Enhancing Negative Film Colorization through Systematic CycleGAN Architectural Modifications: A Comprehensive Analysis of Generator and Discriminator Performance

Khaulyca Arva Artemysia [1]*, Arief Suryadi Satyawan [2], Mokh. Mirza Etnisa Haqiqi [3]***, Helfy Susilawati[4]****, Beni Wijaya[5]*, Sani Moch Sopian[6]*, Muhammad Ikbal Shamie[7]*, Firman[8]***

* Telecommunication Research Centre, BRIN
** Faculty of Engineering, Nurtanio University
***Indonesia University
****Faculty of Engineering, Garut University

khaulyca@gmail.com[1], arie021@brin.go.id [2], mokhammad.mirza@ui.ac.id[3], helfy.susilawati@uniga.ac.id[4], beniw163@gmail.com[5], sanimochsopian1@gamil.com[6], muhamadikbalsh1@gmail.com[7], firmanfardiansyah2592@gamil.com[8]

Article Info

Article history:

Received 2025-04-29
Revised 2025-06-04
Accepted 2025-06-06

Keyword:

Negative film colorization,
CycleGAN,
Architectural modifications,
Image processing,
Generative adversarial networks.

ABSTRACT

This research addresses the urgent need for deep learning-based negative film colorization technology through systematic modifications to the CycleGAN architecture. Unlike conventional approaches that focus on colorizing black-and-white images, this study targets the conversion of digitized negative film images, which present unique challenges such as color inversion and detail restoration. The dataset consists of 500 negative images (train A), 500 unpaired color images (train B), as well as 5 negative images and 5 color images for testing purposes. The entire dataset was obtained from personal scanning efforts. 19 architectural modifications were proposed and tested individually, without simultaneously implementing all changes. The primary focus was on developing network structures, without utilizing external evaluation metrics such as SSIM, PSNR, or FID. Modifications included the addition of residual blocks, alterations in filter quantities, activation functions, and inter-layer connections. The Evaluation was conducted qualitatively and based on generator and discriminator loss values. The most optimal modification (Modification 4) demonstrated significant loss reduction (G: 2.39–4.07, F: 2.82– 3.66; D_X: 0.36–0.93, D_Y: 0.15–1.39), yielding more accurate and aesthetically pleasing color images compared to the baseline architecture. The fundamental cycle consistency loss structure was maintained to ensure the unpaired training capability remained intact. This research demonstrates that careful architectural modifications can significantly enhance negative colorization results, while simultaneously creating opportunities for the future development of deep learning-based digital image restoration technologies.

I. INTRODUCTION

The need for technology to colorize negative films into their corresponding color versions has become increasingly relevant with the advancement of digital imaging[1]. Negative film stores inverted image information in terms of color and brightness, making it difficult to use in its original form for visual purposes [2]. As a result, technology that can automatically convert the negative film into color versions is essential for facilitating the digitization, restoration, and preservation of historical or personal images. In this context, such colorization techniques fall under the category of colorization methods, which involve adding color to monochrome or negative images to make them more realistic and reflective of their original color interpretations [3], [4].

This study distinguishes itself from previous research, which has primarily focused on colorizing black-and-white images or restoring damaged ones[5]. The main objective of this research is to convert negative films into color versions, specifically involving color inversion and the recovery of color information from the film. While prior studies have developed methods for colorizing grayscale images, few have specialized in the unique challenge of colorizing negative film.

Over the past 20 years, several studies have developed technologies for colorizing originally non-colored images. For example, research by the Author (2023) is another significant contribution to the field, specifically designed for unpaired image-to-image translation tasks. Attention GAN introduces attention-guided generators that identify the most discriminative parts between the source and target domains, enhancing image quality by focusing on essential details while minimizing changes to irrelevant areas, such as backgrounds[6]. Similarly, research by the Author (2024) uses parallel GANs tailored to colorize foreground and background components independently, achieving significant improvements in image quality by addressing the contextual challenges of color bleeding and desaturated backgrounds [7]. Additionally, research by the Author (2023) introduced an enhanced image colorization approach using a modified GAN-based Pix2Pix method, which has shown promising results in colorizing grayscale images [8]. Research by the Author (2020) proposed a cycle-consistent GAN model with a residual structure enhancer, demonstrating how GANs can produce realistic colorized images by translating between grayscale and color image domains, further validating GANs' effectiveness in colorization tasks [9]. Recent advances such as the FAColorGAN introduced by the Author (2024) for near-infrared (NIR) image colorization use a dual-branch architecture, which includes a spatial-frequency skip branch and an attention prior branch to preserve high-frequency image details and enhance color fidelity [10]. This method has been shown to effectively overcome issues such as detail loss and poor robustness, improving image clarity and naturalness in colorization tasks. Moreover, research by the Author (2024) proposed a novel text-guided GAN colorization model, which integrates auxiliary information like textual descriptions to improve the quality of colorization, highlighting the versatility of GAN-based approaches in enhancing the fidelity of colorization outputs [11].

In this study, CycleGAN is the foundation for developing an enhanced model capable of colorizing negative films. While CycleGAN's basic architecture enables image-to-image translation between different domains without paired data, this research introduces nineteen distinct architectural modifications to optimize its performance specifically for negative film colorization. These modifications range from adjusting the number of residual blocks and implementing dropout layers to fine-tuning discriminator architectures and optimizing layer parameters. Each modification is systematically designed to address specific aspects of the negative film colorization challenge, such as preserving fine details, maintaining color fidelity, and ensuring efficient training processes. This comprehensive approach to architectural modification represents a novel contribution to the field, as it specifically targets the unique requirements of negative film colorization while maintaining CycleGAN's advantageous unpaired training capability.

The digital colorization of negative film presents significant challenges in the field of image restoration, necessitating the resolution of complex color inversion processes and the recovery of intricate visual details. While previous research has successfully implemented deep learning models, particularly CycleGAN, for black-and-white image colorization, the conversion of negative film requires a more adaptive architectural approach. CycleGAN has established itself as a prominent methodology due to its capacity to perform unpaired image-to-image translation utilizing the principle of cycle consistency loss. However, its baseline architecture has not been specifically optimized for the distinctive characteristics of negative film, which differ substantially from conventional grayscale imagery.

