Real-world Image Noise Reduction through Attention Mechanisms in Generative Adversarial Networks

Abdulhafiz Kemal Ahmed

A Thesis submitted to the Department of Computer Science and Engineering, School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September 2024
Adama, Ethiopia

Real-world Image Noise Reduction through Attention Mechanisms in Generative Adversarial Networks

Abdulhafiz Kemal Ahmed

Advisor: Mesfin Abebe (Ph.D.)

A Thesis submitted to the Department of Computer Science and Engineering, School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September 2024
Adama, Ethiopia

Declaration

I hereby declare that this Master's Thesis entitled "**Real-world Image Noise Reduction through Attention Mechanisms in Generative Adversarial Networks**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of material that are used for this thesis have been duly acknowledged by proper citations.

Abdulhafiz Kemal	_____	_____
Name of the student	Signature	Date

Recommendation of Advisor

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Real-world Image Noise Reduction through Attention Mechanisms in Generative Adversarial Networks**" prepared under my guidance by **Abdulhafiz Kemal Ahmed** submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering. Therefore, I recommend the submission of the revised version of the thesis to the department following applicable procedures.

Dr. Mesfin Abebe (Ph.D.)	_____	_____
Major Advisor	Signature	Date

Approval Sheet

I the advisor of the thesis entitled "**Real-world Image Noise Reduction through Attention Mechanisms in Generative Adversarial Networks**" developed by **Abdulhafiz Kemal Ahmed,** hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Mesfin Abebe (Ph.D.)	_____	_____
Major Advisor	Signature	Date

Approval of Board of Reviewers

We, the undersigned, members of the Board of Examiners of the thesis by **Abdulhafiz Kemal Ahmed** have read and evaluated the thesis entitled "**Real-world Image Noise Reduction through Attention Mechanisms in Generative Adversarial Networks**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

I would like to begin by thanking **Allah the Almighty** for the blessings of health and strength that have supported me throughout this research.

I am also very grateful to my advisor, **Dr. Mesfin Abebe (Ph.D.)**, for his guidance and encouragement. His insightful advice, encouragement, and dedication were very important for this work to come to completion, and I am deeply appreciative of his mentorship.

Finally, I would like to express my sincere gratitude to my family, the Artificial Intelligence SIG members, and my friends for their motivation and support in my thesis journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SAMPLE CODES

LIST OF ACRONYMS

ANR	Attention Noise Reduction
CA	Channel Attention
CNN	Convolutional Neural Network
DL	Deep Learning
DGAN	Denoising Generative Adversarial Network
DnCNN	Denoising Convolutional Neural Network
DND	Darmstadt Noise Dataset
DRCNNs	Deep Residual Convolutional Neural Networks
DRN	Deep Residual Network
GACs	Generative Adaptive Convolutions
GAN	Generative Adversarial Network
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
SA	Self-Attention
SCGAN	Self-Consistent Generative Adversarial Network
SIDD	Smartphone Image Denoising Dataset
SSIM	Structural Similarity Index Measure

ABSTRACT

Real-world Images often suffer from various types of noise, decreasing their clarity and usefulness. In order to tackle this problem and obtain the original scene from the image the noises that occurred on the image has to be removed or the image has to be denoising. Recent years have experienced progress in the field of image denoising by using the strength of Generative Adversarial Networks (GANs). However, the challenge of retaining original image details while reducing noises from real world images is still ongoing. This research proposed an approach to reduce real-world image noises using GANs enhanced with attention mechanisms, specifically channel attention and self-attention mechanisms. The proposed model, ANR-GAN, is specifically designed to concentrate on relevant features of noisy images, resulting in enhanced noise reduction and better preservation of details. The results we obtained from the experiment demonstrate that our model performs better than the baseline D-GAN model, with a PSNR of 37.42 and SSIM of 0.9582. These values from the metrics indicate that the combined use of CA and SA mechanisms within the GAN framework significantly enhances denoised image's quality. ANR-GAN model not only reduce noise effectively but also maintain the contents of the original images; this makes them highly usable for real-world applications. This research contributes in image denoising field by providing a method that integrates attention mechanisms into GANs, offering a more robust solution for reduction of noises that occur in real-world images. The results suggest that attention mechanisms are crucial in the enhancement of GAN-based denoising model performances.

Keywords: image denoising, generative adversarial network, channel attention, self-attention, real-world images.

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Images are visual representation of objects, scene, person, or concept. Images are used in a wide range of fields such as photography, art, computer vision, design, and medical imaging. Real images are obtained from the real-world using imaging devices such as cameras, and they are rich, complex, and often noisy and textured. They are meant to accurately capture the visual elements of the original scene. Noise in an image are unwanted variations in color or brightness which are not from the underlying scene. Noisy pixels cause strong visual effects in an image by appearing as isolated pixels or blocks of pixels (Gong et al., 2020). Real-world images can be affected by various types of noise, including Gaussian, Poisson, Salt-and-Pepper noise (Tran et al., 2021). To obtain the original, noise-free image, we must apply image denoising or noise reduction techniques. These techniques aim to restore the underlying clean image from its corrupted, noisy counterpart. Image denoising is a crucial task in computer vision (Ma, Li et al., 2022) because it's a necessary preliminary step for many other computer vision tasks, such as image classification, image segmentation, object detection, visual tracking, and more.

As digital imagery becomes increasingly essential in fields like medical diagnostics and photography, it's imperative to guarantee the quality of images captured in real-world environments. Real-world images are often compromised by noise introduced through various sources, such as sensor limitations, environmental factors, and even post-processing. Unlike synthetic images, where noise can be artificially modeled and removed using traditional methods, real-world images are often characterized by noise patterns that are significantly more difficult to address compared to those encountered in controlled settings. The complex nature of real-world noise is influenced by a variety of factors, including fluctuating lighting conditions, sensor non-linearity, and color space transformations. These challenges make real image denoising a persistent in computer vision (Plötz & Roth, 2017).

Traditional image denoising techniques, such as Gaussian smoothing, bilateral filtering, and wavelet transforms, have provided some level of success in reducing noise. However, these methods often introduce unwanted artifacts, blur important image details, or fail to generalize across diverse noise types (Buades et al., 2005a). More machine learning

techniques, including convolutional neural networks (CNNs), have demonstrated improved performance by learning how noisy images can be mapped to their clean counterparts. Nonetheless, CNN-based models face limitations when dealing with complex real-world noise, which often exhibits spatially varying characteristics (K. Zhang et al., 2017). The inherent difficulty in real image denoising stems from the non-uniform nature of real-world noise, which is typically not well-modeled by simple statistical distributions such as Gaussian or Poisson noise.

Currently Deep-learning based techniques perform better in image denoising than traditional approaches such as filter-based techniques struggle with complex noise patterns and lack adaptation to diverse real-world images (Tran et al., 2021). Deep learning methods, especially Generative Adversarial Networks (GANs), have emerged as promising approaches for image denoising. GANs consist of two neural networks, a generator and a discriminator that are trained in a competitive manner to produce images which are realistic with high-quality. GANs are composed of two neural networks, a generator, and a discriminator, which are trained together in a competitive process to generate high-quality, realistic images. GANs have been successfully used in many image processing tasks, which include image restoration and image denoising, and have shown potential for generating high-fidelity, noise-free images (Z. Wang et al., 2020).

The Generative Adversarial Network (GAN), introduced by Ian Goodfellow and his colleagues in 2014, is a machine learning framework that employs two neural networks: a generator and a discriminator. These networks engage in a cooperative zero-sum game, where the generator creates new examples, and at the other end the discriminator attempts to classify them as real or fake. GANs have demonstrated remarkable success in various applications, including image-to-image translation, style transfer, as well as generating highly realistic photographs. The training process for GANs involves a competitive interaction between the generator and discriminator. The discriminator learns to distinguish between real data and fake data generated by the generator. Simultaneously, the generator is trained to produce samples that can deceive the discriminator (I. Goodfellow et al., 2020; Karras et al., 2017). The training in adversarial manner motivates the generator in creating highly realistic outputs, making GANs especially valuable for image enhancement tasks that require preservation of fine details (Y. Wang et al., 2018).

The introduction of GANs we first made by the influential work (I. J. Goodfellow et al., 2014) as a powerful framework for generative modeling, particularly in generating realistic

high-resolution images. The study by (Karras et al., 2017) on showcased the improved quality and stability of GAN-generated images, highlighting the potential for GANs in image-related tasks. Additionally, research by (Radford et al., 2015) demonstrated effectiveness of GANs in learning representations from unlabeled data, laying the groundwork for their application in denoising real-world images.

The SIDD dataset, which consists of real noisy images captured using modern smartphone cameras, provides a realistic benchmark for evaluating denoising models (Abdelhamed et al., 2018). Unlike synthetic datasets where noise is artificially added, SIDD represents real-world challenges such as sensor noise and color artifacts, this makes it perfect choice for evaluating the effectiveness of our proposed model. Previous studies using this dataset have demonstrated the limitations of traditional and even deep learning-based approaches, further emphasizing the need for innovative techniques like attention-augmented GANs for superior noise reduction (Y. Zhang et al., 2018).

A recent study proposed a GAN-based model specifically designed for denoising real world images. This model aimed for the performance of the denoiser networks to be enhanced when applied to real-world noise reduction. The model was trained using real and synthetic noise extracted from raw images. The generated noise was then used to create image dataset to train neural network. This innovative approach showed the potential to enhance the effectiveness of existing denoising methods, highlighting a promising nature of GAN-based approaches in addressing real image noise reduction (Tran et al., 2021).

While GANs have proven to be powerful tools in image denoising, integrating attention mechanisms, particularly channel attention and self-attention, presents a promising avenue for improving noise reduction capabilities in real-world images. By leveraging the SIDD dataset and the attention-augmented GAN architecture, this study aims to add some potential improvements in current image denoising techniques, ultimately contributing to the broader field of image enhancement in practical applications.

1.2. Motivation of the Study

Real-image denoising remains an important yet challenging task in image processing. Traditional methods often struggle to preserve delicate details while removing noise, particularly when dealing with complex noise patterns. This motivates the exploration of the GANs for image restoration.

Image denoising is a critical component in various fields, which include, medical imaging, photography, and surveillance, where the quality of visual data is very important. Generative Adversarial Networks (GANs) exhibited exceptional capabilities in generating high-resolution, quality images (Karras et al., 2017) . Furthermore, their potential for image denoising has shown promising results Moreover, their potential for image denoising has yielded encouraging outcomes (Gong et al., 2020; Tran et al., 2021; Z. Wang et al., 2020) By enhancing the strength of GANs with attention mechanisms for image denoising, this study aims to address the limitations that appear in denoising techniques and enhance the quality of denoised images.

Enhancing GANs with attention mechanisms unlocks even greater potential. Attention mechanisms give the model the ability to focus on important features within the image, improving noise removal accuracy and preserving crucial details. Additionally, they can capture long-range dependencies, leading to more consistent and natural-looking denoising results.

There is great promise for transforming image denoising into a more accurate and adaptable process through the integration of GAN with attention mechanisms. This approach has the potential to tackle complex noise patterns, handle diverse image types, and deliver superior restoration quality. Research in this exciting area can not only push the boundaries of image processing but also substantially advance the overall development of GAN.

1.3. Statements of the Problem

Image denoising is an important issue in computer vision, especially for image processing applications. There are a number of factors that might cause image noise in real world images, including low lighting, interference during transmission and sensor limits. The visual quality of images may be reduced by this noise, making it more challenging to determine important information from them. To reduce undesired noise and enhance image quality, it is important to come with precise and effective image denoising techniques.

Recent advancements in deep learning approaches have demonstrated progress in denoising real-world images (Gong et al., 2020; Ma et al., 2022; Tran et al., 2021; Z. Wang et al., 2020). However, many existing denoising methods struggle to balance between effectively removing noise and preserving crucial features of images, which is critical to avoid loss of information during the denoising process (K. Zhang et al., 2017). developing an efficient

deep learning-based image denoising model needs addressing issues, including choosing a suitable network architecture.

The objective of reducing noises is to minimize noise appearing in real images while preserving original features and enhancing overall image quality (Lan et al., 2021). In order to achieve robust image denoising technique we have to consider dynamic noise adaptation and long-range dependencies; noise often exhibits complex spatial relationships. The attention model should capture these long-range dependencies for comprehensive noise removal, and different noise types and levels require different attention strategies. In order to achieve robust denoising algorithm we integrated channel and self attention mechanisms in the generator network of the generative adversarial networks framework so that they can increase the potential of the networks.

1.4. Research Question

This study attempts to address the following research questions.

RQ1: what are the effects of integrating attention mechanisms in GAN denoising process for real-world images?

RQ2: What are specific learning constraints that can enhance the output of our image denoising model?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this study is to develop a model for real-world image noise reduction by integrating attention mechanisms in generative adversarial network.

1.5.2. Specific Objectives

The specific objectives stated below were considered for the achievement of the general objective.

- To review related works done on the area of image denoising.
- To Collect and preprocess publicly available dataset that contain real-world image noise.
- To design and implement a GAN model by incorporating channel and self-attention mechanisms in the generator network to enhance denoising performance.

- To train the proposed attention-enhanced GAN model on the collected dataset, optimizing it for noise reduction in real-world images.
- To evaluate how the model performs using PSNR and SSIM evaluation metrics.

1.6. Significance of the Study

Real-world image denoising is very important task which is recovering clean images from noisy images. This has potential to enhance the quality of digital images and also it can also facilitate computer vision tasks. By reducing the noises that can be found in real-world images and producing clearer image this study can help machines and computers recognize and understand the visual information in images such as faces, objects scenes and actions. This can facilitate many computer vision tasks including image classification, object detection, face recognition, image restoration, image segmentation and visual tracking.

1.7. Scope and Limitations of the Study

1.7.1. Scope of the Study

This research focuses on developing a generative adversarial network (GAN) model capable of effectively reducing noise in real-world images by incorporating attention mechanisms. SIDD dataset containing real-world noisy images is utilized for training and evaluation. Specifically, the research focuses on enhancing the denoising capabilities of GANs by integrating channel attention and self-attention in the encoder and decoder of the generator network respectively. The proposed model is evaluated against previous approach to determine its effectiveness in preserving image details while minimizing noise. The study covers the development, implementation, and evaluation of this GAN-based approach, on real-world noisy image dataset.

1.7.2. Limitations of the Study

Because of time and resource limitations this work does not include the following:

- It only considers image noise reduction, it does not involve image super-resolution
- It only considered integrating channel and self-attention mechanism to the GAN, not other attention mechanisms.

1.8. Contribution of the Study

Our focus was to improve the GAN networks by incorporating channel and self-attention mechanisms into it.

- Improving the quality of denoised image by incorporating attention mechanisms, that helps the denoising model to selectively focus on features of the image that contain important details or that are most affected by noise.
- Reducing the artifacts that may be introduced during the denoising process and over smoothing, by utilizing the pixel wise loss.

1.9. Organizations of the Study

This thesis's remaining sections are organized in the following order:

Chapter Two: The literature review and related works in the area image denoising are discussed in this chapter. Deep Learning and Generative Adversarial Network frameworks are also discussed.

Chapter Three: Includes the research methodology, which includes the various approaches and procedures needed to develop the model. Methodologies for collecting data, preprocessing the data, designing and developing frameworks and tools, and evaluating methods.

Chapter Four: It provides the description about the proposed architecture of the model, which include designing the generator network and discriminator networks, the loss functions used, and the training algorithm employed.

Chapter Five: describes how the proposed model is implemented. Which include the working environments, proposed approach implementation, specifying hyperparameters utilized, and describing experimental class.

Chapter Six: The results obtained from the proposed approach are discussed and evaluated to relevant approaches to draw conclusions.

Chapter Seven: The final chapter presents the conclusions drawn from this research, along with future studies recommendations.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Overview of Image Denoising

An image is worth a thousand words. Images play important and practical role in communication. The rapid development of imaging technology and mobile devices has led to the creation, translation, and alteration of a significant number of images. But the quality of the images varies with the variety of imaging devices. For instance, mobile cameras generally result in images that are noisy or blurry. Furthermore, when images are transmitted between various devices, some data might get corrupted. Therefore, obtaining meaningful information from noisy images while eliminating noise is one of the most crucial problems of our day (Solangi et al., 2017).

The field of image denoising seeks to enhance the quality of images by eliminating noise while preserving their original structure and details (Gupta et al., 2019). The primary objective is to recover the original image from its noisy counterpart (Pardasani & Shreemali, 2018). However, the challenge lies in distinguishing between noise and high-frequency components like texture and edges, which can lead to unintended loss of details in the denoised images. Effectively removing noise while preserving crucial information from noisy images remains a critical goal in today's image processing landscape (Fan et al., 2019).

More effective and efficient denoising techniques are required as digital images with high-resolution become more widely available. Recent studies on image denoising tasks have shown that DL-based systems perform better than conventional techniques (Cui et al., 2019). Convolutional neural networks (CNNs) are used in deep learning-based image denoising techniques to learn the mapping between noisy and clean images (Hashimoto et al., 2019). The goal of training these networks is to reduce the difference between the output of the network and the corresponding clean picture by using datasets of noisy and clean image pairings. The ability of deep learning-based denoising algorithms to comprehend intricate non-linear relationships between noisy and clean image patches is one of its main benefits.

Numerous deep learning-based models have been introduced for denoising, including the residual network (ResNet), generative adversarial network (GAN), and basic CNN models. In order to learn mapping between clean images and noisy images the ResNet utilizes skip connections, which improves training efficiency and enhances denoising performance

compared to the standard CNN model that maps directly from noisy to clean images. The GAN model, on the other hand, incorporates a generator network to produce denoised images and a discriminator network to distinguish between real and generated images, leading to more realistic image generation and improved denoising results (Fu et al., 2022, 2023).

2.2. Image Denoising Techniques

2.2.1. Traditional Image Denoising Techniques

Conventional image denoising methods have been fundamental in image processing, focusing on noise removal while maintaining critical image features like edges, textures, and fine details. Early approaches, such as spatial filtering techniques, work directly on the pixel values of an image. For instance, the mean filter reduces noise by averaging the pixel values within a local area surrounding each pixel. While this method is effective, it often results in the loss of fine details and can cause blurring of edges (Gonzalez et al., 2009). In comparison, the median filter offers greater robustness, especially against impulse noise, by replacing each pixel with the median value from its surrounding neighborhood. This approach better preserves edges compared to the mean filter (Vyas et al., 2018). Another commonly used spatial filter is the Gaussian filter, which applies a Gaussian function to weigh the pixel values, giving more importance to the central pixels and effectively reducing Gaussian noise, albeit at the cost of some blurring (Bovik Alan, 2005). Bayesian methods model the denoising process as an estimation problem, utilizing prior knowledge about the image and noise characteristics. For instance, Markov Random Fields (MRFs) are often employed to model spatial dependencies among pixels (Buades et al., 2005b). The NLM algorithm, on the other hand, substitutes each pixel with an average weighted of corresponding pixels found throughout the image, making it particularly effective in preserving textures and details in images with repetitive patterns (Elad & Aharon, 2006).

