats, dogs, and wildlife - each with 5,000

doi:10.6342/NTU202303360

Figure 4.1: Examples from CODEBRIM dataset.

images. These images are further partitioned into a training set containing 4,500 images and a testing set with 500 images for each domain. In our study, we construct our training set by randomly selecting 500 images from each domain's original training set, while maintaining the original testing set unchanged. All images are randomly resized to a resolution of 256x256 pixels for the training phase.

Figure 4.2: Examples from AFHQ dataset.

4.2 Experiments Setup

For CODEBRIM dataset, we utilize Defect-GAN [10] as our baseline and foundational structure. Defect-GAN, specifically designed for defect pattern generation, has shown impressive results on the CODEBRIM dataset. Its underlying architecture is a variant of StarGAN [42]. However, it adopts one-hot labels as input, which may limit the diversity of the generated images.

doi:10.6342/NTU202303360

For the AFHQ dataset, we utilize StarGAN v2 [38] as our foundational structure, which addresses the limitation of StarGAN's inability to generate diverse images. As our baseline, we use the results of the ReMix [1] paper, which was trained on 10% of the dataset volume (since our reproduced results were inferior). Moreover, we compare our results with those of state-of-the-art methods, ReMix and ScoreMix [2], which address the identical task under the same constraints.

4.3 Evaluation Metrics

In generative tasks, numerous evaluation metrics exist. Prominent among these are the Frechet Inception Distance (FID), used to gauge visual quality, and the Inception Score (IS), utilized for measuring diversity. However, IS has a well-known limitation in that it uses the classification results of the Inception model [43] on ImageNet [44] to compute diversity. This could result in meaningless IS scores if the data, such as the CODEBRIM dataset, doesn't fall within the 1000 classes used for pre-training. This phenomenon is highlighted in our experimental results shown in Table 4.3 and Table 4.4. Therefore, in our study, we measure diversity using the Learned Perceptual Image Patch Similarity (LPIPS), a metric that is commonly adopted in recent image generation tasks.

4.3.1 Frechét inception distance

The Frechét Inception Distance (FID) is utilized to compute the Frechét distance amid two sets of images, thereby assessing the quality and likeness of the synthesized images. This is accomplished by determining the disparity between the distributions of real and generated images. Consistent with common practices, we employ feature vectors from the concluding average pooling layer of the ImageNet-trained Inception-V3.

In the context of our approach, each test image from a specified source domain

doi:10.6342/NTU202303360

is transformed into the target domain employing a set of 10 style codes. For style space sampling, these style codes are derived from our chosen style embedding distribution. In reference-guided synthesis, these style codes are synthesized by averaging the results from a previously trained feature extractor, utilized on 10 clusters of reference images. These reference images are randomly selected from the target domain test set. Consequently, we compute the FID score amid the transformed images and the target domain training images. We report the average FID scores across all image domain pairs.

4.3.2 Learned perceptual image patch similarity

The Learned Perceptual Image Patch Similarity (LPIPS) serves as a measure to quantify the perceptual similarity between a pair of images. This metric essentially measures the similarity amid image patches under a predefined network and is known for its alignment with human perception. The common practice involves generating multiple images for each image in the dataset utilizing the same label but different style codes. Subsequently, pairs of generated images are selected, their LPIPS values calculated, and the outcomes averaged. Notably, a higher LPIPS value signifies a higher image diversity.

In our methodological approach, we use the LPIPS algorithm that StarGAN v2 originally used. Specifically, for each test image from a source domain, we generate 10 target domain images employing 10 different style codes. The methodology for obtaining these style codes is the same as when we calculate FID scores. Subsequently, we calculate the average of the pairwise distances among all outputs derived from the same input (i.e., 45 pairs). Ultimately, we report the average LPIPS values across all test images.

doi:10.6342/NTU202303360

4.4 Evaluation and Results

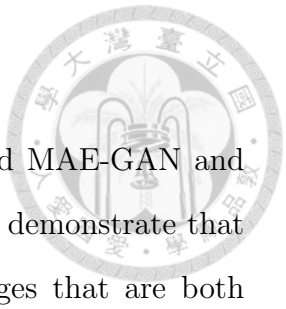


Table 4.1 provides a quantitative evaluation of our proposed MAE-GAN and SEAN methods on the CODEBRIM dataset. The results clearly demonstrate that each method individually contributes to the generation of images that are both more realistic and diverse. When both methods are combined, we achieve results that compete favorably with state-of-the-art techniques, as presented in Table 4.2. The asterisk (*) in this table indicates results directly taken from their respective papers, as detailed implementation details and code were not disclosed. Therefore, these results should be considered for reference only due to differences in dataset partitioning during training. As can be seen, through the utilization of powerful features extracted by our pre-trained feature extractor, our method achieves a significant LPIPS score of 0.523, outperforming all previous methods. Importantly, our approach is based on pre-training, whereas other techniques often resort to data augmentation methods. Consequently, we anticipate that integrating state-of-the-art strategies into our work could yield superior results. However, as the source code is not publicly available for the other two papers, investigating this concept presents a promising direction for future work.

Method Choices				
MAE-GAN	**SEAN**	**Style Space Sampling**	**FID↓**	**LPIPS↑**
✗	✗	✗	86.02	–
✓	✗	✗	65.82	–
✗	✓	✗	89.20	0.1475
✓	✓	✗	64.72	0.1534
✓	✓	✓	**61.05**	**0.2239**

Table 4.1: Comparative Analysis of Different Configurations.