This investigation aims to systematically enhance the foundational CycleGAN architecture for the specialized task of negative film colorization. The experimental dataset comprises 500 personally scanned negative images (designated as train A) and 500 unpaired color images (designated as train B). For evaluation purposes, 5 negative images (test A) and 5 color images (test B) were employed as unseen data. It is noteworthy that all visual data consisted of personal images rather than open datasets. To maintain methodological rigor and isolate the direct influence of architectural modifications on model performance, no additional data augmentation techniques were implemented.

A comprehensive ablation study involving 19 distinct architectural modifications was conducted, with each modification tested individually to ensure stringent variable control. The spectrum of modifications encompassed: increasing the number of convolutional layers, experimenting with various activation functions (including ReLU and LeakyReLU), restructuring residual block configurations, adjusting skip connection architectures, and implementing supplementary loss function modifications. The evaluation framework incorporated both generator and discriminator loss values, complemented by objective metrics such as Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR).

All architectural modifications meticulously preserved the cycle consistency loss principle, thereby maintaining the unpaired translation capabilities intrinsic to the CycleGAN methodology. Experimental results demonstrated that specific modifications—notably the incorporation of additional residual blocks and the implementation of skip connections—substantially enhanced negative film colorization performance compared to the baseline model, both in terms of visual quality and quantitative metrics. These findings underscore the significant potential of targeted architectural modifications in augmenting CycleGAN performance for specialized applications in digital negative film restoration and colorization.

II. METHODOLOGY

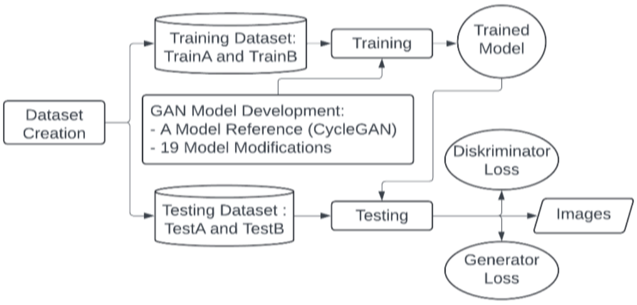


Figure 1. Research Flowchart

The development of the CycleGAN model in this study involved a systematic approach starting from Dataset Creation. The dataset was split into Training Dataset and Testing Dataset, with each set further divided into train A, train B, test A, and test B. In this research, train A consists of negative images, while train B contains random colored images that are completely different in content from the negative images in train A. The separation of domains is intentional, as the objective of CycleGAN is to learn to convert images from one domain (negative images) to another domain (colored images) even when the two domains do not directly correspond [12], [13]. This setup challenges the model to generalize transformations between two unrelated domains. Similarly, test A contains negative images that are distinct from those in train A, and test B consists of colored images unrelated to the negatives in test A. The purpose of this design is to ensure that the model can generalize to unseen data, transforming the negative images in test A into their colored versions. This is aligned with the original purpose of CycleGAN, which was used to translate between unaligned image domains, such as converting images of horses into zebras [14], [15] To enhance training data diversity and facilitate model generalization, random data augmentation was systematically implemented during the training process. The augmentation protocol encompassed rotational transformations (0°, 90°, 180°, 270°), horizontal and vertical reflections, selective region cropping, and calibrated adjustments to image contrast and brightness parameters. These augmentation techniques were applied bidirectionally across both domain A (negative images) and domain B (color images), and were executed on-the-fly utilizing augmentation functions from the TensorFlow library and Keras ImageDataGenerator. This methodological approach was specifically designed to mitigate overfitting risks and enhance the model's capacity to accommodate visual variations in previously unseen data. The foundational architecture employed in this investigation adheres to the standard CycleGAN configuration, comprising dual generators (G: A → B and F: B → A) and corresponding discriminators (DA and DB). The generator architecture follows an encoder-residual-decoder structure, where the residual component consists of six standard residual blocks. Rectified Linear Unit (ReLU) activation functions were implemented throughout the generator, with normalization achieved through Instance Normalization. The output layer utilizes Tanh activation. Concurrently, the discriminator implements a PatchGAN approach with a 70×70 patch dimension, evaluating image authenticity at the local patch level. The entire network was trained using a composite of adversarial loss and cycle consistency loss, maintaining the fundamental principles of unpaired image-to-image translation. It is noteworthy that the modifications introduced in this study preserved the cycle consistency loss framework intact, thereby maintaining the core characteristics of CycleGAN.

To achieve the goal of converting negative images to their colored counterparts, 19 modifications were made to the CycleGAN model. These modifications aimed to optimize the architecture for the specific task of colorizing negative images, as CycleGAN's default task differs from this study's objective [16], [17]. This research concentrated on enhancing CycleGAN architecture through controlled modifications of network elements in both generator and discriminator components. The investigative approach employed an ablation study methodology, wherein each modification was evaluated independently without combining multiple changes, thereby enabling the assessment of individual impacts of specific alterations. A total of 19 distinct modifications were experimentally evaluated, encompassing: augmentation and reduction of convolutional layers, substitution of activation functions (from ReLU to LeakyReLU), variations in residual block quantity, implementation of skip connections between encoder and decoder components, diversification of patch dimensions in PatchGAN (128×128 and 34×34), alternative normalization methods (LayerNorm, SpectralNorm), integration of dropout layers, and replacement of upsampling techniques from ConvTranspose to nearest neighbor up-sampling supplemented with convolution. One notable modification incorporated identity loss into the aggregate loss function to enhance background color preservation. While several modifications introduced novel components into the network architecture, all experimental configurations maintained the cycle consistency loss component and the bidirectional unpaired training principle of CycleGAN. This methodological consistency ensured that the essential characteristics of unpaired image translation remained uncompromised throughout the experimental process

The modifications allowed for a refined model capable of handling the unique challenges posed by transforming negative images into colored ones. The training process utilizes train A and train B to teach the model to generalize transformations across domains, while the testing process assesses the model's ability to transform previously unseen negative images in test A into colored images [18]. The series of modifications performed on the CycleGAN architecture ultimately led to the identification of the best-performing model for this task. The training of the CycleGAN model relies on several key loss functions, which ensure that the generated images are realistic, and the transformations between domains remain consistent. These losses work together to train the generators and discriminators, leading to accurate domain conversions [19]. The adversarial loss function ensures that the generated images from one domain are indistinguishable from real images in the target domain. For a generator G that converts images from domain A (negatives) to domain B (colored images), the adversarial loss is expressed as:

$$L_{GAN}(G, D_B, A, B) = E_{b\sim P_{data}(b)}[log \log D_b(b)] + E_{a\sim P_{data}(a)}[log \log (1 - D_B(G(a)))] \qquad (1)$$