2.2.2. Deep Learning-Based Denoising Algorithms

a convolutional neural network (CNN) was introduced by (K. Zhang et al., 2017) model that combines batch normalization with residual learning techniques. While this approach achieved positive results, it required a large number of iterations to obtain an optimal training model, which in turn affected the algorithm's efficiency and convergence speed.

Deep learning-based denoising techniques have transformed the image processing field by utilizing neural networks to directly learn complex representations and features from data. in contrast to conventional methods that depend on predefined filters or transformations,

deep learning approaches can autonomously learn the noisy to clean images mapping through end-to-end training.

The Convolutional Neural Network (CNN) is one of the most widely used architectures for image denoising. CNN-based models generally comprise many convolutional layers that capture both global and local image features. These models are trained on extensive datasets of noisy and clean image pairs, enabling the network to learn how to minimize the difference between the denoised output and the ground truth clean image. A notable example is the DnCNN model, which has demonstrated excellent effectiveness in removing additive white Gaussian noise (K. Zhang et al., 2017) . DnCNN employs residual learning to directly predict the noise component, it creates the clean image after being removed off of the noisy image.

Another significant method involves Autoencoders, neural networks specifically designed to learn efficient data representations (encodings), often used for dimensionality reduction. In denoising, a variant known as the Denoising Autoencoder (DAE) is utilized, where a noisy input is used to train the network to rebuild a clear image. The image is compressed by the encoder into a representation of lower-dimension, which the decoder then uses to reassemble the original image. This technique works very well for eliminating noise from images while keeping the important details (Vincent et al., 2008).

Generative Adversarial Networks (GANs) are other approaches which has been utilized to the denoising problem with remarkable success. In GAN-based denoising, the generator produces clean images from noisy inputs, and the discriminator tries to distinguish between generated and clean images. The competitive nature of GANs forces the generator to generate high quality denoised images. A well-known application of GANs for denoising is the Noise2Noise approach, where the model trains only on pairs of noisy images without requiring clean ground truth images. This technique has demonstrated that GANs can effectively learn to denoise images even when only noisy data is available (Lehtinen et al., 2018).

Long Short-Term Memory (LSTM) networks are one type of version of Recurrent Neural Networks (RNNs) that have also been utilized in image denoising, especially in cases involving sequential data or videos. RNNs are adept at capturing temporal dependencies, which makes them well-suited for tasks where the properties of the noise change with time or where maintaining temporal consistency is crucial (Xie et al., 2012)

Beyond these methods, attention mechanisms have seen significant interest in deep learning-based denoising models. Attention mechanisms enable the model to concentrate on critical regions of the image while disregarding irrelevant or noisy areas. This approach has resulted in enhanced denoising performance, particularly in complex situations where noise is inconsistent or varies spatially (Vaswani et al., 2017).

(Yan et al., 2019) directly removed noise artifacts from noisy images by developing a model unsupervised noise approach that reduces noise in unpaired noisy images. The architecture is based on the self-consistent generative adversarial network (SCGAN) model. (Tran et al., 2021) proposed a GAN-based model which trains on both synthetic and real noise, thereby creating a paired-image, which are clean-noisy pair dataset to train a denoising deep neural network. This method enhanced the performance of denoising deep neural networks on real images.

The state-of-the-art in image denoising has been substantially upgraded by these deep learning-based algorithms, which offer strong tools for creating high-quality images for a variety of applications, such as remote sensing, photography, and medical imaging.

2.2.3. Denoising Autoencoders

Denoising autoencoders (DAE) have been widely explored for image denoising tasks due to their ability to reconstruct clean images from corrupted inputs. In a typical DAE, the encoder compresses the noisy image into a lower-dimensional representation, and the decoder reconstructs the clean image. (Vincent et al., 2008) first introduced DAEs to learn robust representations, effectively removing noise from input data. Since then, DAEs have demonstrated success in image denoising applications by learning to minimize the pixel-wise difference between noisy inputs and ground truth images (K. Zhang et al., 2017).

Several improvements have been made to enhance DAE performance. (Mao et al., 2016) proposed the residual DAE to improve learning efficiency, while (Lehtinen et al., 2018) introduced deep convolutional denoising autoencoders for more complex image structures. These models have shown significant improvements in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM), which are standard metrics in image quality assessment.

While DAEs are effective in learning pixel-level mappings for image denoising, they often fail to capture complex textures and structural details of images, especially when dealing with real-world noise. Denoising autoencoders focus on minimizing the pixel-wise error,

which can result in oversmoothed outputs. As a result, fine details and textures that are critical for high-quality image restoration may be lost (Isola et al., 2017).

Generative Adversarial Networks (GANs), on the other hand, offer a more sophisticated approach to image denoising. GANs consist of two networks: a generator that learns to produce denoised images and a discriminator that distinguishes between real and generated images. This adversarial setup encourages the generator to produce images that are not only clean but also visually realistic by preserving high-frequency details. Furthermore, GAN-based models can integrate additional loss functions, such as pixel-wise loss and perceptual loss, allowing for a better balance between detail preservation and noise removal (Ledig et al., 2017). Recent advancements in GANs, such as the inclusion of attention mechanisms, have further improved their ability to focus on salient image regions, resulting in higher-quality outputs (Y. Zhang et al., 2019).

Therefore, compared to DAEs, GANs are more suited for real-world image denoising tasks, where maintaining structural details and achieving visually realistic outputs are critical. This makes GANs more aligned with the goals of your research in addressing complex noise patterns through attention mechanisms.

2.3. Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN) is a deep learning model designed to generate new data that closely resembles existing data by employing adversarial training to produce high-quality outputs from its generator and discriminator models within a framework. This network exemplifies an unsupervised learning approach, commonly applied to tasks such as realistic image generation, image restoration, and image denoising. GANs were introduced in a paper by Ian Goodfellow and colleagues, titled Generative Adversarial Networks (I. J. Goodfellow et al., 2014).

Two neural networks build the GANs:

1. The Generator network which is tasked with producing new data samples. It starts with random noise and creates data that should closely resemble real data.
2. The Discriminator network's role is to assess the given data sample is generated or real. It is trained using real data to differentiate between real and generated samples.

The networks are trained in competitive manner. The generator creates data which resemble the realistic data, so that it can fool the discriminator, on the other hand the discriminator

network builds its knowledge to recognize fake data. The adversarial process pushes both networks to improve their performance, and ultimately leading the generator being able to generate data samples which are highly realistic. GANs have been applied to generate various types of data, include which, videos, images, text, and audio. Notable uses of GANs include creating realistic images of faces, landscapes, and objects, restoring damaged images and videos, generating text that mimics human writing, and composing music that resembles human creation. In the context of image denoising, GANs enhance their capabilities through adversarial training, where both networks continuously improve. The generator becomes more adept at removing noise while retaining fine details, and the discriminator improves its capability to detect even the most subtle noise artifacts.

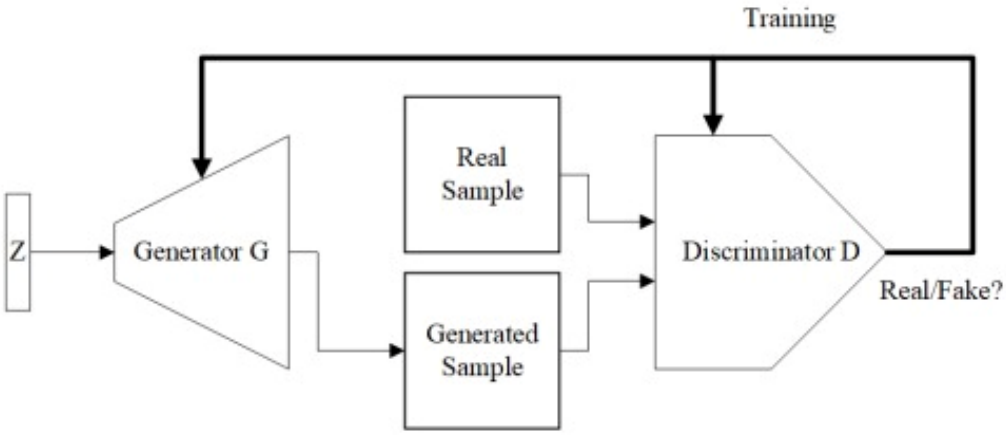


Figure 2-1: Architecture of GANs

Source: Adapted from (Öngün & Temizel, 2019)

In the GAN architecture, Generator (G) is neural network that acts as a counterfeit artist, tasked with creating realistic-looking samples that mimic those in the real dataset. It produces generated samples that aims to closely resemble the samples in the real dataset by taking random noise (z) as input and. Discriminator (D) is neural network which plays the role of a discerning art critic, used to differentiate between genuine samples of the dataset and those created by the G. It receives both real and generated samples as input and produces a probability representing the realness of samples.

2.3.1. GAN Training

Training Generative Adversarial Network involves two-step process that continuously refines both networks. (I. Goodfellow et al., 2020) states during the training of generator, the discriminator's weights are held constant. This ensures that the generator receives a consistent benchmark for evaluation, allowing it to focus on improving its ability to produce

images that are realistic which can deceive the discriminator. Then the generator's weights are updated based on the discriminator’s feedback. This feedback includes the classification of the generated images as fake or real by the discriminator, as well as the gradients calculated through backpropagation. By modifying its weights, the generator produces images that are increasingly capable of deceiving the discriminator. During the discriminator's training phase, the generator's weights are kept constant. This approach prevents the discriminator from altering its evaluation criteria in response to the generator's evolving abilities, ensuring a steady benchmark for distinguishing between real and fake images. Subsequently, the discriminator's weights are adjusted according to its effectiveness in classifying both authentic and generated images. The objective is to improve its capacity to accurately differentiate between the two types of samples, thereby making it more challenging for the generator to mislead it. (Brock et al., 2018; I. Goodfellow et al., 2020; Miyato et al., 2018).

Training the Generator: The goal of the generator is to produce data that closely resembles real data. The training procedure for the generator includes the following steps:

- The generator takes input data or random noise and turns it into synthetic data.
- The data generated is passed through the discriminator.
- The discriminator provides feedback on how well it can differentiate between real and fake data.
- The generator uses this feedback to adjust its parameters through backpropagation, aiming to produce more realistic data in the next iteration.

Training the Discriminator: The discriminator's task is to accurately identify data as either real or generated. The training procedure for the discriminator includes the following steps:

- The discriminator receives a combination of real and generated data.
- generated data is labeled as "fake" and Real data is labeled as "real,"
- The discriminator trains to correctly classify fake and real data.
- The discriminator's parameters are adjusted through backpropagation to improve its ability to differentiate between real and generated data.

The generator and discriminator are trained either simultaneously or sequentially, with the generator working to enhance its ability to deceive the discriminator, while the discriminator hones its skill in differentiating between real and fake data.

2.4. Architectures of Generative Adversarial Network

2.4.1. Deep Convolutional GAN (DCGAN)

DCGANs mark a notable advancement in GAN architecture, especially in terms of improving the stability and quality of the generated images (Jenkins & Roy, 2024). Deep Convolutional Generative Adversarial Networks (DCGANs) represent a significant improvement in the architecture of generative adversarial networks (GANs), particularly in enhancing stability and the quality of generated images. Unlike the original GAN architecture, which relied on fully connected networks, DCGANs leverage deep convolutional neural networks (CNNs) in both the generator and discriminator. The generator network utilizes transposed convolutional layers to upscale the input noise vector into a fully-sized image. In contrast, the discriminator network employs regular convolutional layers to differentiate between real and generated images. DCGANs further enhance their performance by replacing traditional pooling layers with strided convolutions in the discriminator and strided transposed convolutions in the generator. This innovative approach allows the networks to learn their own methods of spatial down sampling and upscaling, resulting in improved image quality and more stable training.

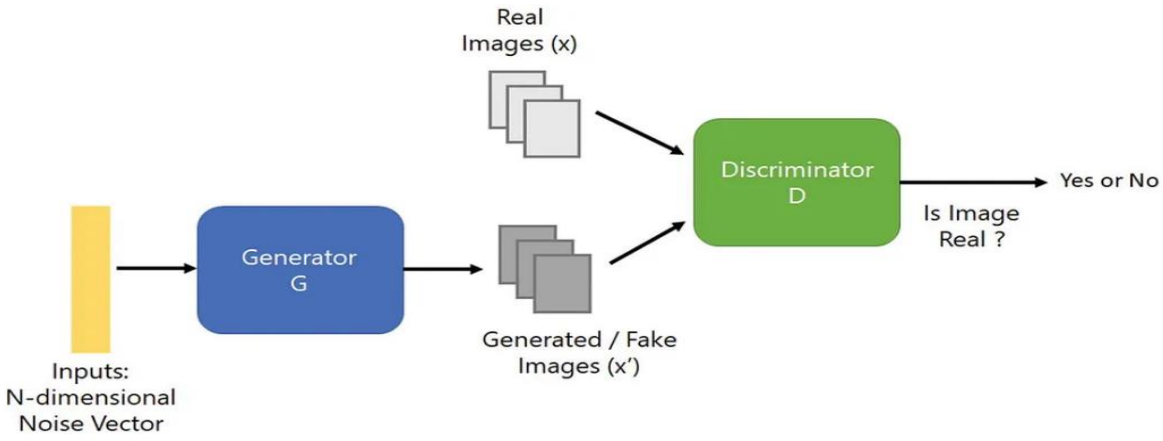


Figure 2-2: Architecture of DCGAN

Source adopted from: (Radford et al., 2015)

2.4.2. Conditional GAN (CGAN)

Generative Adversarial Networks (GANs) are powerful frameworks designed to train generative models capable of creating new, realistic samples from a given dataset. While GANs have shown great potential in generating data that closely resembles real samples, they originally lacked a mechanism for controlling the specific type of data being generated. To address this limitation, Conditional GANs (CGANs), introduced by (Mirza & Osindero,

2014) , Conditional GANs (CGANs) build upon the fundamental GAN architecture by integrating extra information, like class labels, into the process of generation. This conditioning enables the generator to create images that match the specific characteristics defined by the class labels. into the generation process. This conditioning allows the generator to produce images that are aligned with the desired characteristics specified by the class labels. Class label information is incorporated into both the generator and discriminator networks, as depicted in figure 2.3, ensuring that the generated outputs adhere to the specified conditions. This enhancement in GAN technology has paved the way for new opportunities in fields such as image-to-image (I2I) translation, style transfer, and image deblurring, where precise control over image generation is crucial.

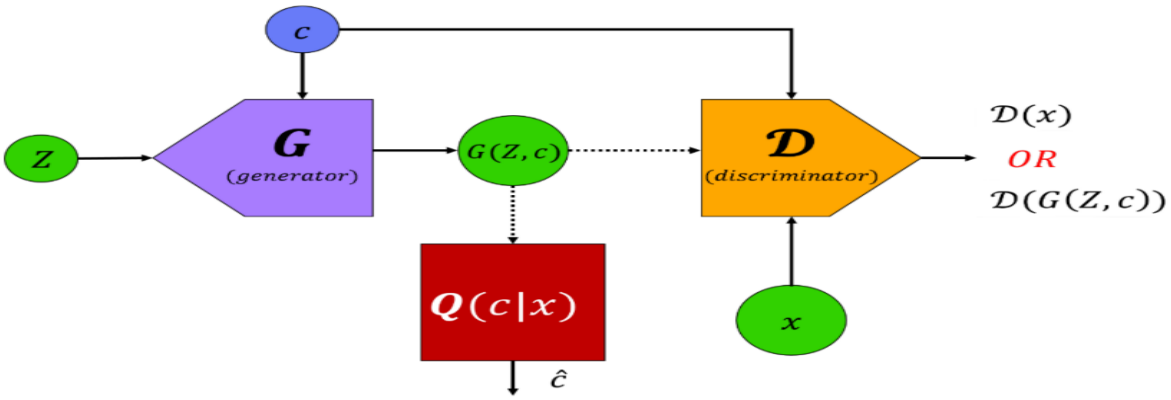


Figure 2-3: Architecture of CGAN

Source adopted from: (Park et al., 2021)

2.4.3. CycleGAN

Using a training set of matched images, image-to-image translation aims to discover the association between an input image and an output image. However, paired data like this is often not available in many real-world situations. To overcome this limitation, CycleGAN (Zhu et al., 2017) introduced a technique for "unpaired image-to-image translation", enabling the model to learn how to convert images from a source domain *A* to a target domain *B* without needing corresponding paired examples. CycleGAN achieves this by learning two mappings: *G: A→B* and *F: B→A*. A significant innovation of CycleGAN is the cycle consistency loss, which ensures that these mappings are inverses of each other, enforcing *F(G(a)) ~ a* and *G(F(b)) ~ b*. By integrating this cycle consistency loss with adversarial losses in both domains A and B, CycleGAN effectively facilitates unpaired image-to-image translation, making it a valuable tool for tasks lacking paired training data.

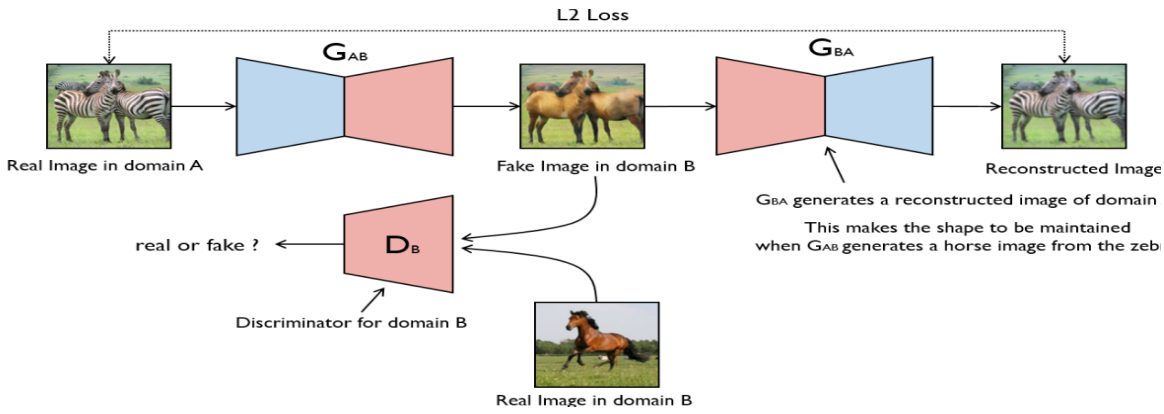


Figure 2-4: Architecture of CycleGAN

Source adopted from: (Zhu et al., 2017)

2.5. U-Net Architecture

The U-Net architecture is known for its symmetric structure, featuring an encoder (contracting path) and a decoder (expanding path). The encoder progressively reduces the spatial size of the input image while capturing high-level features through successive convolutional and pooling layers. Meanwhile, the decoder increases these features back to the original resolution using transposed convolutions, thereby reconstructing the image with enhanced details. A key innovation of U-Net is the use of skip connections, which directly link corresponding layers between the encoder and decoder. These connections facilitate the combination of features with high-resolution from the encoder with the upsampled features in the decoder (Ronneberger et al., 2015).