To focus on numerical comparisons and better demonstrate our achievements, we generated many images on these two testing datasets to showcase our

doi:10.6342/NTU202303360

Method	FID	LPIPS
Baseline 10% data	42.09	0.450
Baseline 100% data	20.32	0.438
*ReMix	22.92	0.460
*ScoreMix	**17.99**	0.466
Ours	22.29	**0.523**

Table 4.2: Results Compared with State-of-the-art Methods

results. Figure 4.3 compares our method with the baseline on more challenging image transformations. As can be seen, the images produced by the baseline do not correspond as closely to the reference images. The generated images are significantly inaccurate, particularly in cases with multiple subcategories, such as the "Wild" category. Our proposed method, however, excels in learning and merging the features of both source and reference information.

Moreover, Figures 4.4 and 4.5 demonstrate the successful preservation of continuity in the style code space using multiple style codes mean, which can generate numerous realistic hybrid creatures not originally present in the dataset. In the following sections, we will delve into the roles of the various components in our proposed method.

4.5 Futher Analysis for Each Component in MAE-GAN

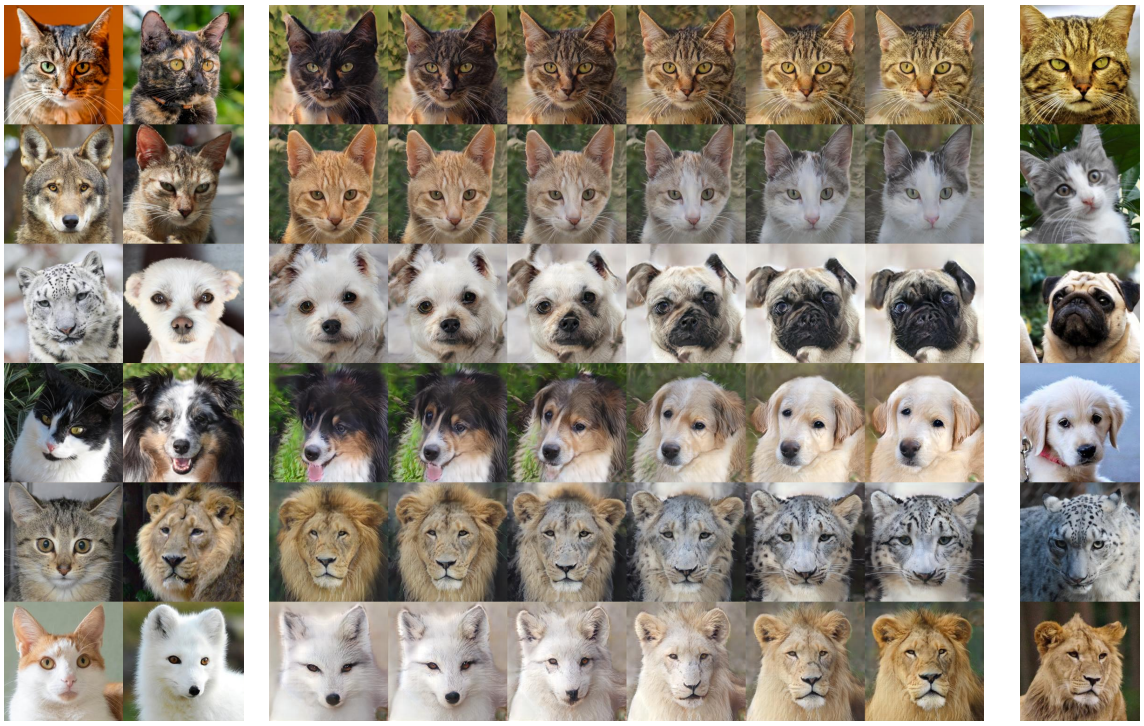
Figure 4.6 displays the outcomes after pre-training on the CODEBRIM datasets. Each triplet displays the Ground Truth (Left), the Reconstructed Image via our MAE-GAN (Middle), and the Original Masked Image (Right). Following pre-training, the model demonstrates robust generative capabilities, producing repaired results that are free from the blurriness observed in previous work [30]. Instead, the outcomes feature distinct details and innovative imagery. Moreover, as illustrated in Figure 4.7, the Generator, post pre-training, has assimilated knowledge across

doi:10.6342/NTU202303360

(a) Baseline (b) Ours

Figure 4.3: Qualitative Comparison of Image Synthesis Results Using 10% of the AFHQ Dataset.