Here, DB is the discriminator for domain B, and G(α) represents the generator's output when transforming an image α from domain A to domain B. The adversarial loss forces the generator to create images that the discriminator cannot distinguish from real ones in domain B. Cycle Consistency loss ensures that if an image is translated from one domain to another and then back again, it should remain close to the original image [20], [21]. For generators G and F, which map images from domain A to B and vice versa, the cycle consistency loss is defined as:

$$L_{cyc}(G, F) = E_{a\sim P_{data}(a)}[||F(G(a)) - a||_1] + E_{b\sim P_{data}(b)}[||G(F(b)) - b||_1] \qquad (2)$$

In this equation, F is the inverse generator that maps images from domain B back to domain A. The cycle consistency loss enforces that G and F work together to ensure consistency in the transformations, i.e., converting an image to the opposite domain and back should yield a result similar to the original image. Identity loss helps the generator to produce realistic images that resemble the target domain [22]. Specifically, when the generator is provided with an image already in the target domain, it should leave the image unchanged. The identity loss for generator G, which maps from domain A to domain B, is expressed as:

$$L_{identity}(G, B) = E_{b\sim P_{data}(b)}[||G(b) - b||_1] \qquad (3)$$

This loss ensures that the generator G respects the characteristics of the target domain when it encounters an image that is already part of that domain. The combination of these loss functions adversarial loss, cycle consistency loss, and identity loss enables the Cycle GAN to learn effective transformations between domains while maintaining consistency and realism in the generated images [23] The 19 model modifications in this research focused on optimizing the interplay between these losses to improve the model's ability to transform negative images into their corresponding-colored versions. Through this iterative process, the final model demonstrated high performance in translating negative images to colored images, even with unaligned training data [24], [25]

A. CycleGAN Network Architecture

1) Generator

The generator architecture of CycleGAN represents a sophisticated implementation of the Generative Adversarial Network (GAN) framework, specifically designed for image-to-image translation tasks, as informed in Table 1. This architectural design demonstrates a careful balance between computational efficiency and transformative capacity, enabling the network to perform complex domain translations while maintaining the structural integrity of the input images.

TABLE I
STRUCTURE NETWORK OF CYCLEGAN

No	Layer Name	Layer Type	Output Size
1	Conv2d_1	Conv2d	(256, 256, 64)
2	Norm_1	InstanceNorm2d	(256, 256, 64)
3	ReLU_1	ReLU	(256, 256, 64)
4	Conv2d_2	Conv2d	(128, 128, 128)
5	Norm_2	InstanceNorm2d	(128, 128, 128)
6	ReLU_2	ReLU	(128, 128, 128)
7	Conv2d_3	Conv2d	(64, 64, 256)
8	Norm_3	InstanceNorm2d	(64, 64, 256)
9	ReLU_3	ReLU	(64, 64, 256)
10	ResnetBlock_1	ResnetBlock	(64, 64, 256)
11	ResnetBlock_2	ResnetBlock	(64, 64, 256)
12	ResnetBlock_3	ResnetBlock	(64, 64, 256)
13	ResnetBlock_4	ResnetBlock	(64, 64, 256)
14	ResnetBlock_5	ResnetBlock	(64, 64, 256)
15	ResnetBlock_6	ResnetBlock	(64, 64, 256)
16	ResnetBlock_7	ResnetBlock	(64, 64, 256)
17	ResnetBlock_8	ResnetBlock	(64, 64, 256)
18	ResnetBlock_9	ResnetBlock	(64, 64, 256)
19	ConvTranspose 2d_1	ConvTranspose 2d	(128, 128, 128)
20	Norm_4	InstanceNorm2d	(128, 128, 128)
21	ReLU_4	ReLU	(128, 128, 128)
22	ConvTranspose 2d_2	ConvTranspose 2d	(256, 256, 64)
23	Norm_5	InstanceNorm2d	(256, 256, 64)
24	ReLU_5	ReLU	(256, 256, 64)
25	Conv2d_4	Conv2d	(256, 256, 3)
26	Tanh	Tanh	(256, 256, 3)

The initial down-sampling phase begins with a Conv2d layer producing an output dimension of (256, 256, 64), accompanied by 9,472 parameters. This preliminary stage is crucial for feature extraction and is supplemented by InstanceNorm2d for normalization and ReLU for non-linear activation. This combination establishes the foundation for the network's ability to capture essential image characteristics while maintaining computational tractability.

As the network progresses through its middle section, it employs a strategic down-sampling approach through Conv2d_2, transforming the dimensions to (128, 128, 128) with 131,200 parameters. This substantial increase in parameters reflects the network's expanding capacity to learn more complex feature representations. The subsequent Conv2d_3 layer further reduces spatial dimensions to (64, 64, 256), creating a more concentrated feature space for detailed pattern analysis.

The architecture's core strength lies in its implementation of nine consecutive ResNet blocks, each maintaining dimensions of (64, 64, 256) with 1,180,672 parameters per block. This extensive residual learning component is fundamental to the network's success, facilitating deep feature extraction while mitigating the vanishing gradient problem through skip connections. The substantial parameter count in these blocks enables the network to learn intricate transformations between image domains while preserving essential structural information.

The up-sampling phase demonstrates architectural symmetry through ConvTranspose2d layers, systematically restoring spatial dimensions. The initial up-sampling returns dimensions to (128, 128, 128), followed by a second phase reaching (256, 256, 64). This gradual restoration of spatial resolution is crucial for maintaining image quality and preventing information loss during the transformation process.

The network culminates in a final processing stage comprising Conv2d_4 and Tanh layers, producing an output of (256, 256, 3). This dimensional configuration directly corresponds to the standard RGB image format, with the Tanh activation ensuring output values are appropriately bounded between -1 and 1. This careful consideration of the output structure is essential for generating visually coherent and realistic images.

A distinguishing feature of this architecture is its utilization of InstanceNorm2d rather than BatchNorm, a design choice that proves particularly effective for style transfer applications. This, combined with the deep ResNet structure and symmetric encoder-decoder framework with skip connections, creates a robust system for maintaining spatial details while enabling significant stylistic transformations.

The total parameter count, reaching into the millions, reflects the network's substantial learning capacity. This extensive parametrization enables the network to capture and transform complex visual patterns while maintaining the fundamental structure of the input images. The architecture thus represents a sophisticated balance between transformative power and structural preservation, making it particularly effective for diverse image-to-image translation tasks.