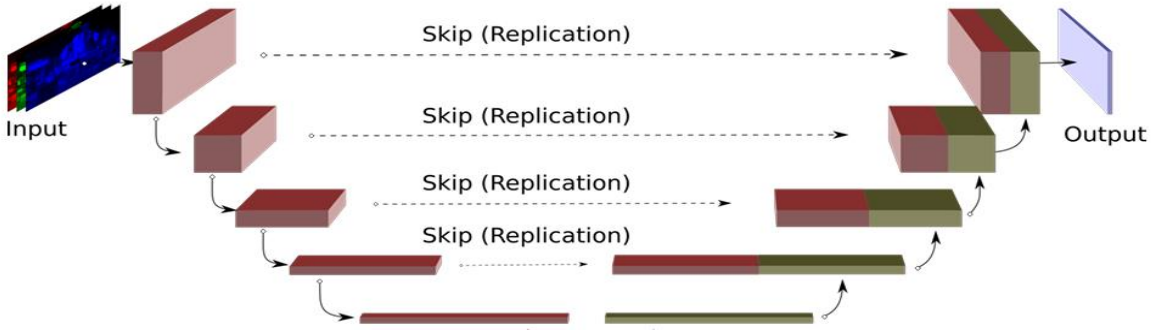


Figure 2-5: Architecture of U-Net

Source adopted from (Marnerides et al., 2020)

2.6. Image-to-Image Translation

Image-to-image translation with GANs has been employed in various applications, such as style transfer, super-resolution, and semantic segmentation. For example, (Park et al., 2021)

investigated the use of GANs for sophisticated image translation tasks, showcasing the capability of GANs to produce realistic and high-quality images. The study also underscored the necessity of fine-tuning both networks which are the generator and discriminator networks to enhance performance in particular domains, including medical image synthesis and artistic rendering.

Figure 2-6: Image-to-Image translation

Source adopted from: (Park et al., 2021)

2.7. Attention Mechanisms

The attention mechanism in image denoising enables the model to concentrate on the most crucial features of the image. This is achieved by employing an attention mechanism to direct the neural network during the denoising process (Tian et al., 2020). Various forms of attention mechanisms have been applied in image denoising, including feature attention, self-attention, channel-wise attention, and spatial-attention, among others (Anwar & Barnes, 2019; Y. Wang et al., 2023).

Self-attention: is a specialized attention mechanism that emphasizes the relationships between different regions within an image. It enables the model to comprehend how various areas and features are interconnected, creating a cohesive understanding of the scene. While convolution is effective for capturing local, geometric, or structural patterns, self-attention complements this by addressing long-range, global dependencies. It also serves a similar purpose to local convolution when modeling dependencies related to simple textures. (H. Zhang et al., 2018).

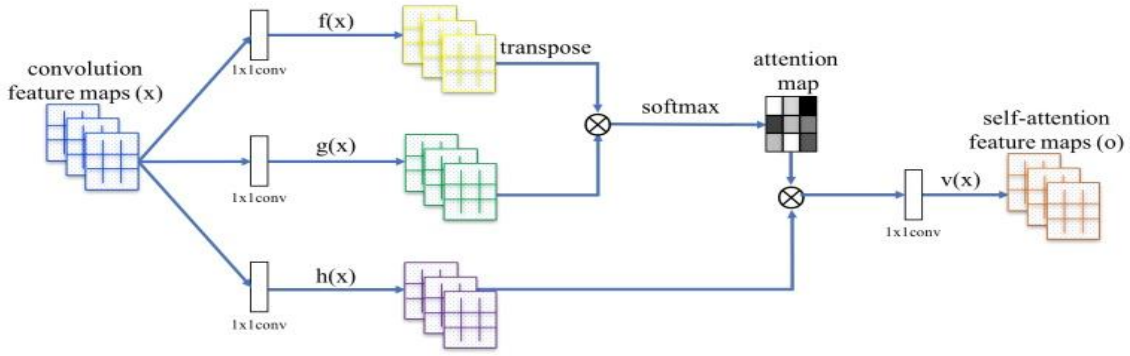


Figure 2-7: Self-attention module

Source: adapted from (H. Zhang et al., 2018)

Spatial and channel attention: Spatial and channel-wise attention are commonly utilized in computer vision, especially in relation to convolutional layers. In convolutional layers, both the input and output are represented as tensors with three dimensions: height (h) for each feature map, width (w) for each feature map, and channels (c), which correspond to the number of feature maps or the depth of the tensor. This is often denoted as (c × h × w). The feature maps can be visualized as slices of this cube-shaped tensor, stacked together. The value of c is determined by the number of convolutional filters used in the layer. Spatial attention enhances the efficiency of neural network learning by identifying and focusing on relevant information within images, thus preventing the loss of important details (Li et al., 2021). Channel attention assigns weights to each channel, thereby amplifying the most influential channels and improving the overall performance of the model (Huang et al., 2023).

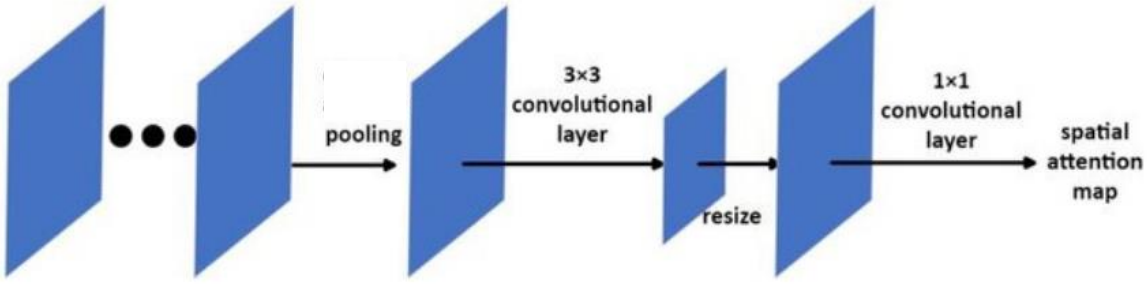


Figure 2-8: Structure of spatial attention

Source: adapted from (H. Zhao & Xu, 2023)

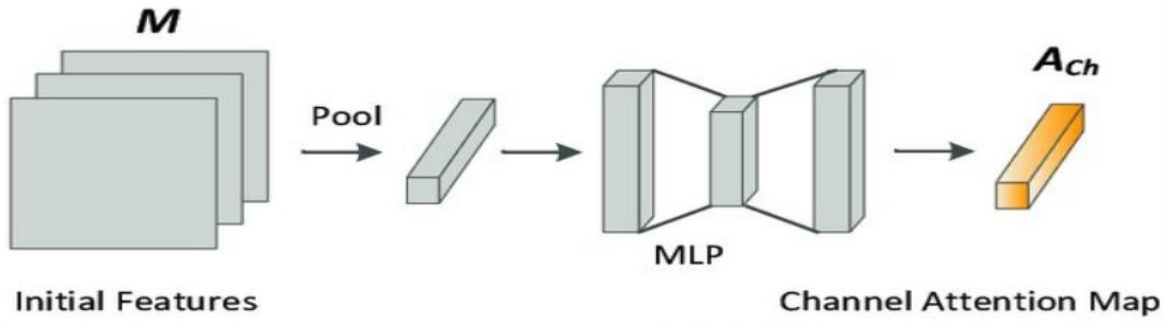


Figure 2-9: structure of channel attention

Source: adapted from (Huang et al., 2023)

2.8. Related Works

(Gong et al., 2020) introduced a GAN-based model featuring an innovative encoder-decoder architecture. In this model, the encoder from the generator extracts features from the noisy image, and the decoder performs noise removal using residual and skip connections. However, the model experienced relatively high color saturation loss.

(Ma et al., 2022) introduced "Generative Adaptive Convolutions" (GACs) that dynamically adjust their filters based on the input noise level. These GACs are used in a U-Net architecture for denoising. The evaluation focus is primarily on noise magnitude, not diverse noise types.

(Tran et al., 2021) proposed an approach by incorporating a GAN with a CNN denoiser, they used paired clean and noisy images. The papers limitations in GAN stability and data efficiency, and the overall denoising performance might be limited by the GAN's noise generation accuracy.

(Z. Wang et al., 2020) came up with a GAN-based model with deep residual connections in both networks of the GAN, the generator and discriminator. They claim these connections improve noise removal. The paper lacks thorough comparisons with existing methods, and the evaluation is limited to a single dataset with specific noise characteristics.

(Alsaiari et al., 2019) the primary idea behind their approach is to generate images using minimal number of samples per pixel and then feed the noisy image into the network, which aims to produce a high-quality, photorealistic output. A key aspect of their study is the defined loss function and the very deep GAN, which are essential to their method. They introduced an updated perceptual loss that not only preserves color and texture but also captures scene features like motion blur and depth of field.

(Lan et al., 2021) proposed approach presents a promising solution for image denoising using deep residual convolutional neural networks (DRCNNs). The core structure of DRCNNs is the residual block, which consists of two convolutional layers with skip connections. These skip connections not only directly transfer the input image information to the hidden layer but also shorten the gradient path, mitigating the vanishing gradient problem.

2.9. Summary of Related Works

The summary of related works is represented in the following table.

Table 2-1: Summary of related works

Authors	***Methods***	***Objective and contribution***	***Limitations/Gaps***
(Gong et al., 2020)	GAN with residual learning	Improve image denoising quality, especially for high-frequency noise.	color saturation loss that caused by their model is a bit high.
(Lan et al., 2021)	DRCNN	Reducing the problem of the gradient vanishing and producing good visual effect	Focuses only on gaussian noise.
(Ma et al., 2022)	FADNet	Enhance image denoising performance on diverse real-world noise types.	The evaluation focus is primarily on noise magnitude, not diverse noise types.
(Alsaiari et al., 2019)	GAN with perceptual loss functions	Enhance denoised image quality and preserve natural textures.	Considering specific noise type and generalizing.
(Z. Wang et al., 2020)	GAN with deep residual network (DRN)	Remove Gaussian noise while maintaining image sharpness.	Primarily focused on Gaussian noise.
(Tran et al., 2021)	GAN with DnCNN	To denoise real-world images or remove noise from real-world images while preserving details.	Over smoothing some image features, and combining a GAN with a DnCNN adds computational overhead.

Some methods struggle to accurately model and remove diverse noise types found in real-world scenarios, focusing only on specific patterns like Gaussian noise. Others sacrifice important image details for complete noise removal, leading to blurry or over-smoothed results. Additionally, certain approaches suffer from color saturation issues in the denoised image. Evaluations might emphasize noise magnitude rather than assessing performance across varied noise types, leading to incomplete understanding of a method's true capabilities. Balancing between the noise removal and detail preservation trade-off remains a challenge, as does generalizing methods to effectively handle diverse real-world images and noise characteristics.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Overview of the chapter

To achieve the research objectives outlined earlier, this chapter delves into the specific techniques, methodologies, and procedures employed. This includes an examination of the datasets, data augmentation and preprocessing steps, an overview of baseline procedures, and the development tools used. Additionally, the chapter covers the evaluation metrics chosen to assess the model's performance.

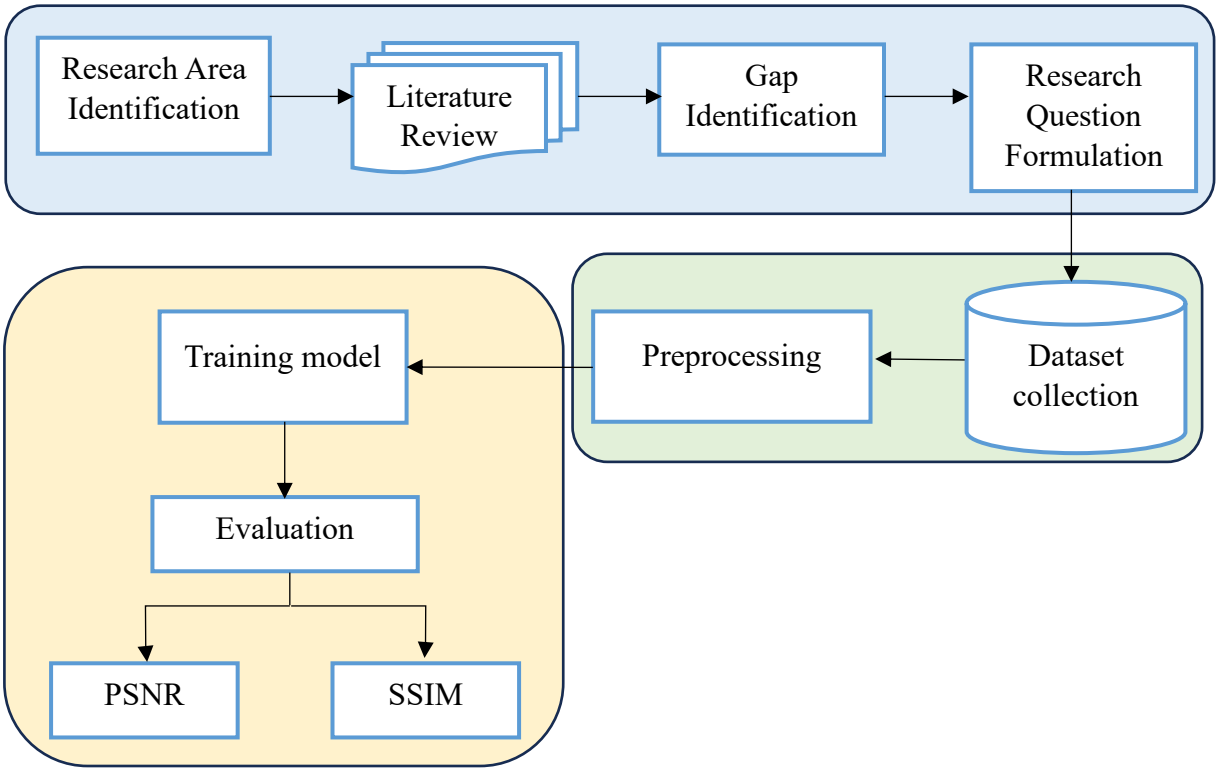


Figure 3-1: The overall process

3.2. Datasets

To conduct attention-based generative adversarial network for real-world image noise reduction we used real-world image dataset, SIDD dataset (Abdelhamed et al., 2018). This dataset is widely recognized for its role in advancing image denoising techniques, particularly in real-world scenarios. The SIDD is common dataset that is used for image denoising, it consists of 30,000 real world noisy images captured using five sample smartphone cameras, with their corresponding ground truth images (Abdelhamed et al., 2018). Realistic noise patterns observed in smartphone photography, including sensor

noise, photon shot noise. We resized the input images to 128x128 for computational efficiency and consistent input representation.

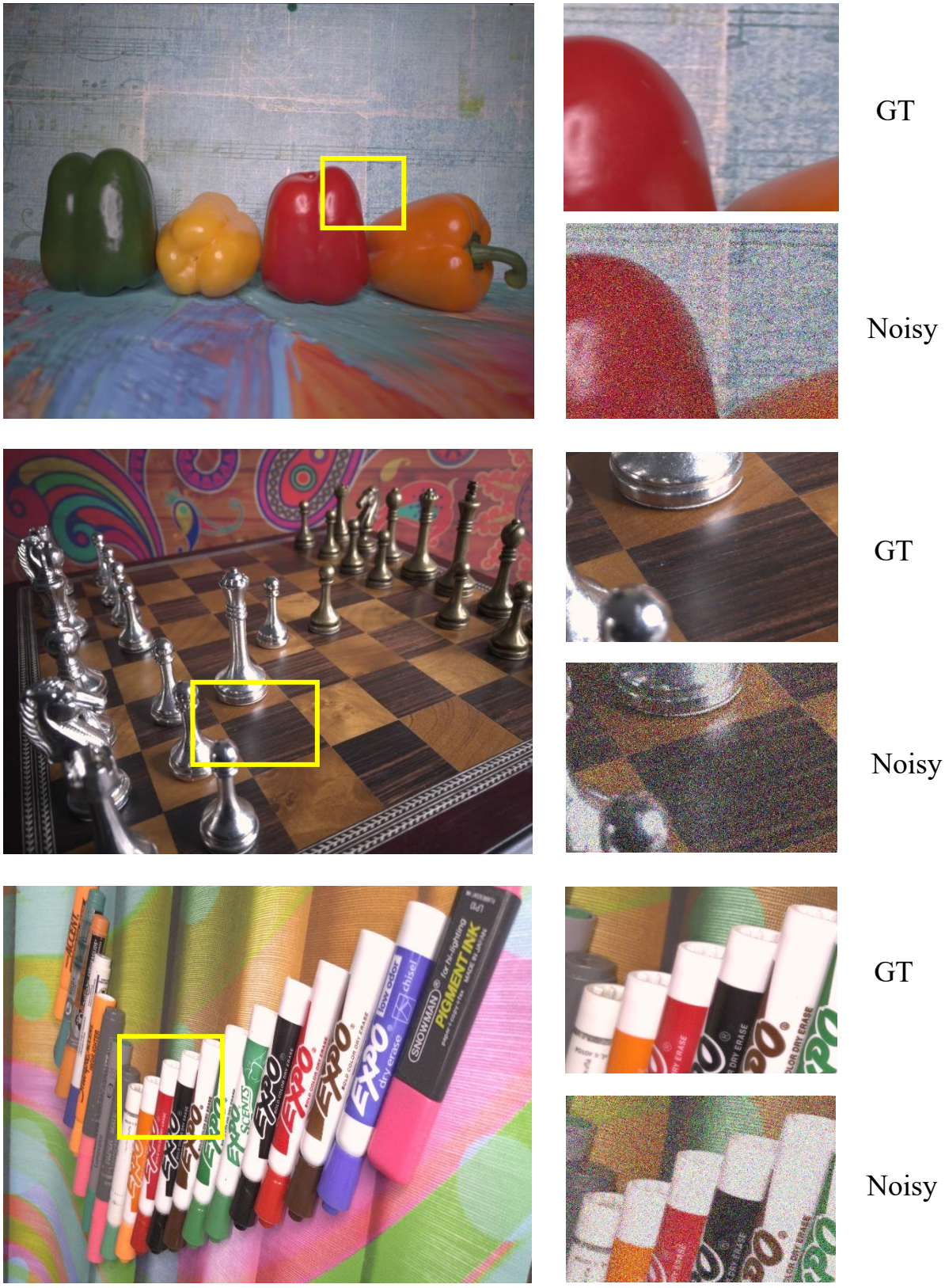


Figure 3-2: SIDD dataset samples

3.3. Pre-processing Techniques

3.3.1. Resize

Image resizing is a crucial pre-processing step. It involves modifying the dimensions of the input images to align with the model's expected input size. Given that deep learning models need a large number of instances, processing larger images involves additional computations. Resizing to a smaller size can minimize the computational resources required and greatly speed up training. We resized the images in to 128x128 to minimize the complexity of computation.