doi:10.6342/NTU202303360

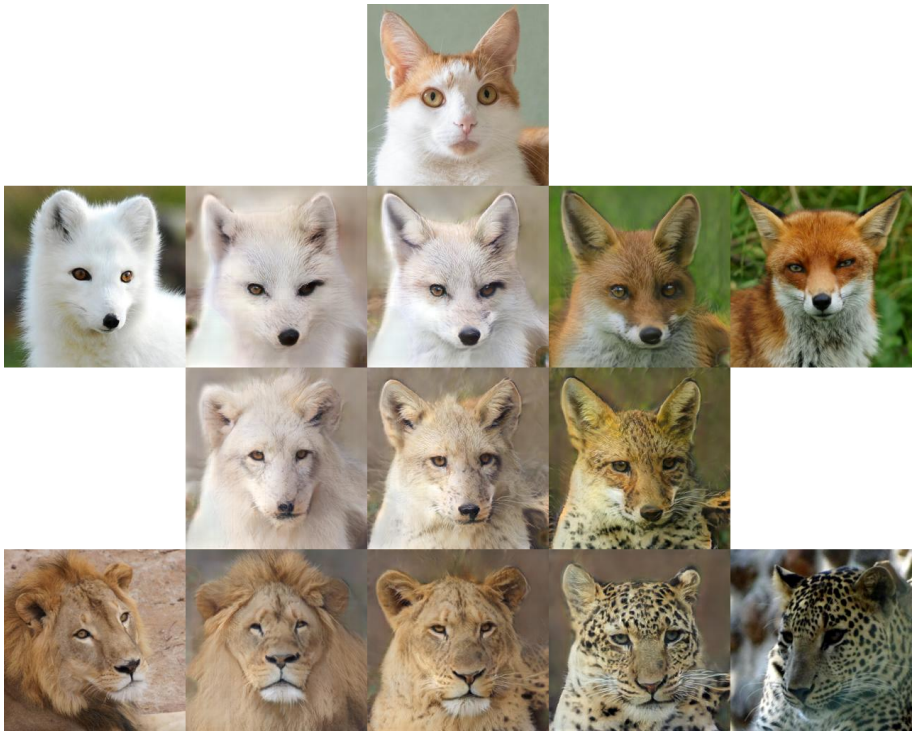


Source | First Reference | Interpolation Outputs | Second Reference

Figure 4.4: Diverse Translation Results by Interpolating Two Style Codes in the SEAN Block.

doi:10.6342/NTU202303360

(a) Mixed Cat Outputs



(b) Mixed Wild Animal Outputs

Figure 4.5: Diverse Translation Results by Mixing Multiple Style Codes in the SEAN Block.

doi:10.6342/NTU202303360

various domains. As a result, it can expedite training in downstream tasks and generate more realistic images.

(a) CODEBRIM dataset (b) AFHQ dataset

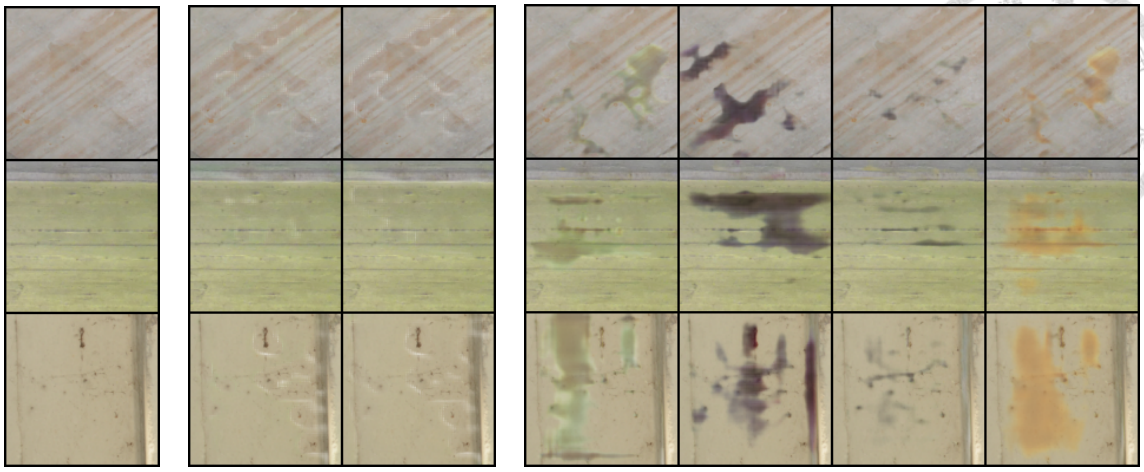
Figure 4.6: Reconstructed images from the validation set after pre-training with our MAE-GAN.

4.5.1 Masked Autoencoder Generative Adversarial Network Framework

Table 4.3 showcases the performance of various Masked Autoencoder (MAE) and Generative Adversarial Networks (GAN) framework combinations. The integration of MAE and GAN results in significant improvements, evidenced by a 21-point reduction in the FID score, demonstrating the efficacy of this combined approach.

Importantly, the beneficial effect is observed when MAE and GAN are combined; separating the training of the Generator and Discriminator negatively impacts the outcome. In this scenario, the Generator is solely pre-trained with MAE, and the Discriminator is trained using a classification task, which leads to an increase in FID from 65 to 75.

doi:10.6342/NTU202303360



Source (a) Train from Scratch (b) After MAE-GAN Pre-training

Figure 4.7: Comparison of Images Generated During Initial Training With and Without Pre-training

On the other hand, if the integration is maintained and only style code insertion is removed, thereby preventing images from undergoing AdaIN transformation, the FID only increases from 65 to 70. This suggests that the impact isn't as substantial, indirectly proving the utility of style code insertion. These results affirm the importance of harmonious interaction between the various components of the MAE-GAN framework for optimized performance.

MAE-GAN Framework	**FID**	**IS**
Baseline : DefectGAN	86.02	3.031
MAE-GAN	**65.82**	2.991
w/o style code insertion	70.82	2.935
Split training	75.73	**3.095**

Table 4.3: Performance Comparison of Different Framework Combining Masked Autoencoder (MAE) and Generative Adversarial Networks (GAN).

