Anime Sketch Colorization using GAN-based Techniques

Jiexin Liao, Mohammed Faysal Ahmed, and Prabin Kumar Shrestha
Memorial University of Newfoundland
jiexinl@mun.ca, mohammedfa@mun.ca, pkshrestha@mun.ca

Abstract—Anime sketch colorization, a subset of image-to-image translation in computer vision, has witnessed notable advancements, particularly through the application of Generative Adversarial Networks (GANs). This paper explores the efficacy of Pix2Pix, a conditional GAN framework, in automating and enhancing the process of anime sketch colorization. We present a comprehensive review of related works in this domain, highlighting the evolution from heuristic-based methods to deep learning techniques like GANs. Specifically, we delve into the architecture and methodology of Pix2Pix, detailing its generator and discriminator designs, loss functions, and experimental setups. Our experiments utilize a dataset of paired anime sketch and color images, employing evaluation metrics such as Structural Similarity Index (SSIM), Fréchet Inception Distance (FID), and Peak Signal-to-Noise Ratio (PSNR) to assess model performance. Results demonstrate progressive improvements in image quality and realism over training epochs, with notable enhancements observed in FID and PSNR scores. Despite challenges such as training instability and fluctuating SSIM scores, our experiments underscore the potential of Pix2Pix in automating anime sketch colorization tasks, paving the way for further research and advancements in this domain.

Index Terms—GANs, Conditional GANs (cGANs), pix2pix, Image to Image Translation, Anime Sketch Colorization.

1 Introduction

Anime, renowned for its varied genres and compelling narratives, has attained global cultural prominence, captivating audiences across demographics. Originating from manga, where character sketches hold fundamental significance, the transition to colored manga for anime production demands considerable time investment, as colored manga serves as a pivotal reference for crafting vibrant and immersive anime visuals.

The domain of anime sketch colorization, a subset of computer vision applications, particularly in image-to-image translation, has witnessed notable progress. A key technique introduced aims to automate the colorization process for grayscale anime sketches, utilizing extensive datasets to predict a range of potential colors for individual pixels. This method emphasizes the inclusion of rarely occurring colors during training to enrich diversity, positioning colorization as a pretext task for acquiring features through self-supervision.

Neural Style Transfer [1], employing convolutional neural networks (CNNs), presents an innovative approach to disentangling and recombining content and stylistic elements within images. Deep learning methodologies, notably Generative Adversarial Networks (GANs) [2] and Conditional GANs (cGANs) [3], have emerged as transformative forces in various image processing tasks, including colorization.

Pix2Pix [4], a conditional GAN framework, showcases versatility across diverse image-to-image translation assignments. Unlike traditional GANs, pix2pix not only learns the mapping from input to output images but also autonomously derives a loss function, simplifying the process and obviating the necessity for specialized formulations.

Furthermore, domain-specific adaptations such as AnimeGAN [5] and novel techniques like attention-based models have expanded the horizons of anime sketch colorization. These methodologies address issues concerning color consistency, semantic alignment, and the generation of high-fidelity anime-style imagery.

Drawing from these advancements, this study delves into the potential of pix2pix, a variation of conditional Generative Adversarial Networks (cGANs), in the field of anime sketch colorization. Utilizing its capabilities in image-to-image translation, the study aims to streamline and enhance the colorization process. The experimentation phase involves the implementation of pix2pix, utilizing a generator employing U-Net architecture [6] with skip connections in both encoder and decoder, alongside PatchGAN in the discriminator. This implementation is executed on the Anime Sketch Colorization Pair [7] dataset sourced from Kaggle. The trained model undergoes testing on the validation dataset, and are evaluated using evaluation metrics including Fréchet Inception Distance (FID) [8], Structural Similarity Index (SSIM) [9], and Peak Signal-to-Noise Ratio (PSNR) [10] for assessing its performance.

Subsequent sections of the report contains a review of prior research in the field, a detailed explanation of the methodology employed, an elaboration on the architecture of the pix2pix model, the experimental setup, examination of evaluation metrics, and an analysis of the results obtained..

2 Related Work

The domain of anime sketch colorization, a subset within the broader field of computer vision focusing on image-to-image translation, has experienced notable progressions.

A automation technique [11] was introduced to automate the process of adding color to grayscale anime sketches. Leveraging extensive datasets, this approach estimates a spectrum of potential colors for individual pixels, with particular emphasis on incorporating infrequently occurring colors during the training phase to enhance diversity. This research investigates colorization as a pretext task for acquiring features through self-supervision.

In recent times, deep learning methodologies has emerged as a transformative influence in various image processing endeavors, including colorization. A technique known as Neural Style Transfer [1], based on convolutional neural networks (CNNs), introduced an innovative strategy to disentangle and reassemble the content and stylistic elements of images. By utilizing pre-trained CNN models to extract content and style representations, coupled with an optimization process, it effectively separates content, focusing on capturing objects, from style, which captures texture and color patterns.