3.3.2. Normalization

We normalized our dataset to the range of -1 to 1 to ensure consistent input scaling, which helps in stabilizing and speeding up the training process. This was achieved using the formula:

$$x = \frac{y-\mu}{\delta} \qquad \textit{eq. 3.1}$$

Where, y represents the input image within range of 0 to 1, while, δ denote the variance and μ denote the mean. Both the mean and the variance is set to 0.5.

3.3.3. Data Augmentation

In the field of image denoising, data augmentation is an important method, particularly when utilizing deep learning models. It involves generating variations of the original images to enhance the diversity of the training data, which can help prevent overfitting. When implemented thoughtfully, augmentation methods can significantly improve the ability of model to generalize across different noise patterns and real-world conditions. As image denoising technology advances, data augmentation will continue to be essential for developing more robust and effective denoising algorithms. In our study, we applied the following techniques:

- Rotation: Images were randomly rotated at angles between -30° and 30°.
- Flipping: To ensure the models ability on handling images from various perspective, Horizontal and vertical flips were applied to the images to increase the dataset's diversity.

3.4. Development tools

To design and implement our study we used various development tools, hardware and software.

3.4.1. Hardware Tools

The hardware tools in the following table were used to implement our research.

Table 3-1: Hardware Tools

No.	*Tools*	*Purpose*
1.	Hard Disk	For datasets and models storage
2.	RAM	To improve the performance hardware
3.	GPU	Increasing the computation and accelerating the training process

3.4.2. Software Tools

software and coding tools were used for the implementation of the study, helping with the research process, from preprocessing of the dataset to model development and visualization.

Anaconda[1]: Anaconda is a widely-used open-source distribution for scientific computation, data analysis, and machine learning. It comes with numerous pre-installed packages and libraries, including popular ones like NumPy, pandas, SciPy, Matplotlib, and scikit-learn.

Jupyter Notebook[2]: is an open-source web application that enables users to create and share interactive documents containing live code, equations, visualizations, and descriptive text. It provides a flexible and collaborative environment for data analysis, scientific computing, and machine learning tasks.

TensorFlow[3]: TensorFlow is a comprehensive, open-source platform for machine learning, offering a vast array of tools, libraries, and community resources for building and deploying machine learning models.

PyTorch[4]: is a powerful and flexible framework for deep learning research and development, known for its ease of use and strong community support.

[1] https://anaconda.org/

[2] https://jupyter.org/

[3] https://tensorflow.org

[4] https://pytorch.org

3.5. Baseline Works

To evaluate the effectiveness of our model, we compared it with models that employ generative adversarial networks (GANs) for real-world image noise reduction. The D-GAN model proposed by (Tran et al., 2021) was selected as the baseline for this study. This model is particularly suitable for comparison as it uses a GAN framework to model and reduce noise in real images, specifically leveraging the SIDD dataset, similar to our research context.

The baseline model by (Tran et al., 2021) incorporates a single generator that focuses on noise modeling and removal, with a discriminator that differentiates between real and generated denoised images. Generator network uses a deep convolutional network, emphasizing the importance of capturing noise characteristics from real-world images. The authors highlight that their GAN-based approach handles the complex noise patterns found in real images. This baseline work serves as a foundation for our model, which extends this approach by integrating attention mechanisms within the GAN framework, aiming to further enhance the preservation of image details while effectively reducing noise.

3.6. Feature Extraction Network

Deep learning advancements has made researchers have interest to deepen the neural networks been a trend among researchers to increase the depth of neural networks to enhance model performance and capture more detailed features for image denoising. However, it has been noted that excessively deep networks can result in the loss of crucial information during feature extraction, negatively impacting the reconstruction of image features. to address this issue, (He et al., 2016) introduced the residual structure, which incorporates a skip connection that links the input and output of each block. This connection helps preserve essential information and improves feature reconstruction. By passing information to deeper layers of the neural network through these skip connections, even deeper networks can retain important image features. We utilized the residual learning network to effectively counter the issues of vanishing and exploding gradients, thereby stabilizing the performance of the networks. An illustration of residual block is shown in Figure 3-3.

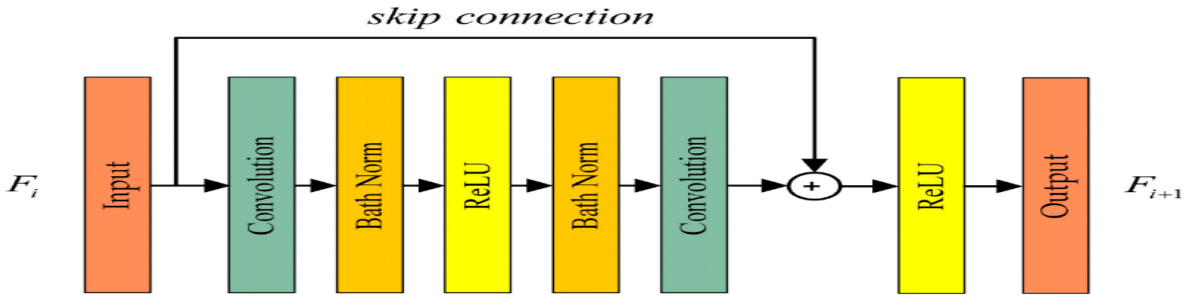


Figure 3-3: Residual Network Architecture

Source: adapted from (He et al., 2016)

3.7. Evaluation Metrics

For the assessment of the denoising results we are going to use evaluation metrics such as PSNR and SSIM values.

3.7.1. PSNR (Peak Signal to Noise Ratio)

Peak-Signal-to-Noise-Ratio (PSNR) is a widely utilized metric to assess the extent of compression or reconstruction of images in relation to their original quality. It calculates the ratio of an image's maximal power to the power of noisy interference that lowers the image's quality of representation. Better image quality is indicated by higher PSNR values (Isa et al., 2015). After noise reduction we used PSNR to evaluate the quality of image, with a higher PSNR indicating a better reduction. PSNR is calculated based on MSE and MSE can be calculated as follows:

$$MSE = \frac{1}{mn}\sum_{m=0}^{m-1}\ \sum_{n=0}^{n-1}\ (f(x) - g(x))^2 \qquad \textit{eq. 3.2}$$

where m*n is the number of data points, f(x) is the x^{th} measurement, and g(x) is its corresponding prediction. PSNR is measured in decibels (dB) and is calculated as using the following formula:

$$PSNR = 20\log_{10}\left(\frac{K}{\sqrt{MSE}}\right) \qquad \textit{eq. 3.3}$$

where K represents the highest signal strength.

3.7.2. SSIM (Structural Similarity Index Measure)

The Structural Similarity Index (SSIM) is a metric used to assess how similar two images are perceived to be. Unlike metrics like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR), which focus on pixel-wise differences, SSIM considers, luminance, contrast,

and texture similarity, and structural information. This approach provides a more accurate evaluation of image quality (Bovik et al., 2005).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\,\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \qquad \textit{eq. 3.4}$$

Where x and y are the two images being compared, μ_x, and μ_y are means of x and y respectively, σ_x^2 and σ_y^2 are the variance of x and y, σ_{xy} is the covariance of x and y, and c_1 and c_2 are constants to stabilize the division with weak denominator.

CHAPTER FOUR

4. PROPOSED REAL-WORLD IMAGE NOISE REDUCTION APPROACH

4.1. Overview of Chapter

This chapter outlines the proposed method's architecture for real-world image noise reduction, which incorporates attention mechanisms within a generative adversarial network (ANR-GAN) framework. The chapter is divided into four sections: the first part provides a comprehensive overview of the model's architecture, emphasizing the integration of attention layers to enhance feature extraction and noise suppression. The second section delves into the design of the generator, which employs a deep convolutional network with residual and attention blocks, and the discriminator, which distinguishes between real and denoised images. The third segment explains the learning objectives, including loss functions and optimization strategies tailored for effective noise reduction. The chapter's last section presents the training pseudocode, detailing the steps required to train the ANR-GAN model effectively. This structure ensures a comprehensive understanding of the proposed architecture, from conceptual design to practical implementation.

4.2. Model Architecture

The proposed approach named "ANR-GAN" basically relies on the principle of Generative Adversarial Network (GAN) from (I. Goodfellow et al., 2020). Generator (G) and Discriminator (D) networks training in adversarial manner makeup the GAN network. In our proposed model we used U-Net for the generator network (Nemni et al., 2020) which consist of encoder-decoder structure with skip connection.

The proposed model incorporates a self-attention module(H. Zhang et al., 2018) into the decoder portion of the generator to effectively capture long-range contextual information. Additionally, in the encoder part of the generator, we implemented a channel attention module (Hu et al., 2018) to prioritize the most significant features essential for image denoising. A channel attention layer was placed before the skip connection to selectively emphasize important information before it is transferred to the decoder. The generator then synthesizes an artificial image by leveraging information from the real image, aiming to deceive the discriminator and achieve effective noise reduction.

In the discriminator network, we enhanced its performance by adopting a method that goes beyond simple binary classification. Instead of simply classifying an image as real or fake, the discriminator assesses the realism of an image in comparison to another known fake image. The discriminator's primary role is to differentiate between authentic images and images generated by the generator or fake images. It continuously adjusts its evaluation criteria to better differentiate between these images, while the generator is simultaneously learning to produce images that are increasingly difficult to distinguish from real ones.

The proposed model incorporates channel attention within the encoder blocks and self-attention within the decoder blocks of the U-Net-based generator. These attention mechanisms enhance the model's ability to reduce noise properly while retaining or preserving important details of image. The channel attention in the encoder helps to prioritize significant features during the down-sampling process, while the self-attention in the decoder facilitates capturing long-range dependencies, thereby improving the realism of the denoised images and effectively challenging the discriminator.

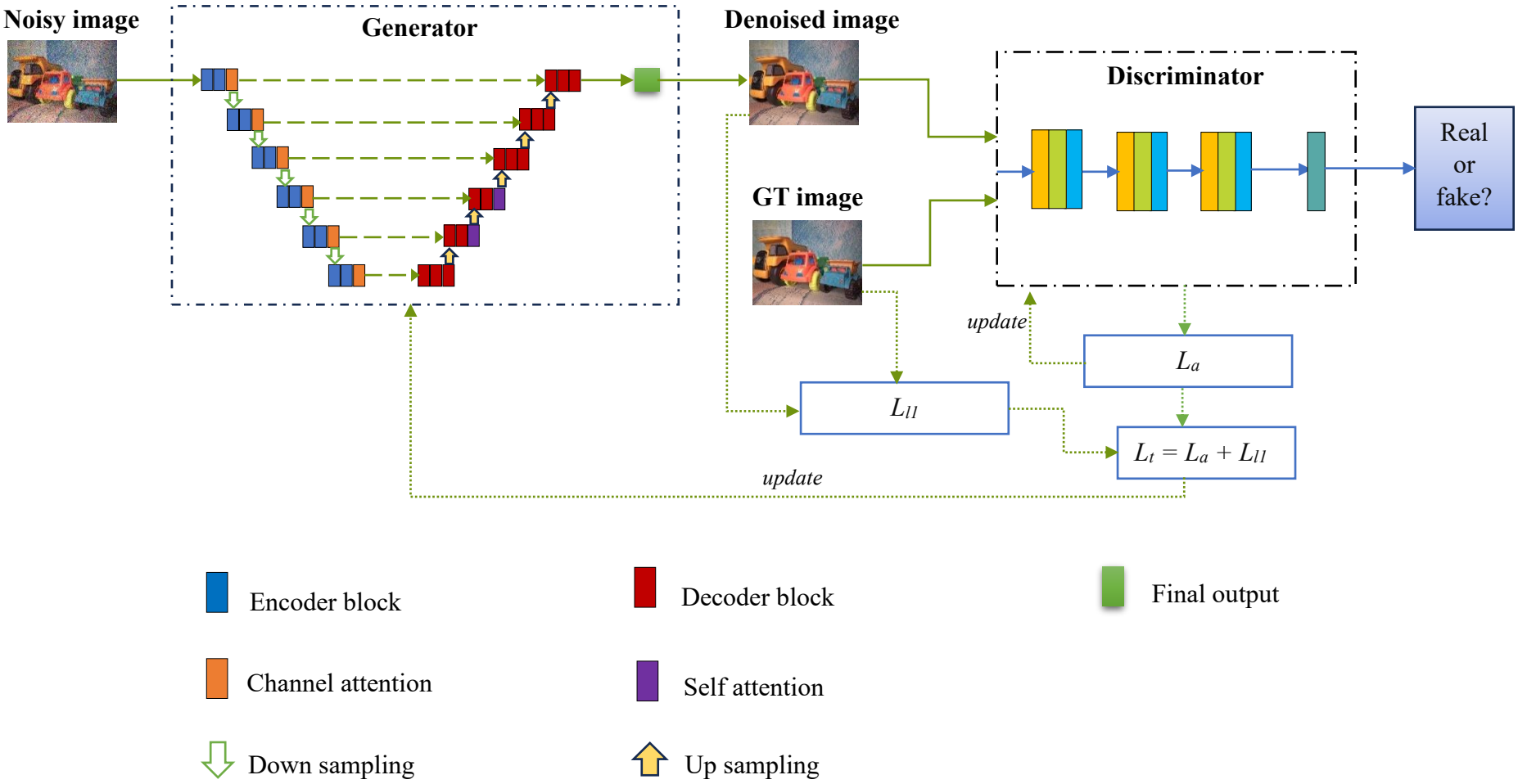


Figure 4-1: Overall architecture of the proposed model

In the GAN framework, self-attention plays a good role in enabling the generator network and discriminator network to effectively capture the image features long-range dependencies (H. Zhang et al., 2018). This capability is particularly important in tasks like image denoising, where maintaining relationships between key features across the image is essential for preserving fine details such as texture, sharpness, and structure. Self-attention helps to reduce common GAN artifacts like distortions and blurriness by refining and enhancing generated features based on their global context. This results in cleaner and sharper denoised images. To fully capture these advantages, we incorporated self-attention in the decoder part of the generator in our proposed model.

Channel attention is a technique used in convolutional neural networks to focus on most relevant channels within a feature map, thereby enhancing network performance by filtering out noise and irrelevant information (Hu et al., 2018). In the area of noise reduction, channel attention enables the model to prioritize critical features, such as the texture and structure of key image elements, resulting in better preservation of these features in the final denoised output. We incorporated channel attention in the encoder part of the generator to ensure that crucial features are emphasized early in the network, laying a strong foundation for subsequent layers. By refining these essential features in the encoder, the model is better equipped to produce accurate and detailed denoised images.

4.3. Generator Network

In a Generative Adversarial Network (GAN), the generator's responsibility is for creating artificial data that has resemblance with real data. Often designed as a neural network with convolutional layers, the generator receives random noise or inputs and produces sample data meant to be indistinguishable from actual data. For our generator, we employed the U-Net architecture (Nemni et al., 2020), that features encoder-decoder architecture with skip connections and integrates hybrid attention mechanisms, including both self-attention and channel attention.

U-Net, a deep learning technique which was proposed by (Ronneberger et al., 2015), is characterized by two primary pathways: contraction and expansion. The expansion pathway consists of decoder layers that decode the encoded information and combine it with details from the contraction pathway through skip connections to generate a segmentation map. The contraction pathway follows a traditional convolutional network structure, employing

encoder blocks that capture contextual information while decreasing the dimensions of the inputs (Ronneberger et al., 2015).

In our generator's architecture, the encoder incorporates layers which include Convolution layer, Batch Normalization layer, and ReLU layer, and the decoder utilizes the same, except for the transposed Convolution. The bottleneck layer in the generator compresses the input and lowers the number of dimensions of the feature maps in order to learn more condensed and abstract representations of the image. Additionally, generator utilizes skip connections to preserve high-resolution features from earlier layers, which is crucial for retaining detailed image information throughout the denoising process.

Self-attention was integrated in two layers of decoder, the second and third which helps the generator to access the global and local dependencies in feature maps. This can help to improve the spatial details and structures of the denoised images. Also, channel attention is integrated in all encoder layers to allow the relevant features to be focused on by the network.