4.5.2 Masking Strategy

Table 4.4 presents the performance comparison across various masking strategies. Each component significantly reduces the FID score, demonstrating their unique utilities. When utilizing the MAE-GAN framework before applying the three

doi:10.6342/NTU202303360

methods under consideration, the FID score already displays a substantial reduction. Then a comprehensive analysis of the individual impact of each component will be carried out in the upcoming sections. **Mask Size** Figure 4.8 examines

Mask Strategies	**FID**	**IS**
Baseline : DefectGAN	86.02	3.031
Original MAE-GAN	74.67	3.018
+ large mask	71.25	2.924
+ shifted mask	69.73	**3.057**
+ learnable mask	**65.82**	2.991

Table 4.4: Performance Comparison of Various Masking Strategies.

the effect of varying mask sizes on the performance of downstream tasks. Given an image size of 128x128 pixels, we found that a moderate mask size, ranging from 8 to 16, produces superior results in downstream tasks. This observation aligns with the findings of SIMMIM [31]. Specifically, if the mask size is too small, the generator tends to repair the mask directly from nearby pixels, bypassing the learning of useful features. Conversely, increasing the mask size does not significantly impact performance, suggesting that using a larger mask size during pre-training could be beneficial.

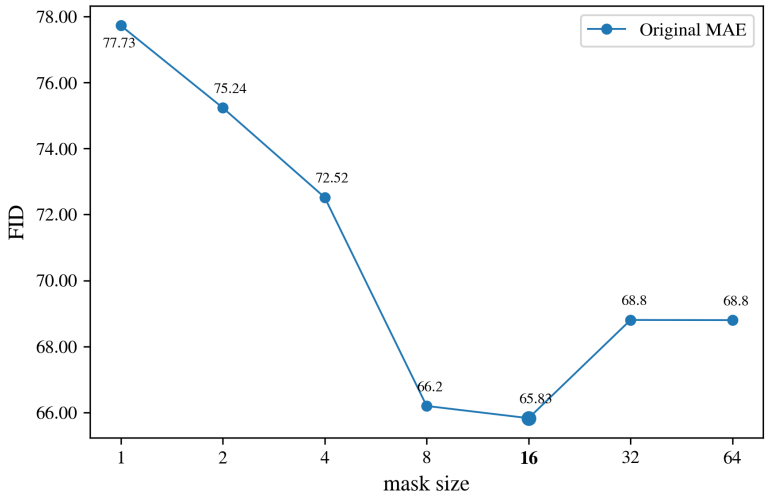


Figure 4.8: Efficacy of Mask Size

doi:10.6342/NTU202303360

Shifted Mask The idea of using a shifted mask is borrowed from the data augmentation techniques mentioned in [39]. Figure 4.9 analyzes the impact of using a shifted mask in combination with different mask sizes on the performance of downstream tasks. The results suggest that there is not much difference when the mask size is relatively small. However, as the mask size increases, not using a shifted mask can easily cause the model to learn the generative information at fixed locations, significantly reducing the effectiveness of pre-training. Therefore, it is best to use a shifted mask when increasing the mask size. **Learnable Mask** Our work diverges

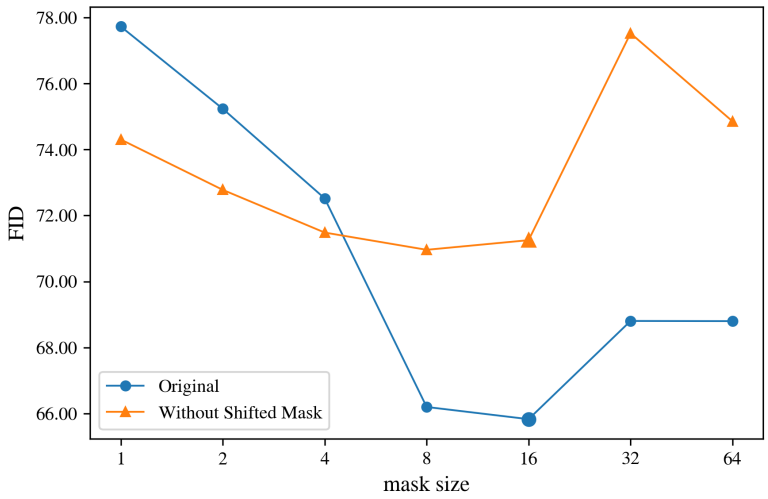


Figure 4.9: Impact of Mask Size with and without Shifted Mask.

from previous Mask Autoencoder (MAE) approaches by emphasizing the generative capabilities of MAE. To prevent the generator from being overly influenced by the mask values, while still maintaining the original purpose of the corrupted input, we employ learnable mask tokens to replace the original mask-zero values. In Figure 4.10, 'Zero' denotes the original mask method, while 'mean' uses the average of the unmasked pixel values as the mask value. 'Scalar' and 'vector' refer to channel 1 and 3 learnable masks, respectively, and 'position' and 'full' represent channel 1 and 3 learnable masks, with each pixel acting as a learnable mask token. As shown in this figure, learnable mask tokens that incorporate positional information, particularly

doi:10.6342/NTU202303360

of the 'position' and 'full' types, consistently outperform other mask token types in downstream tasks. Otherwise, the original zero-mask would be adequate.