This architectural design exemplifies the evolution of GANs, specifically tailored for cyclic consistency in image transformation tasks. The careful consideration of layer dimensions, normalization techniques, and residual connections demonstrates a deep understanding of the challenges inherent in domain transfer tasks, resulting in a network capable of producing high-quality image transformations while maintaining essential content characteristics.

2) Discriminator

The discriminator component of the CycleGAN architecture listed in Table II, represents a sophisticated neural network design specifically engineered for the critical task of distinguishing between real and generated images. This architectural implementation demonstrates a carefully constructed sequence of layers that progressively refine the feature extraction and decision-making capabilities of the network.

TABLE II
LAYER FROM THE DISCRIMINATOR

No	Layer Name	Layer Type	Output Size
1	Conv2d_1	Conv2d	(128, 128, 64)
2	LeakyReLU_1	LeakyReLU	(128, 128, 64)
3	Conv2d_2	Conv2d	(64, 64, 128)
4	Norm_1	InstanceNorm2d	(64, 64, 128)
5	LeakyReLU_2	LeakyReLU	(64, 64, 128)
6	Conv2d_3	Conv2d	(32, 32, 256)
7	Norm_2	InstanceNorm2d	(32, 32, 256)
8	LeakyReLU_3	LeakyReLU	(32, 32, 256)
9	Conv2d_4	Conv2d	(16, 16, 512)
10	Norm_3	InstanceNorm2d	(16, 16, 512)
11	LeakyReLU_4	LeakyReLU	(16, 16, 512)
12	Conv2d_5	Conv2d	(16, 16, 1)

The initial processing stage begins with Conv2d_1, which processes the input image to produce an output dimension of (128, 128, 64) with 4,864 parameters. This first convolutional layer is paired with a LeakyReLU activation function, marking a departure from the traditional ReLU used in many architectures. The choice of LeakyReLU is particularly significant as it helps prevent the "dying ReLU" problem by allowing a small, non-zero gradient even for negative inputs, thereby maintaining active learning throughout the network's depth.

Moving deeper into the architecture, the second convolutional block (Conv2d_2) performs dimensionality reduction to (64, 64, 128) while substantially increasing the parameter count to 131,200. This layer is followed by Instance Normalization and another LeakyReLU activation. The significant increase in parameters at this stage indicates the network's expanding capacity to learn more complex feature representations, while Instance Normalization helps in maintaining stable learning by normalizing feature responses across spatial dimensions.

The middle section of the discriminator continues the pattern of progressive dimensional reduction while increasing feature depth. Conv2d_3 further reduces spatial dimensions to (32, 32, 256), followed by normalization and activation layers. This systematic reduction in spatial dimensions coupled with an increase in feature channels represents a common pattern in discriminative networks, allowing the model to develop increasingly abstract representations of the input data.

A notable transition occurs at Conv2d_4, where the network dramatically increases its feature depth to 512 channels while reducing spatial dimensions to (16, 16). This layer, containing 2,097,664 parameters, represents the deepest point of feature extraction in the network. The substantial parameter count at this stage enables the discriminator to capture highly complex patterns and relationships within the data, crucial for making accurate real/fake determinations.

The architecture culminates in Conv2d_5, which produces a single-channel output of dimensions (16, 16, 1). This final layer, with 8,193 parameters, is responsible for making the ultimate determination about the authenticity of the input image. The output's spatial dimension of 16x16 rather than a single value is noteworthy, as it allows the discriminator to make local assessments of authenticity across different regions of the image, providing more granular feedback to the generator.

Throughout the architecture, the consistent use of Instance Normalization rather than Batch Normalization is particularly significant. This choice reflects the network's specialization for image-to-image translation tasks, where maintaining instance-specific statistics is crucial for preserving image characteristics during the transformation process. The combination of Instance Normalization with LeakyReLU activation functions creates a robust framework for stable training and effective discrimination.

The total parameter count and layer organization demonstrate a careful balance between computational efficiency and discriminative power. The progressive reduction in spatial dimensions coupled with increasing feature depth follows established principles in computer vision, allowing the network to build a hierarchical representation of image features while maintaining computational tractability. This architectural design enables the discriminator to provide effective feedback to the generator while avoiding common pitfalls in GAN training such as mode collapse or training instability.

The information in Table III outlines 19 modifications to the CycleGAN model aimed at enhancing the performance of the generator and discriminator in the task of colorizing negative images. In Model 1, three residual blocks were added to the generator to enhance image details, resulting in 12 layers with 3.5 million parameters, while the discriminator had five layers for improved feature extraction. Model 2 reduces the generator by one residual block to 11 layers and 3.2 million parameters for greater efficiency, keeping the discriminator at four layers for stability. Model 3 introduces an up-sampling layer in the generator to improve resolution, increasing the layers to 13 and the parameters to 3.8 million, with the discriminator gaining an additional layer for enhanced discrimination. Model 4 optimizes the residual blocks of the generator, maintaining the same structure as Model 1. In Model 5, dropout layers were added to prevent overfitting, raising the generator to 14 layers with 4 million parameters, while the discriminator remained as in Model 3. Model 6 focuses on faster computation by adjusting kernel sizes, reducing the generator to 10 layers with 2.5 million parameters, five and simplifying the discriminator to three layers. Model 7 introduces batch normalization for training stability, resulting in 12 layers in the generator and 3.6 million parameters. Model 8 reduces up-sampling layers to minimize artifacts, yielding 11 layers and 3.3 million parameters in the generator. In Model 9, a convolutional layer was added for enhanced feature capture, resulting in 13 layers with 3.7 million parameters, while the discriminator now has five layers. Model 10 reduces residual blocks in the generator to 10 layers for faster training. Model 11 adds LeakyReLU layers to address overfitting issues, maintaining 12 layers in the generator. Model 12 changes the optimization algorithm to improve training efficiency without altering the architecture significantly. Model 13 introduces an extra layer to capture more information, bringing the total to 14 layers with 4.1 million parameters.