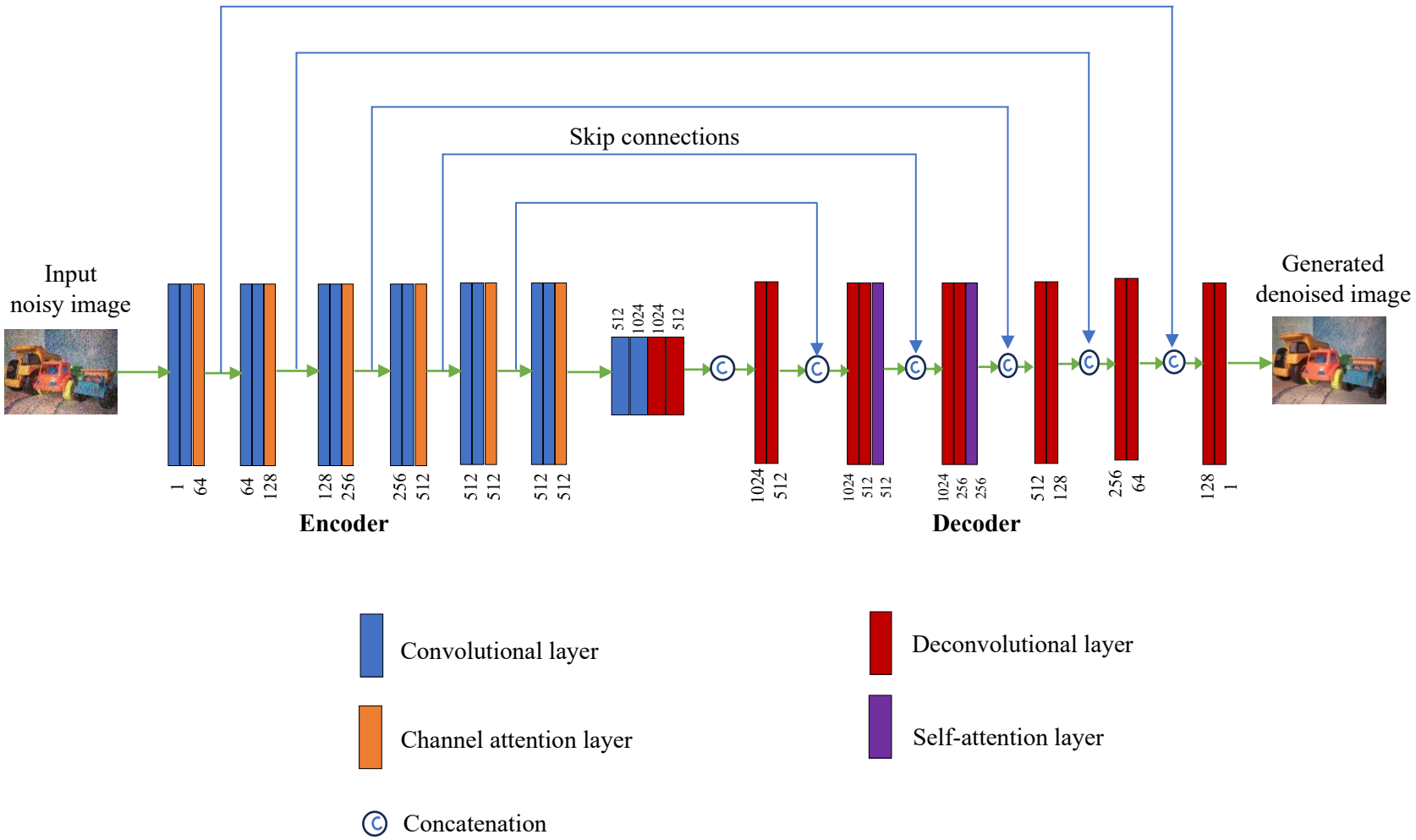


Figure 4-2: Architecture of Attention U-Net Generator

In our proposed architecture, the encoder block serves as a down-sampling mechanism, consisting of six convolutional layers (conv1 – conv6), with a channel attention block following each layer. Each convolutional layer operates with kernel size of 4 and stride of 2, and also Batch Normalization and a ReLU activation function.

The decoder block functions as an up-sampling mechanism, consisting of six deconvolutional layers (deconv1 – deconv6). Self-attention blocks are incorporated after the second and third layers to improve the ability of the model to identify long-range dependencies. Similar to the encoder, each deconvolutional layer has stride of 2 and kernel size of 4, with Batch Normalization and the ReLU activation is applied on each layer, except for the final deconvolutional layer (deconv6), where the activation function employed is Tanh.

4.3.1. Channel Attention

Channel attention is a valuable method that enhance the performance of convolutional neural networks, which works by enabling the model to prioritize the most informative features within each channel of a feature map. In the image denoising field, this mechanism is particularly useful because it helps the network distinguish between important image features, such as textures, edges, and fine details, and less important elements, like noise.

The underlying principle of channel attention is to assigning different importance levels for every channel based on their contribution to the task at hand. For instance, channels that capture significant features like sharp edges or textures receive higher attention, while those representing noise or irrelevant background information are suppressed. This selective focus allows the network to more effectively denoise the image, as it can concentrate on preserving critical details while minimizing the influence of noise.

Channel attention was popularized by the introduction of the Squeeze-and-Excitation (SE) block, which was first presented by (Hu et al., 2018). The SE block operates by first globally pooling each channel to capture its overall importance, subsequently, a fully connected network is applied to compute attention weights, which are used to rescale feature map channels. This process enhances the network's ability to learn which channels contain the most relevant information for the task, leading to improved image quality and more accurate feature preservation (Hu et al., 2018).

In our research, we integrate channel attention within the encoder blocks of our U-Net-based generator. By applying channel attention early in the network, we ensure that the most

important features are highlighted before they are passed through subsequent layers. This early emphasis on crucial features helps to create a strong foundation for the network, leading to more effective denoising and better preservation of essential image details throughout the processing pipeline.

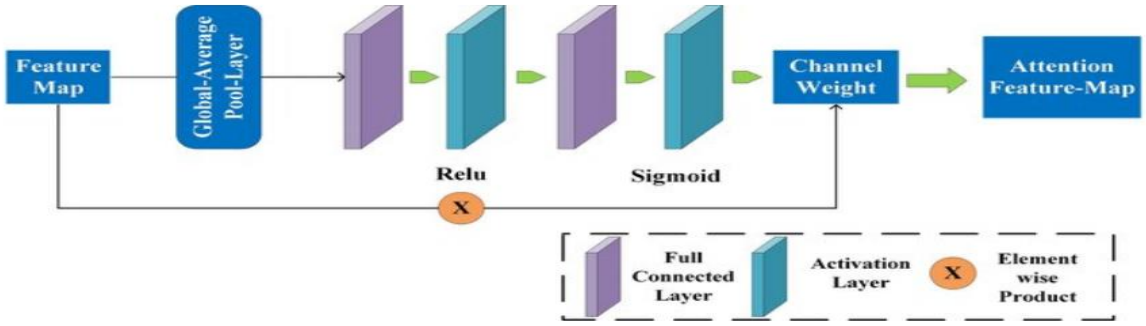


Figure 4-3:Structure of channel attention mechanism

Source: adapted from (Chen et al., 2023)

4.3.2. Self-Attention

Self-attention is another advanced mechanism that has been incorporated into our model, particularly within the decoder blocks. While channel attention focuses on enhancing the importance of specific feature channels, self-attention addresses the relationships between spatially distant pixels within an image. This mechanism is vital for capturing global dependencies, which are often crucial for maintaining the overall coherence and structure of an image.

The concept of self-attention, originally introduced by (Vaswani et al., 2017) in the area of natural language processing, has since been adapted to various tasks in computer vision, including image generation and denoising. Self-attention works by allowing each pixel (or feature map position) to attend to all other pixels, effectively considering the global context when refining features. This approach enables the network to maintain consistency across different regions of the image, ensuring that details like texture and structure are preserved even in areas that are spatially distant from one another (H. Zhang et al., 2018).

In our proposed model, self-attention is strategically placed within the decoder layers of the generator. By enhancing the decoder with self-attention in middle layers, we enhance the network's ability to refine and sharpen features during the up-sampling process. This refinement is particularly important for denoising, as it helps to reduce common GAN-

related artifacts, such as blurriness and distortions, by considering the global context of the image. The result is a cleaner, sharper output that closely resembles the original image.

The combination of channel attention in the encoder and self-attention in the decoder creates a robust mechanism for image denoising, where critical features are emphasized from the start, and global dependencies are maintained throughout the processing pipeline. These attention mechanisms significantly contribute to the effectiveness of our model, enabling it to produce denoised images that are high in quality by retaining fine details and overall structural integrity.

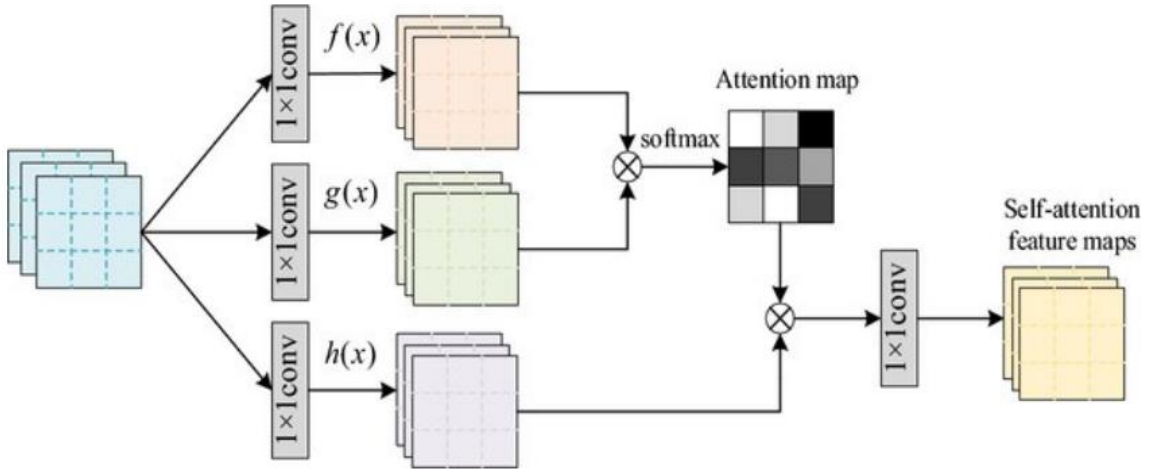


Figure 4-4: Self-Attention (SA) structure diagram

Source: adapted from (Gao et al., 2023)

4.4. Discriminator Network

Discriminator network is a crucial component to differentiate between real and generated (fake) images. As part of the GAN framework, the discriminator has a key role in directing the generator to produce realistic images by constantly challenging it with its capacity to distinguish real data from synthetic data (Isola et al., 2016).

The discriminator in our model is constructed with convolutional layers, then batch normalization and Leaky ReLU activation functions. The layers gradually extract features from input images, increasing the depth of the feature maps while reducing their spatial dimensions. The extracted features are then analyzed to determine the authenticity of the image, whether it is a fake image generated by the generator or a real image.

The discriminator is designed to focus not only on global image features but also on local details that are critical for accurate noise reduction and image denoising. This focus is achieved through the use of multiple layers of convolution, which allows the network to capture both fine-grained details and high-level abstractions that differentiate a real image from a generated one.

Moreover, the discriminator is optimized using an adversarial loss function. This function measures how effectively the discriminator correctly classifies generated images versus real images. In response, the generator is fine-tuned to reduce this loss, forcing it to generate increasingly similar images to actual ones.

By incorporating these techniques, the discriminator in our model serves as a robust adversarial opponent for the generator, continuously improving the denoised image's quality that are produced. Interplay between the networks ensures that images generated retain fidelity of original data in addition to being noise-free, with sharp details and accurate texture representation.

This discriminator setup, combined with the attention mechanisms employed in the generator, creates a powerful framework for real image noise reduction, pushing the boundaries of previous image denoising methods.

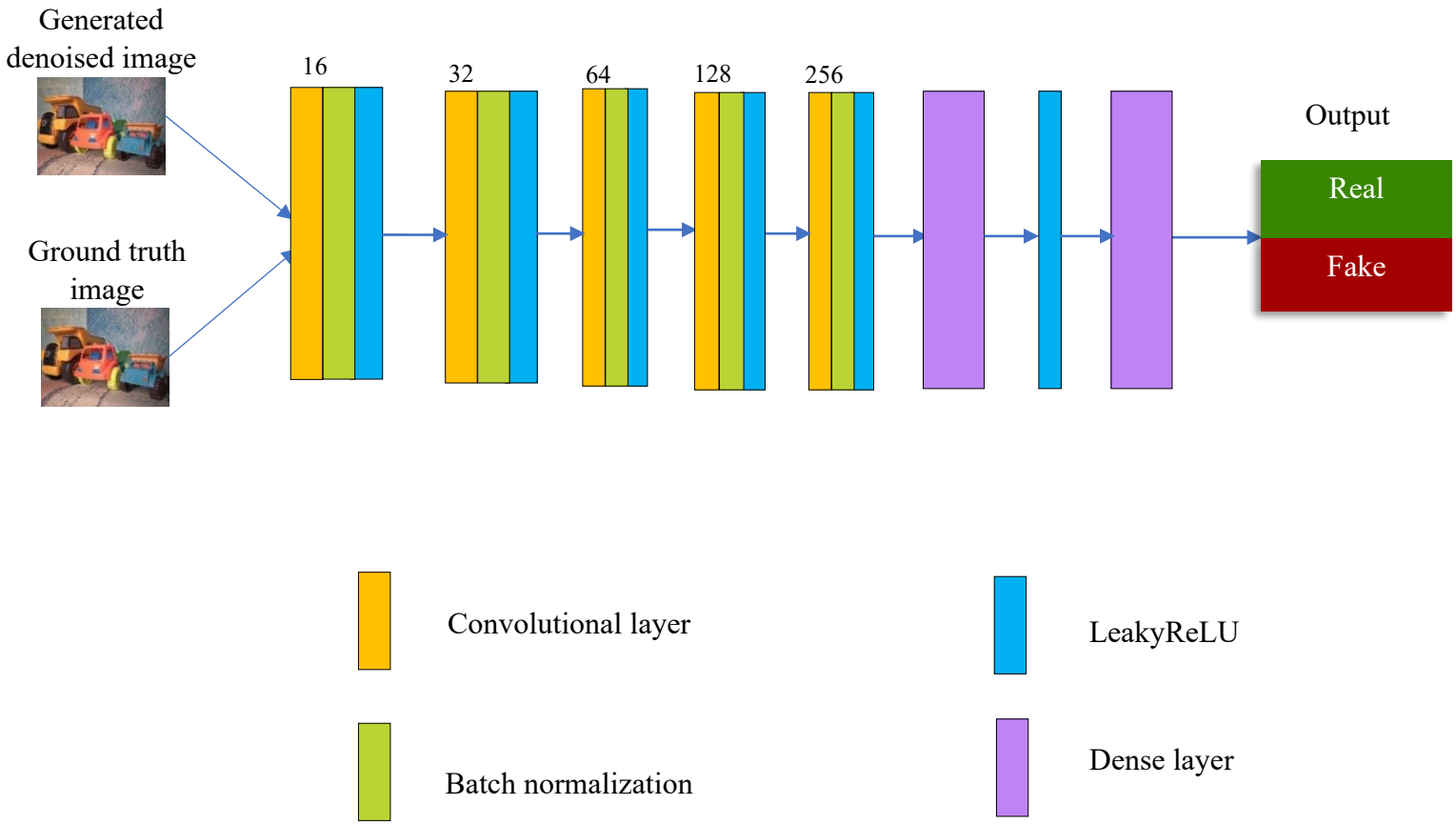


Figure 4-5: Architecture of Discriminator Network

4.5. Loss functions

In our study, we employ a comprehensive loss function strategy to effectively guide the training of the networks of the GAN. Primary components of this strategy include the pixelwise loss, adversarial loss, and the total loss. These losses play an important role ensuring that the images generated does not only retain the essential features, but also are visually realistic resembling the original images, especially when denoising is involved.

4.5.1. Adversarial Loss

The adversarial loss is integral to the GAN itself, which competes the two networks, which are generator and discriminator against each other in a minimax manner (Dong & Yang, 2019). In our model, the aim of the generator is producing denoised images that has high resemblance of the ground truth (GT) images, while the discriminator's work is differentiating between these real, noiseless images and generated images.

For the discriminator, the adversarial loss is formulated as:

$$LadvD = -Ex[logD(x)] - Ez[log(1 - D(G(z)))] \qquad eq.\ 4.1$$

Where *D(x)* represents the discriminator's probability estimate that a real image *x* is genuine. *G(z)* denotes the generator's output when provided with noise *z*. *D(G(z))* is the discriminator's assessment of the probability that a fake instance is rea. E_x is the expected value across all real data instances. *Ez* represents the expected value across all random inputs to the generator (in effect, The estimated amount of all created artificial instances *G(z))*.

For the generator, the adversarial loss aims to fool the discriminator into classifying the generated (denoised) images as real:

$$LadvG = -Ez[logD(G(z))] \qquad eq.\ 4.2$$

This loss motivates the generator to generate images that are highly realistic, minimizing the contrast between real images and the generated as perceived by the discriminator.

4.5.2. Pixelwise Loss

While adversarial loss encourages realism, it does not guarantee that the generated images will accurately preserve the structural characteristics of the original images. In order to address this, we incorporate pixelwise loss, which directly compares the ground truth image to the generated image on a per-pixel basis (S. Zhao et al., 2019). This loss is particularly

effective when used purposes image denoising, where the fine structures and details of the image must be preserved (Johnson et al., 2016). L1 loss is a widely employed loss function that reduce the absolute differences of predicted and target values. It is known for its robustness to outliers, making it less susceptible to the influence of extreme data points. In the context of image generation, L1 loss is often employed to compare generated images with real ones, as demonstrated in the work of (Isola et al., 2016).

$$\mathcal{L}l1 = |It - \widetilde{I}t| \qquad \textit{eq. 4.3}$$

Where It is the gt image and $\widetilde{I}t$ is generated image.

4.5.3. Total Loss

A weighted combination of the adversarial loss and the pixelwise loss is the total loss function used in our model is. This combined loss function is focuses to balance the trade-off between generating realistic images (through adversarial loss) and ensuring that these images do not loss the important features of the original images (through pixelwise loss).

The total loss is defined as:

$$L_{total} = \lambda_{adv} L_{adv} + \lambda_{pixel}\, L_{pixel} \qquad \textit{eq. 4.4}$$

Where: λ_{adv} is the weight assigned to the adversarial loss. λ_{pixel} is the weight assigned to the pixelwise loss.

By adjusting these weights, our model can be fine-tuned to prioritize either their similarity to the clean image or the authenticity of the generated images, based on the particular demands of the denoising assignment.

4.6 Training Pseudocode

The training algorithm we used to utilized in our work is stated in this section.

Algorithm 1: The training algorithm of the Proposed ANR-GAN model

1. **Initialize** *generator G and discriminator D weights.*
   Incorporate ***channel attention*** *in the encoder and* ***self-attention*** *in the decoder block of U-Net architecture.*
2. **Input:** *A batch of noisy images I_n and corresponding ground truth (GT) images I_{gt}.*
3. ***For*** *number of epochs E* ***do***
4. ***For*** *number of batches in epoch* ***do***

5. ***Update*** *discriminator D using:*

$$Ep_{data}\,[log(D(x))]\;+\;Ep_z\,[log\,(1\;-\;D(G(z)))]$$

Backpropagate *to adjust discriminator weights.*

6. ***Update*** *generator G using:*

$$Ep_{data}\left[log\big(D(x)\big)\right]+\;Ep_z\;\left[log\left(1\;-\;D\big(G(z)\big)\right)\right]$$

$$\frac{1}{N}\sum_i (di\;-\;d'i\,)^2$$

Backpropagate *to adjust generator weights.*

7. ***Save*** *the generator and discriminator weights periodically.*
8. **End for**
9. **Output:** *Denoised image I_d generated by G that closely matches the ground truth I_g*

The **Bold,** indicates the updated components

CHAPTER FIVE

5. IMPLEMENTATION DETAILS

5.1. Chapter Overview

This chapter discusses about how our proposed approach is implemented, which consists of the establishment of the working environment, data preprocessing, implementation of ANR-GAN, and the experimental class employed.