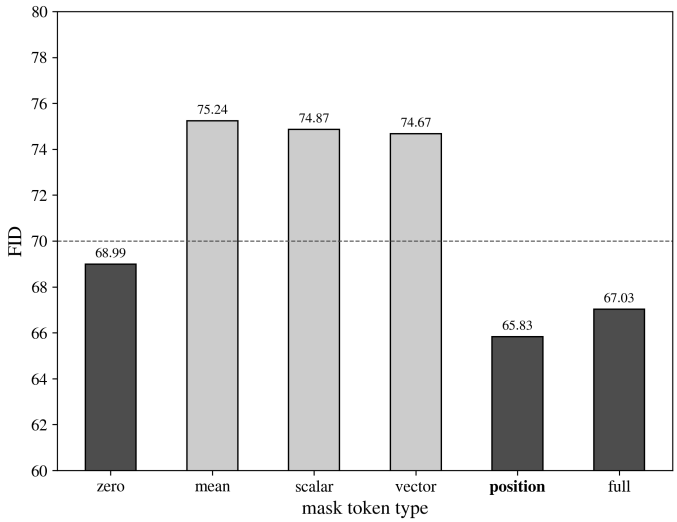


Figure 4.10: Performance of Various Mask Token Types.

Mask Ratio Figure 4.11 presents an analysis of the impact exerted by various masking ratios on the performance of downstream tasks. Consistent with conclusions drawn from previous work, our experiments also substantiate that a moderate masking ratio, specifically between 40% and 75%, consistently provides robust performance across a variety of downstream tasks.

4.6 Futher Analysis for Each Component in SEAN

Figure 4.12 showcases the diversity of the images generated by our method. Given the same label input, our approach can generate a range of realistic defect patterns while preserving the features of the original source images. Therefore, our method can be utilized not only for style transfer and generation of non-existent images but also for augmenting existing datasets.

doi:10.6342/NTU202303360

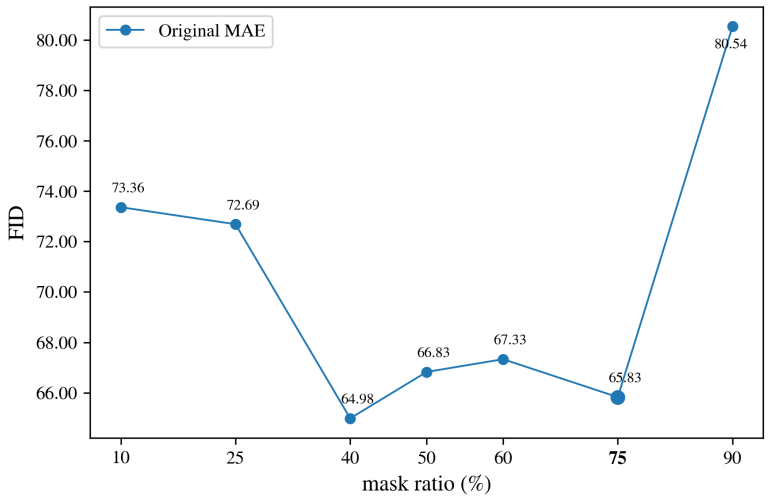


Figure 4.11: Efficacy of Different Mask Ratios.

4.6.1 Comparison of Various Style Code Insertion Methods

	Train from Scratch		MAE-GAN Pre-trained	
Method	**FID**	**LPIPS**	**FID**	**LPIPS**
Baseline : DefectGAN	86.02	–	65.82	–
(A) with latent decoder	95.14	0.0236	71.15	0.0539
(B) with style encoder	109.29	0.0001	85.11	0.0145
(C) with single fc	**81.13**	–	71.67	–
(D) with finetuned ViT	99.37	0.0662	73.99	0.1427
(E) with ViT	89.20	**0.1475**	**64.72**	**0.1534**

Table 4.5: Comparison of different style code insertion methods.

Table 4.5 contrasts different style code insertion methods and their performance with and without pre-training via MAE-GAN, with the original DefectGAN utilizing SPADE [20] to insert the style code. In contrast, (A) - (C) in the table utilizes distinct style code generators combined with AdaIN [22]. Under conditions of training from scratch with limited data, due to the additional training complexities presented by an extra style code generator in (A) and (B), the diversity of generated images could not be effectively improved. As a result, opting for a simpler option such as the single multi-layer perceptron (mlp) in (C), is found to be a superior choice. Though it does not generate diverse images like the original method, it