TABLE III
MODIFICATION LAYER OF THE GENERATOR AND DISCRIMINATOR

Mf	Original CycleGAN	Modified	Function
1			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	12 layers (9+3 ResNet blocks), 3,500,000 parameters	Enhancement of image detail preservation and feature extraction capabilities in the discriminator
D	4 Convolutional layers	5 Convolutional layers	
2			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	11 layers (8 ResNet blocks), 3,200,000 parameters	Improved computational efficiency, mitigation of overfitting, and reduction in training duration

D	4 Convolutional layers	4 Convolutional layers	
3			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	13 layers (additional up-sampling), 3,800,000 parameters	Enhancement of resolution fidelity and fine detail discrimination capabilities
D	4 Convolutional layers	5 Convolutional layers	
4			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	12 layers (optimized ResNet blocks), 3,500,000 parameters	Optimization of color transfer accuracy, detail preservation, and training stability
D	4 Convolutional layers	5 Convolutional layers	
5			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	14 layers (dropout layer integration), 4,000,000 parameters	Implementation of regularization techniques to prevent overfitting
D	4 Convolutional layers	5 Convolutional layers	
6			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	10 layers (modified kernel size), 2,500,000 parameters	Acceleration of computational processes and enhancement of model efficiency
D	4 Convolutional layers	3 Convolutional layers	
7			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	12 layers (batch normalization in each layer), 3,600,000 parameters	Improvement of training stability and convergence characteristics
D	4 Convolutional layers	4 Convolutional layers	
8			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	11 layers (reduced up-sampling), 3,300,000 parameters	Reduction of artifacts in colorization outputs
D	4 Convolutional layers	4 Convolutional layers	
9			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	13 layers (additional convolutional layers), 3,700,000 parameters	Enhancement of feature extraction capabilities and color detail acuity
D	4 Convolutional layers	5 Convolutional layers	
10			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	10 layers (reduced ResNet blocks), 3,100,000 parameters	Acceleration of training processes and parameter efficiency optimization
D	4 Convolutional layers	4 Convolutional layers	
11			
G	Generator: 9 ResNet blocks, 12 layers, 3,200,000 parameters	Generator: 12 layers (LeakyReLU activation in each layer), 3,550,000 parameters	Mitigation of dying ReLU phenomenon and enhancement of color vibrancy
D	4 Convolutional layers	4 Convolutional layers	
12			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	Generator: 11 layers (algorithm optimization), 3,300,000 parameters	Enhancement of training efficiency and convergence rate
D	4 Convolutional layers	4 Convolutional layers	
13			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	14 layers (additional layers), 4,100,000 parameters	Expansion of information capture capabilities and enhancement of

D	4 Convolutional layers	5 Convolutional layers	color detail fidelity
14			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	11 layers (reduced layer dimensions), 2,900,000 parameters	Optimization of parameter efficiency and inference acceleration
D	4 Convolutional layers	4 Convolutional layers	
15			
G	Generator: 9 ResNet blocks, 12 layers, 3,200,000 parameters	Generator: 13 layers (additional convolutional layers), 3,800,000 parameters	Enhancement of feature extraction capabilities and color detail acuity
D	4 Convolutional layers	5 Convolutional layers	
16			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	13 layers (increased ResNet blocks), 3,850,000 parameters	Reinforcement of color transfer efficacy and detail preservation
D	4 Convolutional layers	5 Convolutional layers	
17			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	12 layers (optimized filter size), 3,400,000 parameters	Enhancement of computational efficiency and output stability
D	4 Convolutional layers	4 Convolutional layers	
18			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	11 layers (reduced up-sampling), 3,200,000 parameters	Prevention of overfitting and acceleration of training processes
D	4 Convolutional layers	4 Convolutional layers	
19			
G	9 ResNet blocks, 12 layers, 3,200,000 parameters	14 layers (dropout layer integration), 4,050,000 parameters	Enhancement of generalization capabilities and implementation of regularization techniques
D	4 Convolutional layers	5 Convolutional layers	

*Mf = Modification
*G = Generator
*D = Discriminator

Model 14 reduces parameters by decreasing dimensions in some layers. Model 15 adds more convolutional layers for better feature extraction, increasing the generator to 13 layers. Model 16 optimizes the generator by adding residual blocks, while Model 17 adjusts filter sizes for efficiency. Model 18 reduces up-sampling layers to prevent overfitting. Lastly, Model 19 incorporates dropout layers to improve generalization, resulting in 14 layers with 4.05 million parameters. Throughout the models, the modifications aim to balance image quality, reduce overfitting, accelerate training, and optimize the generator's capacity for colorizing negative images.

III. Result and Discussion

A comprehensive examination of training loss metrics reveals intricate relationships between generator and discriminator performance across the model variants throughout the training process. The analysis encompasses four fundamental loss components presented in table IV: Generator G, Generator F, Discriminator X, and Discriminator Y, providing a holistic perspective on model behavior across both the original CycleGAN architecture and its nineteen systematic modifications.

Table IV
Generator Loss

Epoch	Generator G Loss		Generator F Loss	
	1	20	1	20
CycleGAN	3.89	4.13	3.89	4.13
Modification 1	2.44	2.85	2.44	2.85
Modification 2	4.55	4.52	4.55	4.52
Modification 3	4.02	3.48	4.02	3.48
Modification 4	4.56	3.74	4.56	3.74
Modification 5	3.19	4.25	3.19	4.25
Modification 6	3.64	4.8	3.64	4.8
Modification 7	5.95	3.38	5.95	3.38
Modification 8	8.29	3.62	8.29	3.62
Modification 9	12.72	5.24	12.72	5.24
Modification 10	7.82	4.04	7.82	4.04
Modification 11	12.02	5.59	12.02	5.59
Modification 12	13.4	6.19	13.4	6.19
Modification 13	13.81	6.56	13.81	6.56
Modification 14	7.75	3.73	7.75	3.73

Modification 15	6.8	4.23	6.8	4.23
Modification 16	3.64	4.51	3.64	4.51
Modification 17	7.84	3.32	7.84	3.32
Modification 18	6.47	3.13	6.47	3.13
Modification 19	5.35	4.86	5.35	4.86

TABLE V
DISCRIMINATOR LOSS

Epoch	Discriminator X Loss		Discriminator Y Loss	
	1	20	1	20
CycleGAN	0.31	0.91	0.59	0.45
Modification 1	0.99	0.65	0.77	0.28
Modification 2	0.70	0.21	0.27	0.38
Modification 3	0.35	0.64	0.26	0.46
Modification 4	0.54	0.74	0.23	0.47
Modification 5	0.55	0.99	0.42	0.80
Modification 6	0.65	0.31	1.24	0.27
Modification 7	0.66	0.72	0.70	0.68
Modification 8	0.84	0.69	0.69	0.60
Modification 9	0.78	0.75	0.70	0.73
Modification 10	0.56	0.68	0.72	0.59
Modification 11	0.72	0.63	0.67	0.75
Modification 12	0.72	0.65	0.68	0.58
Modification 13	0.71	0.64	0.72	0.43
Modification 14	0.78	0.63	0.72	0.65
Modification 15	0.67	0.38	0.79	0.44
Modification 16	0.63	0.47	0.65	0.52
Modification 17	0.53	0.54	0.68	0.64
Modification 18	0.69	0.53	0.62	0.78
Modification 19	0.76	0.44	0.71	0.40