5.2. Working Environment

The hardware tools used in the working environment are described in this section, which were essential for the implementation of the proposed model. The selection and configuration of these hardware tools directly influenced the model's performance, processing speed, and overall efficiency. By detailing the specific hardware components, we can provide a clear understanding of the resources required to achieve the reported results and ensure the reproducibility of the study.

Laptop:

- Processor: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40GHz
- Operating system: Windows 10 Pro
- RAM: 6.00 GB
- System type: 64-bit operating system, x64-based processor

Desktop:

- Processor: Intel(R) Core (TM) i7-8700M CPU @ 3.20GHz 3.19 GHz
- Operating system: Windows 10
- Installed RAM: 8.00 GB
- System type: 64-bit operating system, x64-based processor
- Graphics: Graphics: Intel ® HD Graphics 520/ NVIDIA Corporation
- GPU: GeForce RTX 2070 Super 8 GB RAM

5.3. Environmental Setup

In this research, we utilized a variety of software tools and IDEs that support machine learning frameworks such as PyTorch. These tools facilitated the efficient development and analysis of our models.

Anaconda: This environment was important in the implementation of our proposed GAN model for real image noise reduction. Anaconda is an open-source distribution which is widely-used, that simplifies the management and deployment of scientific computing libraries and machine learning frameworks. It comes pre-packaged with essential libraries such as NumPy, pandas, SciPy, Matplotlib, and scikit-learn, which are integral for data analysis and model development. For our research, we utilized Anaconda version 23.7.4 to install and manage all necessary libraries and dependencies.

Jupyter Notebook: Jupyter Notebook was employed for developing and documenting our workflows. It allows for the seamless combination of code, text explanations, and visualizations in a single document, making it easier to share and reproduce our research findings.

Kaggle: a cloud-based platform that offers an accessible environment for executing Python code without requiring local setup. Kaggle provides free access to powerful computational resources, including GPUs, which is particularly beneficial for accelerating the training as well as the testing of deep learning models. This platform was instrumental in efficiently managing the computational demands of our GAN model for real image noise reduction.

These tools collectively contributed to the effective execution of our research by providing robust environments for model development, experimentation, and analysis.

5.4. Proposed Model Implementation

The proposed ANR-GAN model for real image noise reduction incorporates a U-Net based generator as well as a discriminator. The generator is constructed with two primary blocks: the encoder block and the decoder block, each integrated with hybrid attention mechanisms to enhance feature extraction and image reconstruction. The encoder block consists of six layers of neural networks, utilizing channel attention mechanisms to focus on the most salient features, while the decoder block, also comprising six layers, employs self-attention mechanisms for the purpose of capturing long-range dependencies within the images. Both blocks in the generator utilize the ReLU activation function to ensure efficient training and effective feature learning.

```
class Generator(nn.Module):
    def __init__(self):
        super(G, self).__init__()
        # Encoder
        self.encoder1 = nn.Sequential(
            nn.Conv2d(3, 64, 4, 2, 1),
            nn.BatchNorm2d(64),
            nn.ReLU(True),
            ChannelAttention(64))
        self.encoder2 = nn.Sequential(
            nn.Conv2d(64, 128, 4, 2, 1),
            nn.BatchNorm2d(128),
            nn.ReLU(True),
            ChannelAttention(128))
        self.encoder3 = nn.Sequential(
            nn.Conv2d(128, 256, 4, 2, 1),
            nn.BatchNorm2d(256),
            nn.ReLU(True),
            ChannelAttention(256))
```

Sample code 5-1: Generator implementation (encoder)

The sample code above shows how the encoder block of the generator is structured and it is discussed in the table below.

Table 5-1: Encoder block components in generator network

Encoder architecture in Generator	
Layers in the encoder	**Content of each block**
1	Conv –- (filters = 64, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
2	Conv –- (filters = 128, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
3	Conv –- (filters = 256, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
4	Conv –- (filters = 512, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
5	Conv –- (filters = 512, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
6	Conv –- (filters = 512, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()

Table 5.1 outlines the components utilized in the encoder block of the generator network. The encoder consists of six convolutional layers, each with a 4x4 kernel size, a stride of 2, and a padding of 1. A batch normalization is applied following each convolutional layer for the training process to stabilize, ensuring that the generated outputs maintain a consistent distribution. Additionally, each layer use of the ReLU activation function to add non-linearity, which is crucial for mitigating the vanishing gradients problem and enhancing the ability model to learn complex patterns.

```
        # Decoder
        self.decoder1 = nn.Sequential(
            nn.ConvTranspose2d(1024, 512, 4, 2, 1),
            nn.BatchNorm2d(512),
            nn.ReLU(True))
        self.decoder2 = nn.Sequential(
            nn.ConvTranspose2d(1024, 512, 4, 2, 1),
            nn.BatchNorm2d(512),
            nn.ReLU(True)
            SelfAttention(512))
        self.decoder3 = nn.Sequential(
            nn.ConvTranspose2d(1024, 256, 4, 2, 1),
            nn.BatchNorm2d(256),
            nn.ReLU(True)
            SelfAttention(256))
```

Sample code 5-2: Generator implementation (decoder)

Table 5-2: Decoder block components in generator network

Decoder architecture in Generator	
Layers in the decoder	**Content of each block**
1	Deconv – (filters = 1024, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
2	Deconv – (filters = 1024, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
3	Deconv – (filters = 1024, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
4	Deconv–- (filters = 512, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
5	Deconv–- (filters = 256, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, ReLU ()
6	Deconv–- (filters = 128, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Tanh ()

Table 5.2 provides an overview of the components incorporated within the decoder block of the generator network. The decoder consists of six deconvolutional layers, each with a 4x4 kernel size, a stride of 2, and padding of 1. Batch normalization is applied to all layers except the final one, helping the training process to stabilize and ensure that the denoised images maintain consistent distributions. In each layer the ReLU activation function is used, with the exception of the last layer, to introduce non-linearity and prevent the vanishing gradients issue. Given that our images are normalized within the range of [-1,1], in the last layer the activation function which is Tanh is employed to produce output values within this range, ensuring that the generated images adhere to the expected normalization.

```
class Discriminator(nn.Module):
    def __init__(self):
        super(D, self).__init__()
        self.main = nn.Sequential(
            nn.Conv2d(3, 16, 4, 2, 1),
            nn.LeakyReLU(0.2, inplace = True),
            nn.Conv2d(16, 32, 4, 2, 1),
            nn.BatchNorm2d(32),
            nn.LeakyReLU(0.2, inplace = True),
            nn.Conv2d(32, 64, 4, 2, 1),
            nn.BatchNorm2d(64),
            nn.LeakyReLU(0.2, inplace = True),
            nn.Conv2d(64, 128, 4, 2, 1),
            nn.BatchNorm2d(128),
            nn.LeakyReLU(0.2, inplace = True),
```

Sample code 5-3: Discriminator implementation

Table 5-3: Content of discriminator architecture

Layers	**Content of each block**
1	Conv –- (filters = 16, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, LeakyReLU (0.2)
2	Conv –- (filters = 32, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, LeakyReLU (0.2)
3	Conv –- (filters = 64, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, LeakyReLU (0.2)
4	Conv –- (filters = 128, kernel size = (4,4), stride = (2,2),

	padding = (1,1), bias = True), Batch normalization, LeakyReLU (0.2)
5	Conv –- (filters = 256, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True), Batch normalization, LeakyReLU (0.2)
6	Conv –- (filters = 512, kernel size = (4,4), stride = (2,2), padding = (1,1), bias = True)

As outlined in Table 5.3, discriminator is structured consisting of six convolutional layers with a 4x4 kernel size of, stride of 2, and padding of 1. The Batch normalization and LeakyReLU activation are applied in the layers, with the exception of the final layer. a sigmoid activation function is used in the final layer, to generate a probability-like result, which indicates the likelihood that the input image is either real or fake.

5.4.1. Channel attention Implementation

Channel attention is incorporated into encoder block of the generator network to enhance concentration of the model on the most relevant features for effective noise reduction in images. The channel attention mechanism is divided into two components: the squeezing block and the excitation block. In the squeezing block, the input feature maps are subjected to global average pooling and global max pooling in order to produce channel descriptors that effectively remove noise from the image while capturing its key features. The excitation block consists of two fully connected layers: the first performs dimensionality reduction, reducing the number of channels from C to C/r (where r is the reduction ratio), and the second restores the channels back to C. To produce the channel attention weights a sigmoid activation function is applied. These attention weights are used to reweight the original feature map X, with each channel being scaled according to its importance, allowing the network to better preserve important image details while effectively reducing noise.

```
class ChannelAttention(nn.Module):
    def __init__(self, in_channels, reduction_ratio=16):
        super(ChannelAttention, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.max_pool = nn.AdaptiveMaxPool2d(1)
        self.fc = nn.Sequential(
            nn.Conv2d(in_channels, in_channels // reduction_ratio, 1, bias=False),
            nn.ReLU(inplace=True),
            nn.Conv2d(in_channels // reduction_ratio, in_channels, 1, bias=False),
            nn.Sigmoid()
        )
```

Sample code 5-4: Channel attention implementation

5.4.2. Self-attention Implementation

In our model, the self-attention mechanism enables the generator to focus not only on the fine details (local dependencies) but also on the relationships between different regions of the image (global dependencies). This capability allows the generator to produce images with sharper details and more refined structures, contributing to a higher level of realism. In our architecture, the self-attention layer is applied to the second and third layers of the decoder.

```
class SelfAttention(nn.Module):
    def __init__(self, in_dim):
        super(SelfAttention, self).__init__()
        self.query_conv = nn.Conv2d(in_channels=in_dim, out_channels=in_dim//8, kernel_size=1)
        self.key_conv = nn.Conv2d(in_channels=in_dim, out_channels=in_dim//8, kernel_size=1)
        self.value_conv = nn.Conv2d(in_channels=in_dim, out_channels=in_dim, kernel_size=1)
        self.gamma = nn.Parameter(torch.zeros(1))
```

Sample code 5-5: Self attention implementation

Before implementing self-attention, the input image is typically transformed into a vector sequences, with each vector representing a pixel or a group of pixels. These vectors are treated as queries, keys, and values. To generate these, the input image undergoes a convolution operation with a 1x1 kernel. The query vectors are then compared with the key vectors through dot products, determining the relevance (attention) of each key in relation to the query. These relevance scores are normalized using SoftMax to establish weights for attention. Based on the attention scores, the final output for every query is calculated as a weighted sum of the value vectors. After processing all vectors through the self-attention

mechanism, they are typically reshaped back into their original spatial format to preserve spatial information.

5.4.3. Discriminator Implementation

To enhance the performance of discriminator in our GAN-based denoising model, we employed a specialized discriminator design. The discriminator processes two input images, one real and one generated, independently via a sequence of convolutional layers in order to extract the corresponding features. After feature extraction, the features from the generated and real images are combined and passed through an average pooling layer. This layer decreases the dimensionality of the concatenated features by averaging the values. The combined features then undergo further processing through additional convolutional layers, incorporating a sigmoid activation function applied in the final layer to produce a probability-like output. This output indicates the resemblance of the input image belonging to either the real or generated categories. The discriminator also provides a response to the generator with the output it gained.

5.5. Hyperparameter configuration

Hyperparameters are components that can be adjusted to influence performance of model's training. Key hyperparameters include number of epochs, batch size, optimization strategy, and learning rate. For this research, we employed a learning rate of 0.0002 to update the network weights with Adam optimizer. Optimizers play an important role in tuning a neural network’s parameters, such as weights and learning rate, to minimize loss. Adam, as described by (Kingma & Ba, 2014), integrates advantages from two stochastic gradient descent methods. It is noted for its ease of implementation, computational efficiency, and memory efficiency. The learning rate, which dictates the extent of weight adjustments during training. The model was trained for 100 epochs with a batch size of 16.

Table 5-4: Hyperparameter configuration

Hyperparameter	Values
Batch size	16
Number of epochs	100
Optimizer	Adam
Learning Rate	0.0002

Table 5.4 illustrates the hyperparameter configuration, which is used during the training time of proposed model. To stabilize training and enhance performance of the model we use batch number 16 and learning rate 0.0002.

5.6. Experimental Classes

To determine the effectiveness of integrating attention mechanisms in GANs for improving denoised image quality, it is essential to conduct and test multiple experiments.

Four distinct experiments were employed, as outlined in Table 5.5. First experiment serves as the baseline work, which forms the foundation for subsequent comparisons. As detailed in Section 3.5, the DGAN model integrated attention mechanisms and residual blocks, along with a deep feature encoder within the generator. A conventional discriminator was employed for the adversarial training. Second experiment initiated the development of our proposed noise reduction model by introducing a channel-attention block in the U-Net-based generator. Third experiment involved the addition of a self-attention block within the decoder of the generator, refining the capability of the model to concentrate on crucial features.

In the fourth experiment, we enhanced the generator with self-attention in the decoder part and channel attention in the encoder block, and made the discriminator deeper, aiming to improve noise reduction performance in real images. We used adversarial loss and pixelwise loss for fine-tuning purpose.

Table 5-5: List of experimental classes

Notation	Experiment	Dataset
D-GAN	baseline	SIDD
CA+D-GAN (ANR-GAN-1)	The encoder block improved by integrating channel attention mechanism	SIDD
SA+D-GAN (ANR-GAN-2)	The decoder block improved by integrating self attention mechanism	SIDD
CA+SA+D-GAN (ANR-GAN-3)	Incorporating channel attention in encoder and self-attention in decoder of the generator network	SIDD

Table 5.5 outlines the experimental class conducted in this study. In the first experimental model we integrated the Channel Attention (CA) mechanism within the encoder block of the generator. The introduction of CA helps the network focus on the most informative channels in the noisy images, improving the denoising performance by selectively emphasizing relevant features.

In the second configuration, Self-Attention (SA) is applied within the decoder block of the generator network. The self-attention mechanism enables the network to identify local and global dependencies within the image, which leads to more efficient denoising, especially in complex regions of the image.

Our last experimental model, incorporating both Channel Attention (CA) in the encoder block and Self-Attention (SA) in the decoder block of the generator network. This dual-attention mechanism is designed to leverage the strengths of both CA and SA, aiming to produce the highest-quality denoised images by effectively managing the complexity of the noise and the underlying image features.

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1. Chapter Overview

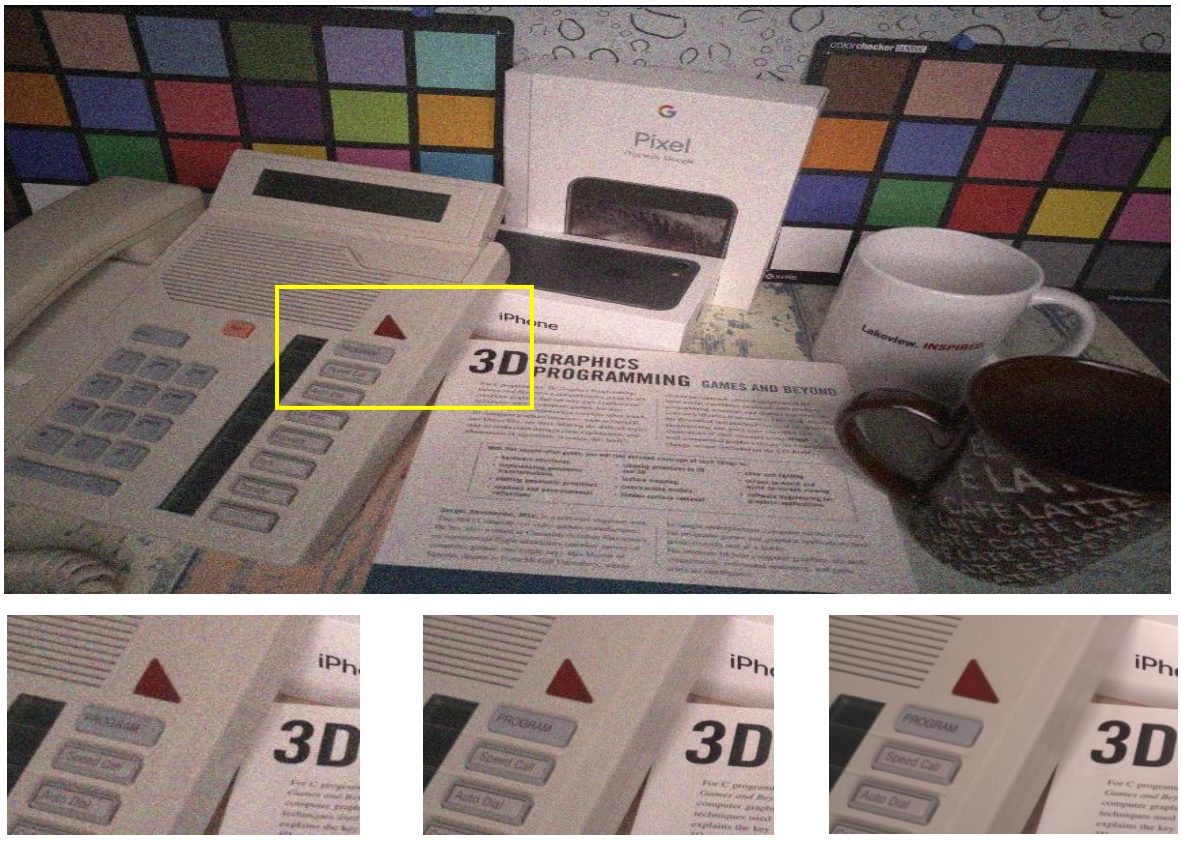
This chapter discusses about the outcomes of our implementation, including the newly proposed method. At last, the results are systematically illustrated and compared through the use of tables, figures, and graphs.