doi:10.6342/NTU202303360

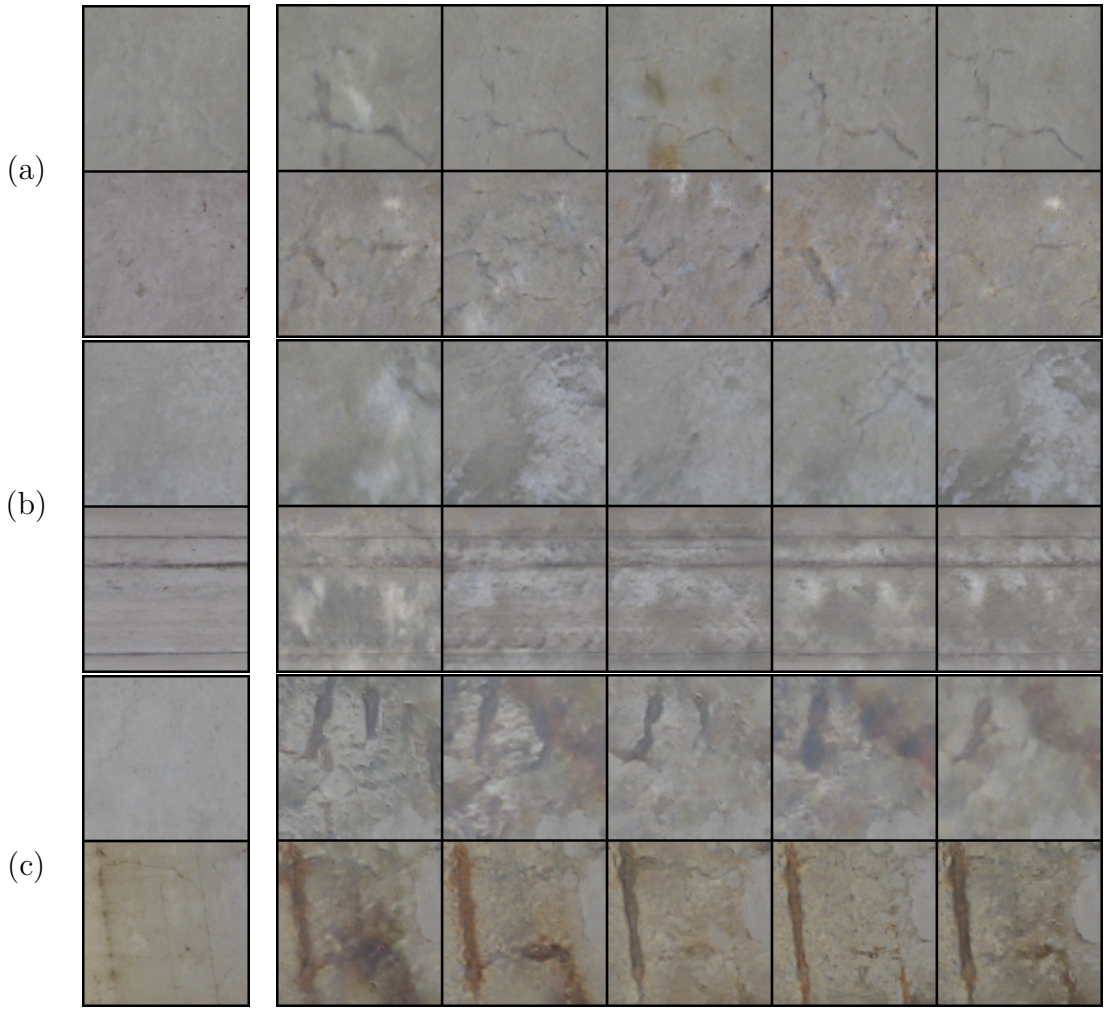


Figure 4.12: Diverse images generated from the same label input in the CODEBRIM dataset, for: (a) Crack, (b) Efflorescence, and (c) Spallation, Exposed Bars, Corrosion.

doi:10.6342/NTU202303360

guarantees the usability of the generated images.

In the case of employing our proposed SEAN block to input the style code, we compared two different approaches. (D) first finetune the pre-trained feature extractor on the target CODEBRIM dataset, then freezes the model's weight to extract features. In contrast, (E) utilizes a model directly pre-trained on ImageNet with frozen weights. It was observed that while (D) allowed the extraction of features closer to the dataset, it reduced the richness of feature information leading to a decrease in both the quality and diversity of generated images. Instead, (E) significantly improved the diversity of generated images with negligible loss in FID.

Under the conditions of pre-training using MAE-GAN, (A) - (C) have been observed to learn style-related information in advance, not only greatly reducing FID but also enhancing the diversity of the generated images. Moreover, the results were even better when using SEAN. As can be seen in Figure 4.13, after pre-training, the style codes for the same labels congregate together, thereby improving the learning outcome of downstream tasks. Ultimately, this led to a significant increase in the diversity of generated images and a slight improvement in image quality.

4.6.2 Multiple Style Codes Mean and Label Embedding

Table 4.6 presents a comparative analysis of various style code extraction methods while utilizing the SEAN block. These strategies can be divided into two categories: (i) and (B) extracting a style code from a single style image, (ii) and (A) computing the average of style codes obtained from multiple style images. It was found that using the aggregation of multiple style codes during the training phase helps the model to create a smoother and more continuous style space. As a result, the choice between a single style code or multiple style codes during the inference phase has a minimal impact on the outcomes. However, it is worth noting that training the model with a single style code can potentially lead to a significant decline in performance during the inference phase if the model encounters previously