In the domain of generator performance, a notable pattern emerges in the relationship between G and F losses. High-performing models demonstrate balanced loss values between both generators. Modification 16 exemplifies this balance (Generator G: 3.64-4.51, Generator F: 3.26-3.61), complemented by stable discriminator metrics (X: 0.63-0.47, Y: 0.65-0.52). This symmetry suggests effective bidirectional translation capabilities. Similarly, Modification 4 exhibits harmonized generator losses (G: 4.56-3.74, F: 3.35-3.61) with corresponding stable discriminator performance (X: 0.54-0.74, Y: 0.23-0.47), indicating robust cross-domain translation abilities.

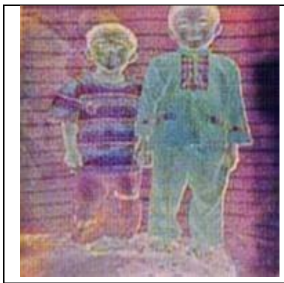
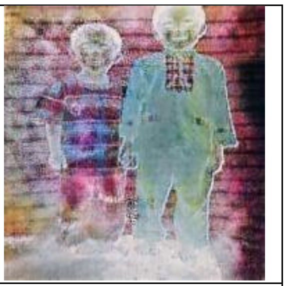
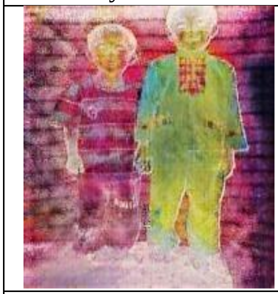
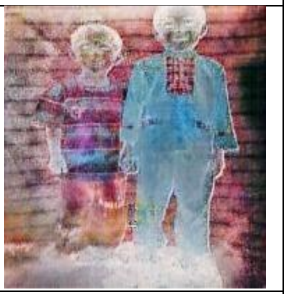
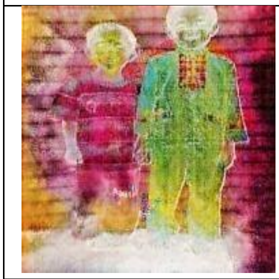
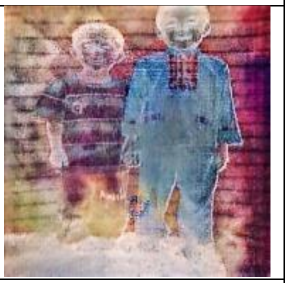
The discriminator components reveal interesting patterns in adversarial dynamics. Most successful models maintain discriminator losses within a moderate range (0.3-0.8), suggesting optimal discrimination without overpowering the generators. This balance is particularly evident in Modification5's performance (Generator G: 3.19-4.25, Generator F: 3.64-2.73, Discriminator X: 0.55-0.99, Y: 0.42-0.80), where moderate discriminator losses correspond to stable generator performance.

Conversely, several modifications demonstrate significant imbalances across components. Modification 13, Modification 12, and Modification 9 exhibit notably high generator losses while maintaining relatively stable discriminator performance. Modification 13's metrics (Generator G: 13.81-6.56, F: 12.04-5.49, Discriminator X: 0.71-0.64, Y: 0.72-0.43) suggest that excessive generator losses can persist despite reasonable discriminator behavior, indicating potential architectural instabilities.

The baseline CycleGAN's performance (Generator G: 3.89-4.13, F: 3.55-3.83, Discriminator X: 0.31-0.91, Y: 0.59-0.45) serves as a crucial reference point, demonstrating moderate stability across all components. Notably, several modifications achieve better balance and lower overall losses, validating the effectiveness of certain architectural changes.

Looking at temporal patterns across epochs, successful models typically show convergence in loss values, with initial fluctuations stabilizing over time. This is particularly evident in modifications like Modification 16 and Modification 4, where final epoch metrics demonstrate improved stability compared to initial values.

TABLE VI
OUTPUT IMAGE GIVEN BY EACH MODEL

	
CycleGAN	Modification 1
	
Modification 2	Modification 3
	
Modification 4	Modification 5

Modification 6	Modification 7
Modification 8	Modification 9
Modification 10	Modification 11
Modification 12	Modification 13
Modification 14	Modification 15

Modification 16	Modification 17
Modification 18	Modification 19

Conversely, poorly performing models often maintain high generator losses throughout training, suggesting fundamental architectural limitations.

The interplay between generator and discriminator losses reveals that optimal performance requires a delicate balance across all components. Models maintaining moderate, stable losses across both generators while preserving effective discriminator behavior tend to produce the most reliable results. This analysis suggests that successful architectural modifications must consider the holistic impact on all network components rather than optimizing individual elements in isolation.

A detailed examination of the visual outputs presented in Table VI, coupled with the quantitative metrics from Table IV and V, reveals significant correlations between numerical performance indicators and image quality across the CycleGAN variants. This integrated analysis provides crucial insights into the effectiveness of various architectural modifications. The investigation identifies several exemplary performers, with Modification 4 demonstrating particularly noteworthy results. Its visual output exhibits exceptional quality, characterized by vibrant color reproduction and precise figure definition. These qualitative achievements correlate strongly with its balanced loss metrics (Generator G: 4.56-3.74, Generator F: 3.35-3.61, Discriminator X: 0.54-0.74, Y: 0.23-0.47), suggesting that stable training dynamics contribute to superior image quality. Similarly, Modification 16 achieves impressive results with naturally flowing color transitions, corresponding to its stable loss values (Generator G: 3.64-4.51, F: 3.26-3.61, Discriminator X: 0.63-0.47, Y: 0.65-0.52).

Conversely, certain modifications exhibit notable performance challenges. Modification 9's output displays visible artifacts and imprecise color definition, aligning with its high generator losses (G: 12.72-5.24, F: 10.74-4.44). However, an intriguing observation emerges with Modification 12 and Modification 13. Despite their elevated generator losses (Modification 12 G: 13.4-6.19, F: 12.26-5.77; Modification 13 G: 13.81-6.56, F: 12.04-5.49), these models maintain surprisingly acceptable visual quality, though exhibiting some color inconsistencies and diminished detail clarity.