6.2. Experimental Results

In order to get better results for our study we conducted four experiments, which all of them were conducted the same datasets, hardware and software setup, to ensure the results are reliable. We present the experimental results obtained from the different GAN-based models tested in our research on real image noise reduction through attention mechanisms in GAN. Our experiments were conducted using the SIDD dataset, which provides a challenging benchmark for evaluating image denoising techniques. The proposed ANR-GAN provides the best quantitative values based on quantitative evaluation metrics as shown by a comparison of the scores with previous method. These results from the experiments we conducted are discussed below with figures.

6.2.1. CA+DGAN (ANR-GAN1)

In this experiment, for the enhancement of the denoised image's quality by preserving structural details, channel attention was utilized in encoder block and it enables the model to concentrate on most relevant features. Figure 6-1 shows the denoising result we obtained from the experiment that used CA+DGAN (ANR-GAN-1) model. As the figure illustrates the image shows some improvement but it can be enhanced more.



Noisy **ANR-GAN-1** **GT**

Figure 6-1: Results of ANR-GAN1

6.2.2. SA+D-GAN (ANR-GAN-2)

Figure 6-2 illustrates the output of SA+DGAN (ANR-GAN2) model, where self-attention (SA) was employed within the decoder block. The images demonstrate a further improvement in denoising performance, especially in capturing local and global dependencies of the image. This leads to more coherent and visually accurate reconstructions, with the denoised images displaying finer details and a higher similarity to the clean image compared to both the baseline and CA+D-GAN.

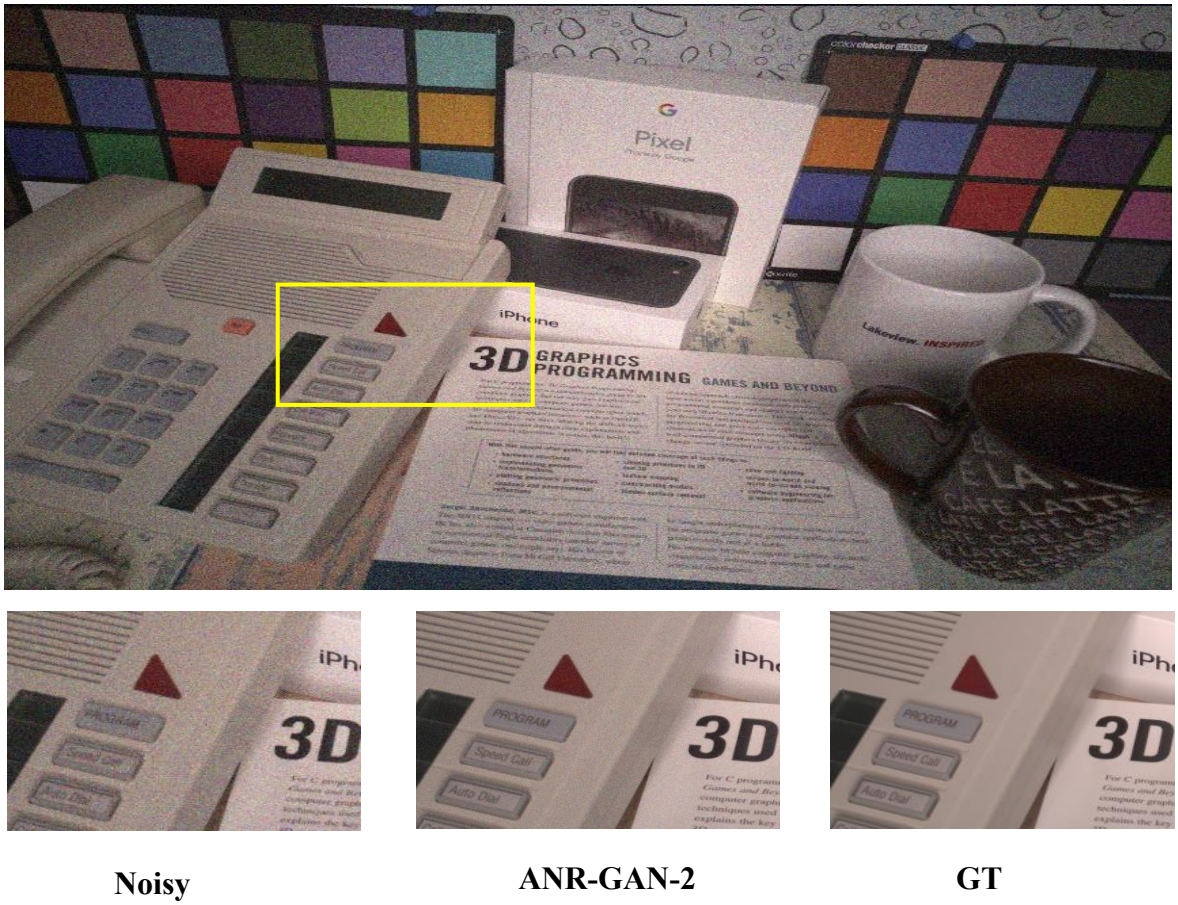
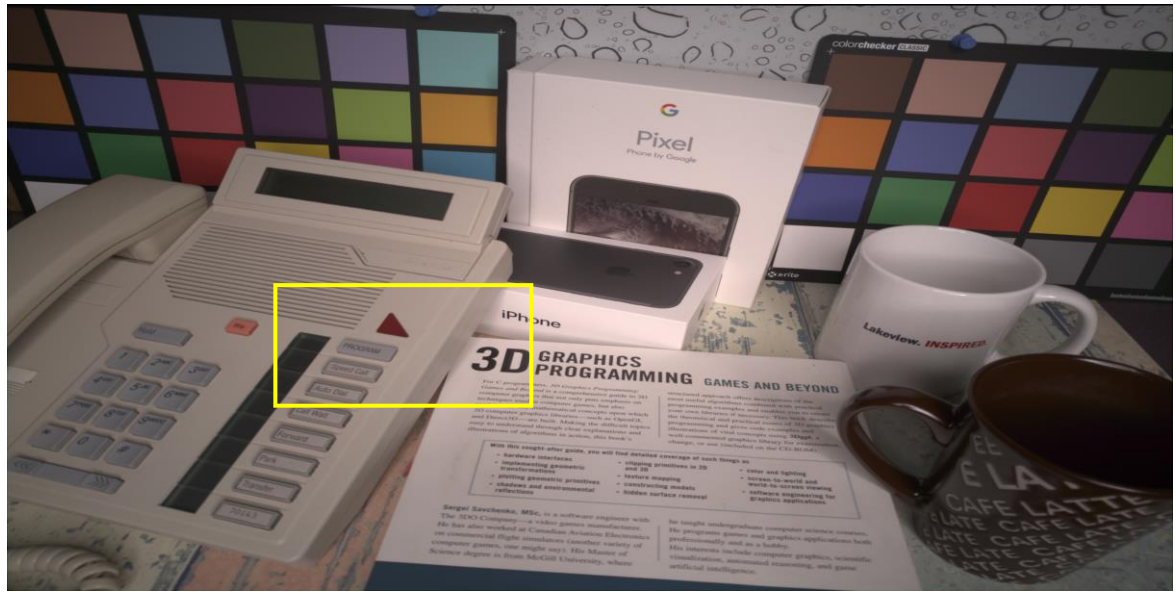
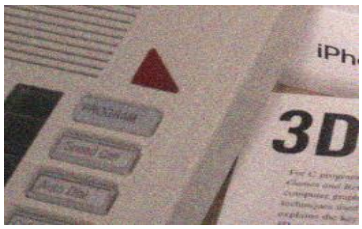


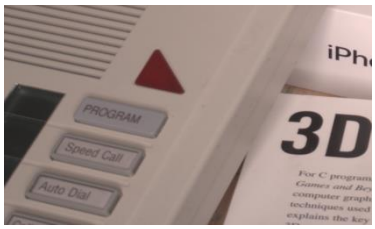
Figure 6-2: Results of ANR-GAN-2

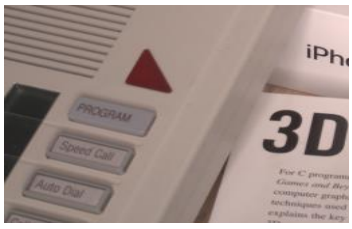
6.2.3. CA+SA+D-GAN (ANR-GAN-3)

The images in Figure 6-3 show the results of the CA+SA+D-GAN model, which combines channel attention and self-attention mechanisms. This model scores the highest quality denoised images among all the experiments. The fusion between CA and SA results in images that closely resemble the ground truth, with minimal noise and excellent preservation of fine details. The enhanced clarity and reduced noise in these images highlight the effectiveness of combining these attention mechanisms within the GAN framework. Additionally, the CA+SA+D-GAN model demonstrates a superior ability to maintain texture consistency and structural integrity, particularly in complex regions of the image. This combination ensures that both fine-grained local features and broader contextual information are accurately captured, leading to a significant improvement in overall image quality.









Noisy **ANR-GAN-3** **GT**

Figure 6-3: Results of ANR-GAN-3

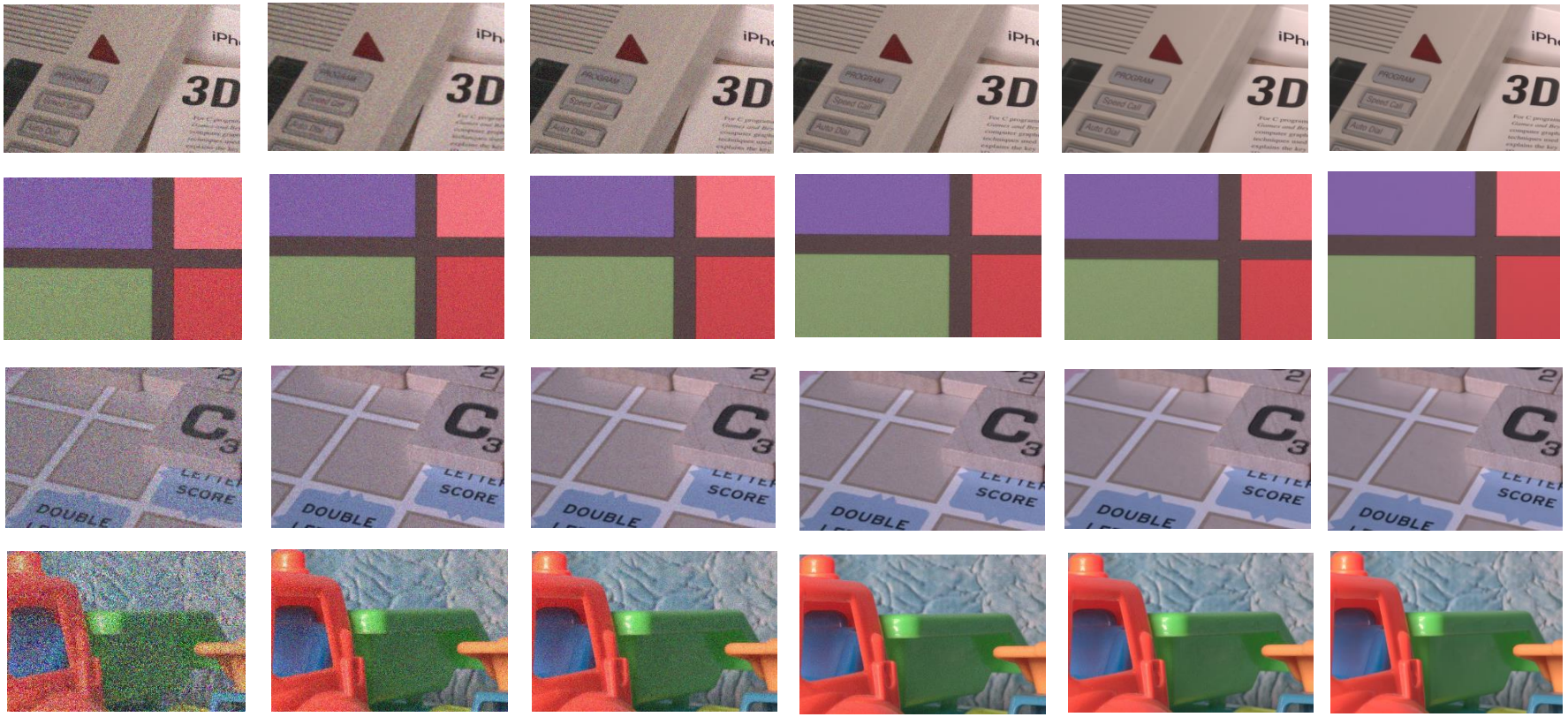


Figure 6-4: Model comparison with SIDD dataset

6.3. Sample training log for ANR-GAN

We employed visual representations of log figures, such as the generator and discriminator losses, to track the model's performance. These logs provide insight into how effectively the model is performing during training in terms of noise removal and the generation of denoised images.

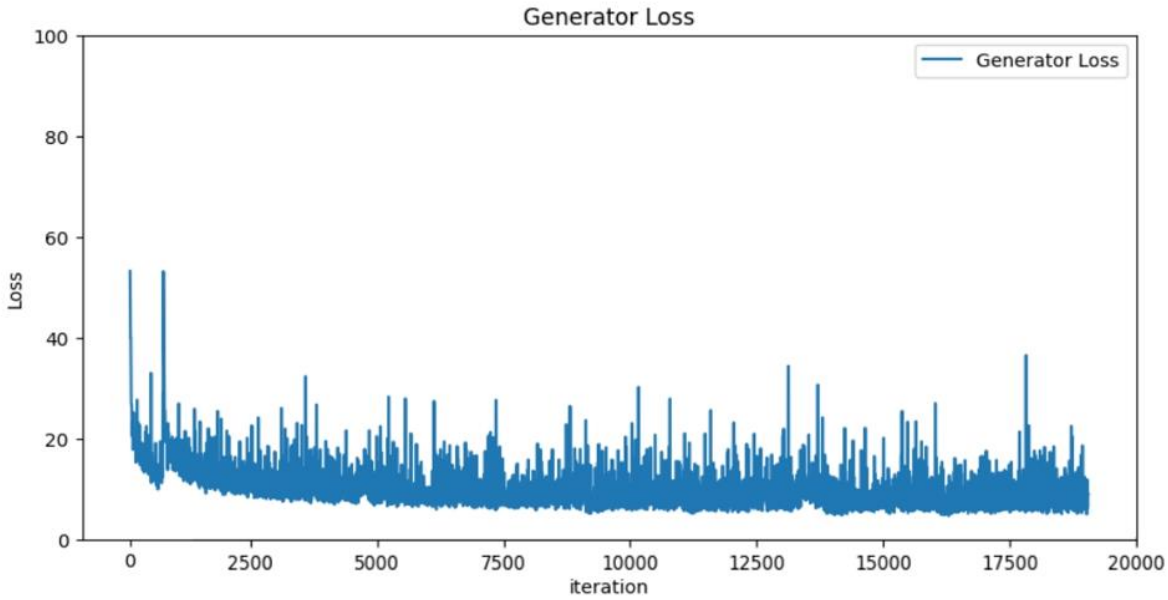


Figure 6-5: Generator loss graph

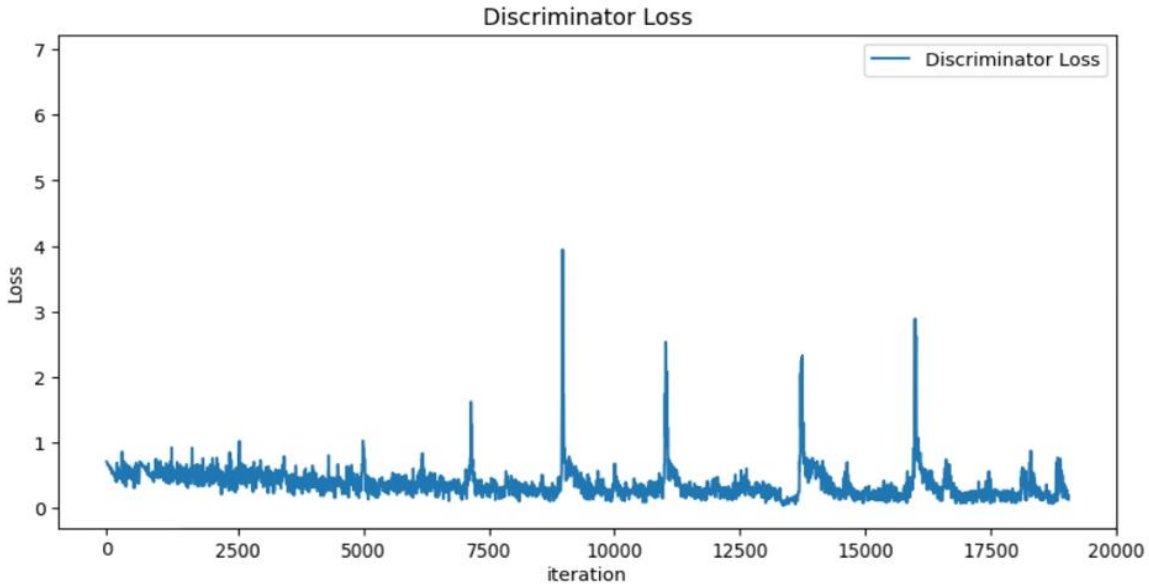


Figure 6-6: Discriminator loss graph

6.4. Results based on Evaluation Metrics

To objectively assess the performance of the proposed models, two widely accepted metrics were utilized: PSNR and SSIM, both of which are particularly effective for evaluating image denoising models. PSNR measures how well the denoised image maintains the structural integrity of the original, quantifying the difference between the generated image and the ground truth image based on pixel intensity. High PSNR value suggests that the generated image closely resembles the original, with lower noise levels and better preservation of image details. SSIM goes beyond pixel-level comparison and evaluates the perceived quality by taking consideration of changes in luminance, structural information, and contrast. This metric is essential for evaluating how well the denoised image retains the spatial relationships and fine details of the original, such as edges, textures, and other critical features. The complementary nature of PSNR and SSIM offers a thorough evaluation of our model's ability to reduce noise and preserving key characteristics of the ground truth images. Table 6-1 presents the PSNR and SSIM results from our experiments on the SIDD dataset, highlighting the effectiveness of our proposed models in reducing real image noise. The higher values in these metrics indicate that our models generate denoised images that have resemblance of the original in both quantitatively and perceptually, showcasing their robustness and reliability for practical applications.