doi:10.6342/NTU202303360

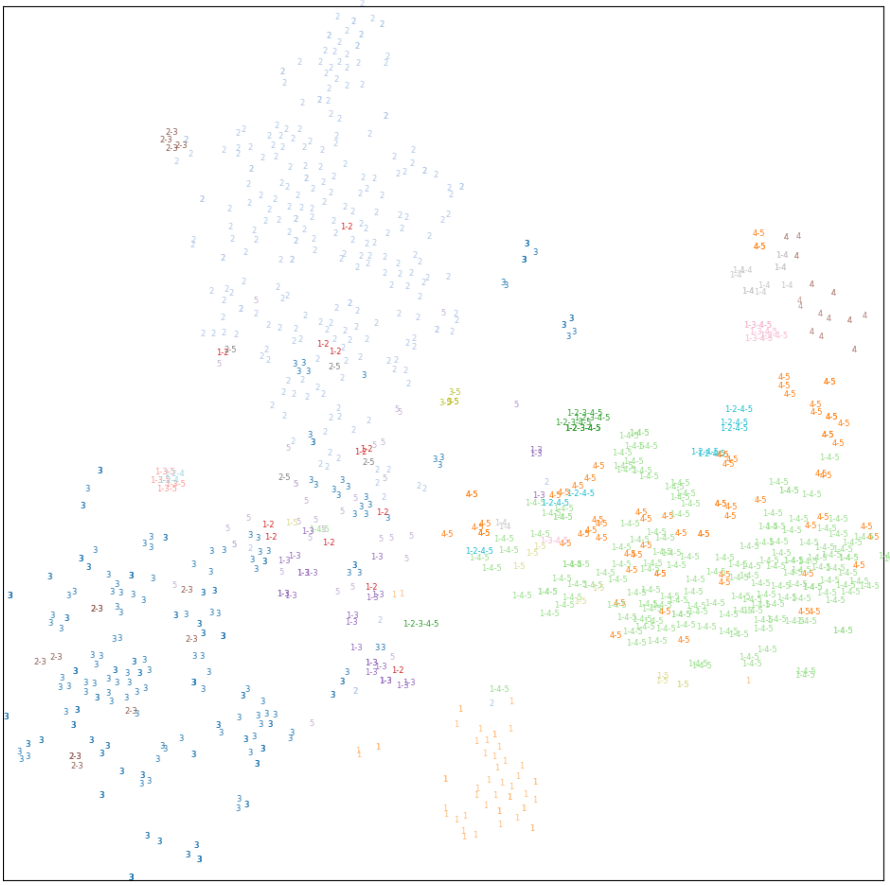


Figure 4.13: t-SNE visualization of style codes from CODEBRIM test images extracted using ViT

doi:10.6342/NTU202303360

unseen style codes.

In addition to these two methods, we considered that although a pre-trained feature extractor has strong feature extraction capabilities, it may not cover all types of images during pre-training. Furthermore, feature extraction, a process of compressing images, could inevitably result in losing some original image features. Thus, we introduced (iii) a label embedding to supplement the original style code, leading to a minor improvement in performance.

During inference, we also employed a proposed (C) style space sampling technique. This involves determining the distribution of the style code for each label during training, such that during inference, as long as the style code space is continuous, any style code extracted from it should be usable. It is observed that this not only allows us to increase the diversity of generated images by adjusting the multiplier of the distribution's standard deviation within a safe range but also reduces the FID, mitigates bias, and enhances the quality of generated images.

		Train from Scratch		MAE-GAN Pre-trained	
Train	**Inference**	**FID**	**LPIPS**	**FID**	**LPIPS**
Baseline : DefectGAN		86.02	–	65.82	–
(i) Single Style Image	(A) multiple style images	103.00	0.0612	76.21	0.0522
	(B) single style image	98.95	0.1392	73.16	0.1397
	(C) style space sampling	90.68	0.1420	69.15	0.1418
(ii) Multiple Style Images	(A) multiple style images	89.63	0.0613	65.85	0.0609
	(B) single style image	89.20	**0.1475**	64.72	0.1534
	(C) style space sampling	87.13	0.1214	63.65	0.1749
(iii) +Label Embedding	(A) multiple style images	87.48	0.0610	64.42	0.0628
	(B) single style image	86.50	0.1419	63.90	0.1450
	(C) style space sampling	**85.50**	0.1127	**61.05**	**0.2239**

Table 4.6: Comparison of downstream performance with different methods of obtaining style codes using the SEAN block.

4.6.3 Style Space Sampling

Finally, Table 4.7 illustrates the rate of defective image generation under different standard deviation (std) multipliers. The depth of color signifies the probability

doi:10.6342/NTU202303360

of generating a defective image: white indicates no defects, and the lightest shade represents less than a 1% defect rate, which is essentially negligible. For models trained from scratch, the color at a 4x std signifies a 50% defective image rate. Under typical usage, utilizing a 2x std multiplier is safe. After pre-training with MAE-GAN, the style code space is even more well-concentrated, allowing the use of up to a 4x std multiplier for style code retrieval. For our final choice of std multiplier, we chose a value that produced defects, but at a very low rate. This corresponds to a 2x std for models trained from scratch and a 4x std for models pre-trained with MAE-GAN. This ensures that under normal circumstances, optimal image quality and diversity are achieved while maintaining the generation of defect-free images.

	Train from Scratch		MAE-GAN Pre-trained	
Different STD Multipliers	**FID**	**LPIPS**	**FID**	**LPIPS**
Single Image Reference	86.50	0.1419	63.90	0.1250
x 1 STD	95.14	0.0536	65.88	0.0537
x 2 STD	89.50	0.1127	63.85	0.1235
x 4 STD	77.81	0.1896	61.05	0.2239
x 6 STD	76.81	0.1853	63.31	0.2831

Table 4.7: Defective image generation rates for different standard deviation multipliers in style space sampling.

doi:10.6342/NTU202303360

Chapter 5

Conclusion

This chapter summarizes the contributions of our thesis and outlines potential future enhancements to our system.

5.1 Summary and Contribution

We propose two methods, MAE-GAN and the SEAN block, which can achieve stable training, enhance the realism of generated images, and increase the diversity of generated images for Unsupervised Image-to-Image (I2I) tasks with limited data. Our methods have been validated through experiments on a less-explored concrete defect bridge image dataset, demonstrating their applicability in less common real-world scenarios. Moreover, on the widely used AFHQ dataset, our methods can generate images of quality comparable to those produced with full data, while only utilizing 10% of the data, and they also yield a higher diversity of images.