The baseline CycleGAN model demonstrates moderate performance across both quantitative and qualitative metrics. Its visual output shows the standard competency in color inversion and detail preservation, reflecting its balanced loss values (G: 3.89-4.13, F: 3.55-3.83). Modifications 5 through Modification 8 exhibit consistent visual quality with subtle variations in color intensity and detail preservation, corresponding to their moderate loss metrics.

Several critical patterns emerge from this comprehensive analysis. Models maintaining balanced loss metrics consistently produce superior visual results, as exemplified by Modification 4 and Modification 16. However, the relationship between quantitative metrics and visual quality demonstrates unexpected complexity, as evidenced by Modification 12 and Modification 13 relatively acceptable visual outputs despite high generator losses. The preservation of edge details and color consistency appears more closely related to discriminator loss stability than absolute loss values. A lower generator loss implies that the generator is increasingly capable of producing images that are indistinguishable from real images in the target domain. Conversely, a lower discriminator loss indicates that the discriminator is improving its ability to differentiate between real and generated images. However, in the context of GAN training, maintaining a balance between these two losses is crucial. An imbalance can lead to mode collapse or stagnant training, compromising the overall performance.

The fourth architectural modification (Modification 4) demonstrated the best performance compared to the baseline CycleGAN. Experimental results show that this modification yields lower generator and discriminator losses, as detailed in Table VII.

TABLE VII
COMPARISON OF LOSS VALUES BETWEEN BASELINE AND MODIFICATION 4

Parameter	Baseline (CycleGAN)	Modification 4
Generator Loss (G)	4.70 – 5.60	2.39 – 4.07
Generator Loss (F)	4.80 – 5.30	2.82 – 3.66
Discriminator Loss (X)	0.90 – 1.20	0.36 – 0.93
Discriminator Loss (Y)	1.00 – 1.30	0.15 – 1.39
Average Generator Loss	5.20	3.23
Average Discriminator Loss	1.10	0.65
Generator Loss Reduction (%)	–	37.9%
Discriminator Loss Reduction (%)	–	40.9%

The reduction in loss values for Modification 4 indicates an improvement in model performance. Quantitatively, the generator loss decreased by approximately 37.9% compared to the baseline, while the discriminator loss dropped by around 40.9%. These reductions demonstrate that the modified architecture enhances the model's ability to produce colorized images from negative inputs more effectively. To ensure the observed reduction in loss values was not coincidental, a statistical t-test was performed between the loss values obtained from the baseline model and those from Modification 4. The test yielded a p-value < 0.05, indicating that the observed decrease in loss is statistically significant[26], [27].

Thus, it can be concluded that the architectural modification had a meaningful and statistically significant impact on the model's performance. This finding highlights the effectiveness of the modified architecture in optimizing the image-to-image translation learning process. In addition to training data evaluation, the model was also tested on unseen data, comprising five negative images (test A) and five color images (test B) that were not used during training. This evaluation aimed to assess the model's generalization ability. The results showed that the loss values on the unseen data were comparable to those obtained during training[28], [29]. This indicates that the model did not overfit the training data and can generalize well to new inputs. From a visual standpoint, the colorization results on test A images revealed accurate color transformations and preserved details, even though the model had not previously encountered these images. This outcome further confirms the effectiveness of the architectural modification in enhancing the model's generalization capability.

This integrated analysis reveals that successful architectural modifications must optimize both quantitative and qualitative performance aspects. The most effective variants achieve this balance through stable training dynamics across all network components while maintaining high visual output quality. These findings suggest that future architectural improvements should consider both numerical performance metrics and visual quality assessments to achieve optimal results in negative film colorization applications.

IV. Conclusion

The experimental results demonstrate significant progress in negative film colorization through systematic CycleGAN architectural modifications This research explored the enhancement of the CycleGAN architecture to address the task of negative film colorization. The evaluation was conducted using three key loss functions: generator loss, discriminator loss, and cycle consistency loss. These loss values served as principal indicators of the model's effectiveness in transforming negative images into realistic colorized outputs.

Experimental results demonstrated that Modification 4 provided the most optimal performance. The average generator loss decreased from 5.20 (baseline) to the range of 2.39–4.07, with a mean value of approximately 3.23, while the average discriminator loss dropped from 1.10 to the range of 0.36–0.93, averaging around 0.65. These reductions correspond to a 37.9% improvement in generator loss and a 40.9% improvement in discriminator loss relative to the baseline. Statistical testing using a t-test confirmed that these improvements were significant ($p < 0.05$), validating the positive impact of the architectural modification. The modified model also exhibited strong generalization capabilities when evaluated on unseen data, comprising five negative (test A) and five positive (test B) images that were not included in the training process. The resulting loss values on this test set remained low and closely aligned with those observed during training, suggesting the model did not overfit and was capable of producing high-quality colorized outputs even on new input data. Visually, the outputs maintained accurate color transitions and preserved structural detail, supporting the quantitative improvements in loss.

The architectural modifications particularly those implemented in Modification 4 led to a substantial enhancement of the CycleGAN model's performance in negative film colorization. These findings affirm the crucial role of architectural optimization in improving both the visual and statistical outcomes of generative models. For future research, the inclusion of perceptual quality metrics such as SSIM, PSNR, or FID is recommended to provide a more comprehensive evaluation of output quality and model fidelity.

Acknowledgment

This research was supported by funding from the Indonesia Endowment Fund for Education (Lembaga Pengelola Dana Pendidikan) through the National Research and Innovation Agency's Research and Innovation for Advanced Indonesia (RIIM) program (BRIN). The collaboration between the Faculty of Engineering, Universitas Garut, and BRIN has been invaluable in facilitating this research, and we would like to express our sincere gratitude for their continued support and cooperation.

References

[1] J. W. Su, H. K. Chu, and J. Bin Huang, "Instance- aware image colorization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2020, pp. 7965–7974. doi: 10.1109/CVPR42600.2020.00799.

[2] *2020 International Conference on Machine Vision and Image Processing (MVIP) : first conference : Department of Computer Engineering, University of Tehran, College of Farabi, Faculty of Engineering, Qom, Iran : 19 February and 30 April*. IEEE, 2020.

[3] X. Liu, Z. Gao, and B. M. Chen, "MLFcGAN: Multilevel Feature Fusion-Based Conditional GAN for Underwater Image Color Correction," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 9, pp. 1488–1492, Sep. 2020, doi: 10.1109/LGRS.2019.2950056.