6.4.1. Evaluation metrics for the experiments

Table 6-1: PSNR and SSIM values for the experiments

Number	***Experiments***	***PSNR***	***SSIM***
1.	D-GAN	35.78	0.9190
2.	CA+D-GAN (ANR-GAN-1)	36.17	0.9342
3.	SA+D-GAN (ANR-GAN-2)	36.33	0.9373
4.	CA+SA+D-GAN (ANR-GAN-3)	**37.42**	**0.9582**

The bold indicates the best value from the experiments

6.5. Results Discussion

The PSNR and SSIM value comparison is made in the figure below. The baseline model, D-GAN, achieved a PSNR of 35.78 and an SSIM of 0.9190. These values indicate that while the model performs reasonably well in reducing noise, this can still be improved in terms of both structural similarity and the quality of the denoised image. In the second experiment

the addition of channel attention to D-GAN results in a good improvement, with the PSNR increasing to 36.17 and SSIM to 0.9342. The increase in PSNR suggests that the channel attention mechanism helps in preserving more of the original image's details, reducing noise more effectively. The improvement in SSIM indicates that CA also enhances the ability of model to retain structural integrity of the images. In our third experiment, when self-attention is integrated into D-GAN, the performance sees a more significant increase, with a PSNR of 36.33 and SSIM of 0.9373. The SA mechanism enables the model to identify local and global dependencies within the image, and this leads to better noise reduction and a more accurate reconstruction of the image's fine details. This improvement highlights the effectiveness of SA in understanding complex image structures. In the last experiment he combination of both channel attention and self-attention got the best results, with a PSNR of 37.42 and SSIM of 0.9582. This indicates that the effect of CA and SA provides a comprehensive approach to denoising, capturing both relevant features at a granular level and understanding broader image contexts. The high SSIM value suggests that this model generate images that are not only less noisy but also structurally very close to the original, making it the most effective approach among the experiments. The improvement in PSNR and SSIM across the experiments demonstrates the importance of incorporating attention mechanisms in GAN-based denoising tasks.

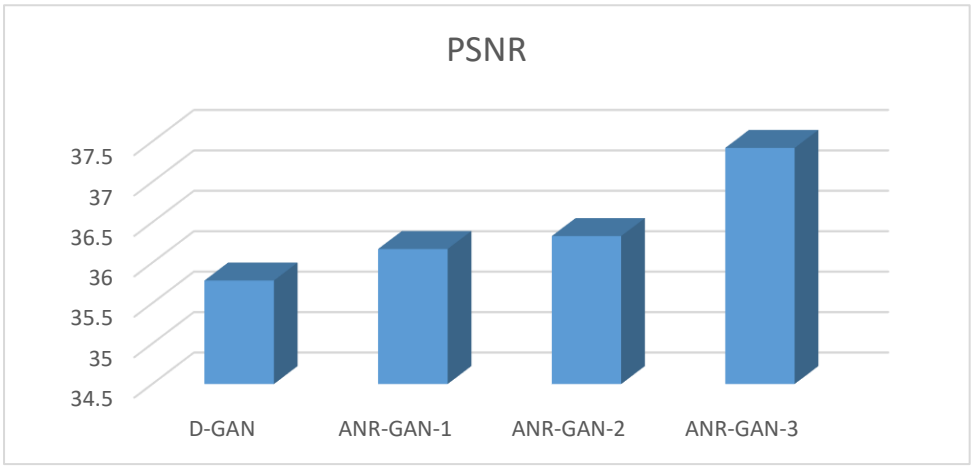


Figure 6-7: PSNR score of the four models

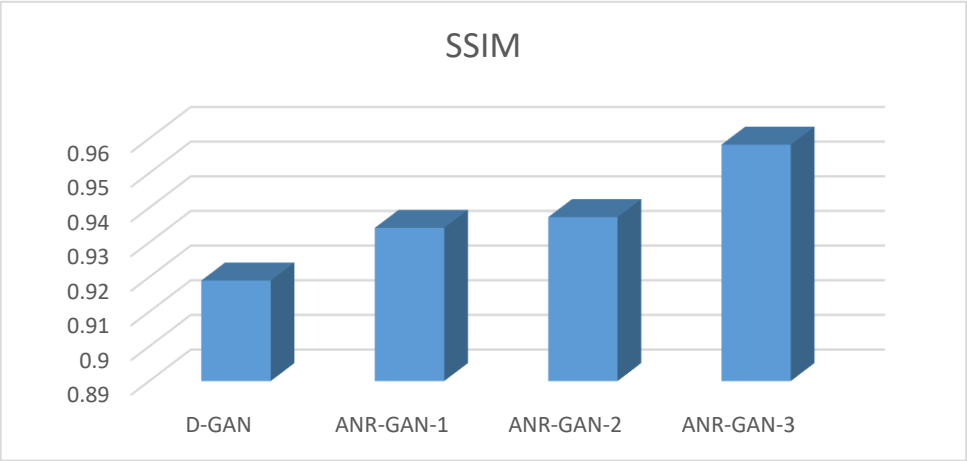


Figure 6-8: SSIM score of the four models

As shown in the figures above the results from our experiments show increase in the performance of the denoising GAN models as attention mechanisms are incorporated.

6.6. Research Questions Discussion

Our study addressed the questions that appeared in the first chapter. Here is how this study responded to the questions.

Answer to Q1: The integration of attention mechanisms, which are channel attention and self-attention, in the GAN framework significantly enhances the denoising process for real-world images. The results experimental clearly show that models incorporating these mechanisms outperform the baseline model. Our models showed improved PSNR and also SSIM values in contrast to D-GAN, indicating better noise removal and preserving image details. This indicates that combining CA and SA allows the model to concentrate on important local features while capturing global dependencies, resulting in increased noise reduction and detail preservation.

Answer to Q2: Effective learning constraints are crucial to enhance the quality of denoised images in GAN-based models. In this research we employed adversarial loss, pixel-wise loss function, particularly *L1* loss, attention mechanisms to achieve better denoising results. These learning constraints, when combined, significantly enhance the denoising performance of the model, resulting in images that are not only clean but also resemble the original images.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORKS

7.1. Conclusion

Recently the advancements of GANs have greatly impacted various tasks in computer vision, including denoising images. This paper, presents an approach to real image noise reduction by integrating attention mechanisms within a GAN framework. Our approach leverages a U-Net-based generator with embedded attention mechanisms, specifically channel attention in the encoder and self-attention in the decoder, to focus on both local and global dependencies in the image. This helps in enhancing the denoising performance by maintaining significant image structures and details.

The primary challenge in image denoising lies in balancing noise reduction with the preservation of fine details, which becomes particularly difficult in complex images. Our method addresses this by introducing a hybrid attention mechanism within the generator network, which enables the model to selectively emphasize relevant features while suppressing irrelevant noise. Additionally, our approach uses a standard discriminator which distinguishes between real and generated images.

The experimental framework demonstrates a progressive improvement in model performance through the incremental integration of attention mechanisms. The baseline model, referred to as D-GAN, serves as the foundation for comparison. As we introduced channel attention in the encoder, leading to the CA+D-GAN model, we observed enhanced focus on essential image features, which improved noise suppression without compromising important details.

Further enhancements were achieved by integrating self-attention in the decoder, resulting in the SA+D-GAN model. This addition allowed the network to maintain a better global understanding of the image, which is crucial for preserving structural coherence while denoising. Finally, the combination of both channel and self-attention mechanisms in the last CA+SA+D-GAN (ANR-GAN-3) model led to our proposed architecture. This comprehensive approach provided the balance between removing images noises and preserving details.

The experiments carried out in this research highlight the effectiveness of our approach in noise reduction while preserving the image quality. Our models showed significant

improvements in both evaluation metrics we utilized, indicating superior performance compared to existing methods. These findings imply the potential of incorporating attention mechanisms in GANs for real image noise reduction, offering potential uses in fields like photography, video enhancement, and medical imaging.

7.2. Future Works

This research has laid a foundation for real world image noise reduction using GANs with attention mechanisms. Moving forward, we propose exploring the integration of our model into real-time image processing systems, particularly in areas requiring high-quality image reconstruction, such as security and surveillance.

In addition to integrating the proposed model into real-time applications, future work could focus on further refining the architecture of the model for its performance to be enhanced. One exploration could involve experimenting with different attention mechanisms, such as spatial attention or cross-attention, to complement the existing channel and self-attention layers. These modifications could potentially improve the abilities of the model to identify complex dependencies in images, leading to even better denoising results. additionally considering unsupervised or self-supervised learning methods to further reduce the need for paired noisy-clean image datasets, enabling the model to learn directly from real-world noisy data. These future directions hold significant potential to enhance the robustness of GAN-based solutions and improve the field of image denoising in real-world applications.

SPECIAL ACKNOWLEDGMENT

This research work was funded by Adama Science and Technology University with grant number ASTU/SM-R/1055/24.

REFERENCES

Abdelhamed, A., Lin, S., & Brown, M. S. (2018). A High-Quality Denoising Dataset for Smartphone Cameras. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1692–1700. https://doi.org/10.1109/CVPR.2018.00182

Alsaiari, A., Rustagi, R., Alhakamy, A., Thomas, M. M., & Forbes, A. G. (2019). Image Denoising Using A Generative Adversarial Network. *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*, 126–132. https://doi.org/10.1109/INFOCT.2019.8710893

Anwar, S., & Barnes, N. (2019). *Real Image Denoising with Feature Attention*. http://arxiv.org/abs/1904.07396

Bovik, A., Wang, Z., & Sheikh, H. (2005). *Structural Similarity Based Image Quality Assessment* (pp. 225–241). https://doi.org/10.1201/9781420027822.ch7

Bovik Alan. (2005). *Handbook of Image and Video Processing*. Elsevier. https://doi.org/10.1016/B978-0-12-119792-6.X5062-1

Brock, A., Donahue, J., & Simonyan, K. (2018). *Large Scale GAN Training for High Fidelity Natural Image Synthesis*.

Buades, A., Coll, B., & Morel, J. M. (2005a). A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, *4*(2), 490–530. https://doi.org/10.1137/040616024

Buades, A., Coll, B., & Morel, J.-M. (2005b). A Non-Local Algorithm for Image Denoising. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 60–65. https://doi.org/10.1109/CVPR.2005.38

Chen, Q., Li, M., Chen, C., Zhou, P., Lv, X., & Chen, C. (2023). MDFNet: application of multimodal fusion method based on skin image and clinical data to skin cancer classification. *Journal of Cancer Research and Clinical Oncology*, *149*(7), 3287–3299. https://doi.org/10.1007/s00432-022-04180-1

Cui, J., Gong, K., Guo, N., Wu, C., Meng, X., Kim, K., Zheng, K., Wu, Z., Fu, L., Xu, B., Zhu, Z., Tian, J., Liu, H., & Li, Q. (2019). PET image denoising using unsupervised deep learning. *European Journal of Nuclear Medicine and Molecular Imaging*, *46*(13), 2780–2789. https://doi.org/10.1007/s00259-019-04468-4

Dong, H.-W., & Yang, Y.-H. (2019). *On Output Activation Functions for Adversarial Losses: A Theoretical Analysis via Variational Divergence Minimization and An Empirical Study on MNIST Classification*.

Elad, M., & Aharon, M. (2006). Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *IEEE Transactions on Image Processing*, *15*(12), 3736–3745. https://doi.org/10.1109/TIP.2006.881969

Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, *2*(1), 7. https://doi.org/10.1186/s42492-019-0016-7

Fu, B., Dong, Y., Fu, S., Wu, Y., Ren, Y., & Thanh, D. N. H. (2023). Multistage supervised contrastive learning for hybrid-degraded image restoration. *Signal, Image and Video Processing*, *17*(2), 573–581. https://doi.org/10.1007/s11760-022-02262-8

Fu, B., Zhang, X., Wang, L., Ren, Y., & Thanh, D. N. H. (2022). Double enhanced residual network for biological image denoising. *Gene Expression Patterns*, *45*, 119270. https://doi.org/10.1016/j.gep.2022.119270

Gao, F., Wang, Y., Yang, Z., Ma, Y., & Zhang, Q. (2023). Single image super-resolution based on multi-scale dense attention network. *Soft Computing*, *27*(6), 2981–2992. https://doi.org/10.1007/s00500-022-07456-3

Gong, P., Liu, J., & Lv, S. (2020). Image Denoising with GAN Based Model. *Journal of Information Hiding and Privacy Protection*, *2*(4), 155–163. https://doi.org/10.32604/jihpp.2020.010453

Gonzalez, R. C., Woods, R. E., & Masters, B. R. (2009). Digital Image Processing, Third Edition. *Journal of Biomedical Optics*, *14*(2), 029901. https://doi.org/10.1117/1.3115362

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gupta, A., Bhateja, V., Srivastava, A., & Gupta, A. (2019). *Suppression of Speckle Noise in Ultrasound Images Using Bilateral Filter* (pp. 735–741). https://doi.org/10.1007/978-981-13-1742-2_72

Hashimoto, F., Ohba, H., Ote, K., Teramoto, A., & Tsukada, H. (2019). Dynamic PET Image Denoising Using Deep Convolutional Neural Networks Without Prior Training Datasets. *IEEE Access*, *7*, 96594–96603. https://doi.org/10.1109/ACCESS.2019.2929230

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

Huang, B., Liu, J., Liu, X., Liu, K., Liao, X., Li, K., & Wang, J. (2023). Improved YOLOv5 Network for Steel Surface Defect Detection. *Metals*, *13*(8), 1439. https://doi.org/10.3390/met13081439

Isa, I. S., Sulaiman, S. N., Mustapha, M., & Darus, S. (2015). Evaluating Denoising Performances of Fundamental Filters for T2-Weighted MRI Images. *Procedia Computer Science*, *60*, 760–768. https://doi.org/10.1016/j.procs.2015.08.231

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2016). *Image-to-Image Translation with Conditional Adversarial Networks*.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jenkins, J., & Roy, K. (2024). Exploring deep convolutional generative adversarial networks (DCGAN) in biometric systems: a survey study. *Discover Artificial Intelligence*, *4*(1), 42. https://doi.org/10.1007/s44163-024-00138-z

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). *Progressive Growing of GANs for Improved Quality, Stability, and Variation*.

Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*.

Lan, R., Zou, H., Pang, C., Zhong, Y., Liu, Z., & Luo, X. (2021). Image denoising via deep residual convolutional neural networks. *Signal, Image and Video Processing*, *15*(1), 1–8. https://doi.org/10.1007/s11760-019-01537-x

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 105–114. https://doi.org/10.1109/CVPR.2017.19

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). *Noise2Noise: Learning Image Restoration without Clean Data*.

Li, G., Lv, J., & Wang, C. (2021). A Modified Generative Adversarial Network Using Spatial and Channel-Wise Attention for CS-MRI Reconstruction. *IEEE Access*, *9*, 83185–83198. https://doi.org/10.1109/ACCESS.2021.3086839

Ma, R., Li, S., Zhang, B., & Li, Z. (2022). *Generative Adaptive Convolutions for Real-World Noisy Image Denoising*. www.aaai.org

Mao, X.-J., Shen, C., & Yang, Y.-B. (2016). *Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections*.

Marnerides, D., Bashford-Rogers, T., & Debattista, K. (2020). *Spectrally Consistent UNet for High Fidelity Image Transformations*.

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). *Spectral Normalization for Generative Adversarial Networks*.

Nemni, E., Bullock, J., Belabbes, S., & Bromley, L. (2020). Fully Convolutional Neural Network for Rapid Flood Segmentation in Synthetic Aperture Radar Imagery. *Remote Sensing*, *12*(16), 2532. https://doi.org/10.3390/rs12162532

Öngün, C., & Temizel, A. (2019). *Paired 3D Model Generation with Conditional Generative Adversarial Networks*.

Pardasani, R., & Shreemali, U. (2018). *Image Denoising and Super-Resolution using Residual Learning of Deep Convolutional Network*.

Park, S.-W., Ko, J.-S., Huh, J.-H., & Kim, J.-C. (2021). Review on Generative Adversarial Networks: Focusing on Computer Vision and Its Applications. *Electronics*, *10*(10), 1216. https://doi.org/10.3390/electronics10101216

Plötz, T., & Roth, S. (2017). *Benchmarking Denoising Algorithms with Real Photographs*.

Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*.

Solangi, S. A., Cao, Q., Solangi, S., Solangi, T., Zaheer, &, & Dayo, A. (2017). IMAGE DENOISING METHODS: LITERATURE REVIEW. In *IJRRAS* (Vol. 32, Issue 1). www.arpapress.com/Volumes/Vol32Issue1/IJRRAS_32_1_01.pdf

Tian, C., Xu, Y., Li, Z., Zuo, W., Fei, L., & Liu, H. (2020). Attention-guided CNN for image denoising. *Neural Networks*, *124*, 117–129. https://doi.org/10.1016/j.neunet.2019.12.024

Tran, L. D., Nguyen, S. M., & Arai, M. (2021). *GAN-Based Noise Model for Denoising Real Images* (pp. 560–572). https://doi.org/10.1007/978-3-030-69538-5_34

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, 1096–1103. https://doi.org/10.1145/1390156.1390294

Vyas, A., Yu, S., & Paik, J. (2018). *Fundamentals of Digital Image Processing* (pp. 3–11). https://doi.org/10.1007/978-981-10-7272-7_1

Wang, Y., Luo, S., Ma, L., & Huang, M. (2023). RCA-GAN: An Improved Image Denoising Algorithm Based on Generative Adversarial Networks. *Electronics*, *12*(22), 4595. https://doi.org/10.3390/electronics12224595

Wang, Y., Tao, X., Qi, X., Shen, X., & Jia, J. (2018). *Image Inpainting via Generative Multi-column Convolutional Neural Networks*.

Wang, Z., Wang, L., Duan, S., & Li, Y. (2020). An Image Denoising Method Based on Deep Residual GAN. *Journal of Physics: Conference Series*, *1550*(3), 032127. https://doi.org/10.1088/1742-6596/1550/3/032127

Xie, J., Xu, L., & Chen, E. (2012). *Image denoising and inpainting with deep neural networks*.

Yan, H., Chen, X., Tan, V. Y. F., Yang, W., Wu, J., & Feng, J. (2019). *Unsupervised Image Noise Modeling with Self-Consistent GAN*. http://arxiv.org/abs/1906.05762

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2018). *Self-Attention Generative Adversarial Networks*. http://arxiv.org/abs/1805.08318

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). *Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising*. https://doi.org/10.1109/TIP.2017.2662206

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). *Residual Dense Network for Image Super-Resolution*.

Zhao, H., & Xu, Y. (2023). Research on Re-recognition Method of Multi-branch Fusion Attention Mechanism for Occluded Pedestrian. *2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, 477–480. https://doi.org/10.1109/ICCCBDA56900.2023.10154800

Zhao, S., Wang, Y., Yang, Z., & Cai, D. (2019). *Region Mutual Information Loss for Semantic Segmentation*.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*.