5.2 Future work

Our future work can be divided into three main areas:

doi:10.6342/NTU202303360

5.2.1 Pre-trained Feature Extractor Selection

We currently use the commonly used Vision Transformer (ViT) [40] as our large-scale pre-trained feature extractor. Yet, there are many other powerful feature extractors in computer vision [30, 45] and Vision Language Models [46] that can extract useful features. Exploring these options or even using multiple feature extractors to get richer features could be a promising direction.

5.2.2 Supervised Image-to-Image Translation

Our MAE-GAN pre-trains Unsupervised I2I models to get style information from different domains. But it's not yet ready for Supervised I2I tasks like Labels-to-Photos. Adapting the MAE-GAN framework for different I2I tasks is an interesting future research direction.

5.2.3 Integration with Data Augmentation Methods

Recent work on Data-Efficient Unsupervised I2I translation [1, 2] has focused on data augmentation techniques, a different category from our pre-training method. These two categories can complement each other, as shown in previous work [3]. But without executable code from these papers, we haven't integrated these methods with ours yet. It's an open question whether such integration could improve results, even surpassing the state-of-the-art.

doi:10.6342/NTU202303360

Bibliography

[1] J. Cao, L. Hou, M.-H. Yang, R. He, and Z. Sun, "Remix: Towards image-to-image translation with limited data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15018–15027, 2021.

[2] J. Cao, M. Luo, J. Yu, M.-H. Yang, and R. He, "Scoremix: A scalable augmentation strategy for training gans with limited data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[3] Z. Li, X. Wu, B. Xia, J. Zhang, C. Wang, and B. Li, "A comprehensive survey on data-efficient gans in image generation," *arXiv preprint arXiv:2204.08329*, 2022.

[4] Y. Wang, C. Wu, L. Herranz, J. Van de Weijer, A. Gonzalez-Garcia, and B. Raducanu, "Transferring gans: generating images from limited data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 218–234, 2018.

[5] S. Mo, M. Cho, and J. Shin, "Freeze the discriminator: a simple baseline for fine-tuning gans," *arXiv preprint arXiv:2002.10964*, 2020.

[6] T. Grigoryev, A. Voynov, and A. Babenko, "When, why, and which pretrained gans are useful?," *arXiv preprint arXiv:2202.08937*, 2022.

doi:10.6342/NTU202303360

[7] L. Yu, J. van de Weijer, *et al.*, "Deepi2i: Enabling deep hierarchical image-to-image translation by transferring from gans," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11803–11815, 2020.

[8] Y. Wang, H. Laria, J. van de Weijer, L. Lopez-Fuentes, and B. Raducanu, "Transferi2i: Transfer learning for image-to-image translation from small datasets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14010–14019, 2021.

[9] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10551–10560, 2019.

[10] 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*, pp. 2524–2534, 2021.

[11] K. Cui, J. Huang, Z. Luo, G. Zhang, F. Zhan, and S. Lu, "Genco: generative co-training for generative adversarial networks with limited data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 499–507, 2022.

[12] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *Advances in neural information processing systems*, vol. 33, pp. 12104–12114, 2020.

[13] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7559–7570, 2020.

[14] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," *arXiv preprint arXiv:1910.12027*, 2019.

doi:10.6342/NTU202303360

[15] Z. Zhao, S. Singh, H. Lee, Z. Zhang, A. Odena, and H. Zhang, "Improved consistency regularization for gans," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 11033–11041, 2021.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[18] Y. Pang, J. Lin, T. Qin, and Z. Chen, "Image-to-image translation: Methods and applications," *IEEE Transactions on Multimedia*, vol. 24, pp. 3859–3881, 2021.

[19] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

[20] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2337–2346, 2019.

[21] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 694–711, Springer, 2016.

[22] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

doi:10.6342/NTU202303360

[23] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

[24] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3106–3121, 2019.

[25] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 35–51, 2018.

[26] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 172–189, 2018.

[27] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, "Pretraining is all you need for image-to-image translation," *arXiv preprint arXiv:2205.12952*, 2022.

[28] G. Parmar, K. K. Singh, R. Zhang, Y. Li, J. Lu, and J.-Y. Zhu, "Zero-shot image-to-image translation," *arXiv preprint arXiv:2302.03027*, 2023.

[29] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[30] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

[31] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings*

doi:10.6342/NTU202303360

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9653–9663, 2022.

[32] Z. Fei, M. Fan, L. Zhu, J. Huang, X. Wei, and X. Wei, "Masked auto-encoders meet generative adversarial networks and beyond," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24449–24459, 2023.

[33] C. Wei, K. Mangalam, P.-Y. Huang, Y. Li, H. Fan, H. Xu, H. Wang, C. Xie, A. Yuille, and C. Feichtenhofer, "Diffusion models as masked autoencoders," *arXiv preprint arXiv:2304.03283*, 2023.

[34] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[35] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4432–4441, 2019.

[36] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan++: How to edit the embedded images?," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8296–8305, 2020.

[37] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2287–2296, 2021.

[38] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8188–8197, 2020.

doi:10.6342/NTU202303360

[39] J. Huang, K. Cui, D. Guan, A. Xiao, F. Zhan, S. Lu, S. Liao, and E. Xing, "Masked generative adversarial networks are data-efficient generation learners," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2154–2167, 2022.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[41] M. Mundt, S. Majumder, S. Murali, P. Panetsos, and V. Ramesh, "Codebrim: Concrete defect bridge image dataset," 2019.

[42] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797, 2018.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

doi:10.6342/NTU202303360

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

doi:10.6342/NTU202303360