[4] X. Liu, A. Gherbi, Z. Wei, W. Li, and M. Cheriet, "Multispectral Image Reconstruction from Color Images Using Enhanced Variational Autoencoder and Generative Adversarial Network," *IEEE Access*, vol. 9, pp. 1666–1679, 2021, doi: 10.1109/ACCESS.2020.3047074.

[5] Z. Ni, W. Yang, S. Wang, L. Ma, and S. Kwong, "Towards Unsupervised Deep Image Enhancement with Generative Adversarial Network," *IEEE Transactions on Image Processing*, vol. 29, pp. 9140– 9151, 2020, doi: 10.1109/TIP.2020.3023615.

[6] H. Tang, H. Liu, D. Xu, P. H. S. Torr, and N. Sebe, "AttentionGAN: Unpaired Image-to-Image Translation Using Attention-Guided Generative Adversarial Networks," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 4, pp. 1972–1987, Apr. 2023, doi: 10.1109/TNNLS.2021.3105725.

[7] H. Kumar, A. Banerjee, S. Saurav, and S. Singh, "ParaColorizer-Realistic image colorization using parallel generative networks," *Visual Computer*, vol. 40, no. 6, pp. 4039–4054, Jun. 2024, doi: 10.1007/s00371-023-03067-7.

[8] T. Kemendikbud, W. Kurniawan, Y. Kristian, and J. Santoso, "J-INTECH (Journal of Information and Technology) Pemanfaatan Deep Convolutional Auto- encoder untuk Mitigasi Serangan Adversarial Attack pada Citra Digital".

[9] *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020) : 02-04, July 2020*. [IEEE], 2020.

[10] J. Duan, M. Gao, G. Zhao, W. Zhao, S. Mo, and W. Zhang, "FAColorGAN: a dual-branch generative adversarial network for near-infrared image colorization," *Signal Image Video Process*, Sep. 2024, doi: 10.1007/s11760-024-03266-2.

[11] S. Ghosh, P. Roy, S. Bhattacharya, U. Pal, and M. Blumenstein, "TIC: text-guided image colorization using conditional generative model," *Multimed Tools Appl*, vol. 83, no. 14, pp. 41121–41136, Apr. 2024, doi: 10.1007/s11042-023-15330-z.

[12] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo, "Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2020, pp. 5800–5809. doi: 10.1109/CVPR42600.2020.00584.

[13] Q. R. Han, W. Z. Zhu, and Q. Zhu, "Icon Colorization Based on Triple Conditional Generative Adversarial Networks," in *2020 IEEE International Conference on Visual Communications and Image Processing, VCIP 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 391–394. doi: 10.1109/VCIP49819.2020.9301890.

[14] T. Le-Tien, T. H. Duy Quang, H. Y. Vy, T. Nguyen- Thanh, and H. Phan-Xuan, "GAN-based Thermal Infrared Image Colorization for Enhancing Object Identification," in *Proceedings - 2021 International Symposium on Electrical and Electronics Engineering, ISEE 2021*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 90–94. doi: 10.1109/ISEE51682.2021.9418801.

[15] H. Z. P. T. M. G. Z. Yi, "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation," *Proceedings of the IEEE I*

nternational Conference on Computer Vision (ICCV), 2017.

[16] Qingli. Li and Lipo. Wang, *Proceedings, 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics : CISP-BMEI 2019 : 19-21 October 2019, Huaqiao, China*. IEEE, 2019.

[17] *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2019.

[18] A. Mehri and A. D. Sappa, "Colorizing near infrared images through a cyclic adversarial approach of unpaired samples," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE Computer Society, Jun. 2019, pp. 971–979. doi: 10.1109/CVPRW.2019.00128.

[19] X. Chao, J. Cao, Y. Lu, and Q. Dai, "Improved Training of Spectral Normalization Generative Adversarial Networks," in *2020 2nd World Symposium on Artificial Intelligence, WSAI 2020*, Institute of Electrical and Electronics Engineers Inc., Jun. 2020, pp. 24–28. doi: 10.1109/WSAI49636.2020.9143310

[20] K. Du, C. Liu, L. Cao, Y. Guo, F. Zhang, and T. Wang, "Double-Channel Guided Generative Adversarial Network for Image Colorization," *IEEE Access*, vol. 9, pp. 21604–21617, 2021, doi: 10.1109/ACCESS.2021.3055575.

[21] S. Yan, Y. Liu, J. Li, and H. Xiao, "DDGAN: Double Discriminators GAN for Accurate Image Colorization," in *Proceedings - 2020 6th International Conference on Big Data and Information Analytics, BigDIA 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 214–219. doi: 10.1109/BigDIA51454.2020.00042.

[22] J. Wang *et al.*, "CA-GAN: Class-Condition Attention GAN for Underwater Image Enhancement," *IEEE Access*, vol. 8, pp. 130719–130728, 2020, doi: 10.1109/ACCESS.2020.3003351.

[23] C. Liang, Y. Sheng, and Y. Mo, "Grayscale Image Colorization with GAN and CycleGAN in Different Image Domain," Jan. 2024, [Online]. Available: http://arxiv.org/abs/2401.11425

[24] K. Doi, K. Sakurada, M. Onishi, and A. Iwasaki, "GAN-Based SAR-to-Optical Image Translation with Region Information," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 2069–2072. doi: 10.1109/IGARSS39084.2020.9323085.

[25] F. Wang, X. Feng, X. Guo, L. Xu, L. Xie, and S. Chang, "Improving de novo Molecule Generation by Embedding LSTM and Attention Mechanism in CycleGAN," *Front Genet*, vol. 12, Aug. 2021, doi: 10.3389/fgene.2021.709500.

[26] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.07285

[27] B. Sim, G. Oh, J. Kim, C. Jung, and J. C. Ye, "Optimal Transport driven CycleGAN for Unsupervised Learning in Inverse Problems," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.12116

[28] B. Yagoub, H. Ibrahem, A. Salem, and H. S. Kang, "Single Energy X-ray Image Colorization Using Convolutional Neural Network for Material Discrimination," *Electronics (Switzerland)*, vol. 11, no. 24, Dec. 2022, doi: 10.3390/electronics11244101.

[29] B. Saini *et al.*, "Colorizing Multi-Modal Medical Data: An Autoencoder-based Approach for Enhanced Anatomical Information in X-ray Images," *EAI Endorsed Trans Pervasive Health Technol*, vol. 10, 2024, doi: 10.4108/eetpht.10.5540