Generative Adversarial Networks (GANs) [2] are a class of deep learning architectures that consist of two neural networks: a generator and a discriminator, trained in an adversarial fashion. A novel approach utilizing conditional Deep Convolutional Generative Adversarial Networks (DCGAN) [12] was proposed. Unlike earlier methods that often necessitate specialized inputs or thematic constraints, the authors aimed for broad applicability across diverse datasets such as CIFAR-10 and Places365.

Conditional GANs (cGANs) [3] represent a significant advancement in the field of generative modeling. cGANs extend the foundational GAN framework to enable conditional generation, facilitating the creation of data samples contingent upon specific inputs. This extension facilitated various applications, including image-to-image translation, text-to-image synthesis, and conditional image generation.

Isola et al. [4] introduced pix2pix, a conditional Generative Adversarial Network (cGAN), as a versatile solution applicable to various image-to-image translation tasks. This framework expands upon the capabilities of cGANs to address a broad spectrum of use cases. Pix2pix networks are designed not only to learn the mapping from input images to their corresponding outputs but also to autonomously derive a loss function that guides and refines this mapping process. This inherent adaptability allows for a unified approach across diverse problems, eliminating the necessity for distinct and specialized loss formulations. Their research demonstrates the effectiveness of this approach in tasks such as generating photographs from label maps, reconstructing objects from edge maps, and colorizing images.

In addition to conditional Generative Adversarial Networks (cGANs), CycleGAN [13] has emerged as a potent technique for unsupervised image translation. CycleGANs represent a subtype of GANs capable of learning to translate images from one domain to another without necessitating paired training data [14]. Unlike conventional GANs, which rely on paired training data, CycleGANs can acquire the mapping between domains in an unsupervised manner. The fundamental concept underlying CycleGANs involves the utilization of a cycle-consistency loss mechanism to ensure coherence between generated and input images. This loss function incentivizes the generator to learn a mapping from domain A to domain B and subsequently back to domain A, aiming to produce generated images akin to the originals.

In the context of anime sketch or manga colorization, researchers have devised domain-specific adaptations of Generative Adversarial Network (GAN) architectures tailored to these tasks. For instance, Fujimoto et al. proposed AnimeGAN [5], a lightweight GAN framework specifically designed for translating photos into anime-style images. AnimeGAN incorporates domain-specific insights and architectural adjustments to facilitate high-quality anime-style image generation, including the colorization of sketches.

Hensman et al. [15] introduced an innovative approach to manga colorization utilizing conditional Generative Adversarial Networks (cGANs). Unlike previous cGAN-based methods that typically require extensive training datasets, their approach relies on a single colorized reference image for training, thus eliminating the need for a large dataset. Moreover, acknowledging the tendency of cGANs to generate blurry outcomes with artifacts and limited resolution, they proposed a segmentation and color-correction technique to mitigate these issues.

The study titled ”Attention-Aware Anime Line Drawing Colorization” [16] addresses challenges related to color consistency and semantic alignment in anime line drawing colorization. This is achieved through the introduction of an attention-based model specifically designed for this task. The proposed model integrates a Convolutional Attention module within the encoder architecture, focusing on both channel-wise and spatial-wise aspects to improve feature extraction and perception of key areas. Additionally, a Stop-Gradient Attention module is incorporated, which combines cross-attention and self-attention mechanisms to effectively manage long-range dependencies across different parts of the drawing. Comprehensive experimental evaluations demonstrate the efficacy of the proposed method compared to existing state-of-the-art approaches, showcasing enhancements in both line structure accuracy and semantic color representation within the colorized images.

GANime [17] and Bhullar et al. [18] have compared various techniques such as Neural Style Transfer, C-GAN, and CycleGAN in the domain of anime color drawing from sketch. Their results showed that pix2pix, or enhanced versions of pix2pix, or cGAN outperformed other methods.

A refined version of the pix2pix model tailored for anime manga colorization was introduced [19], using conditional Generative Adversarial Networks (cGANs). This enhanced model incorporates multi-channel feature extraction to better retain intricate details such as edges and grayscale information in complex images. Through the utilization of multichannel downsampling and fusion overlay techniques, the proposed model is adept at training on and processing complex illustrations. It prioritizes learning the color distribution and contour lines separately, facilitating the accurate coloring of complex black-and-white illustrations.

Shi et al. [20] introduces a novel comics-sketch-style transformation algorithm based on Generative Adversarial Network (GAN). The objective is to improve upon the capabilities of the Pix2Pix network by automatically generating sketch images from comics images. The study enhances the Pix2Pix network by integrating the Local Binary Pattern (LBP) algorithm as a preprocessing step to extract texture features. Additionally, the network architecture is simplified from five layers to three layers to improve the accuracy of the Generator within the U-Net architecture. Experimental findings demonstrate that the proposed algorithm surpasses Pix2Pix in generating comics-sketch-style images, both in subjective visual assessment and objective performance metrics.

While these techniques have shown promise, they also present certain limitations and challenges. Traditional heuristic-based methods lack scalability and struggle to capture the complexity of anime artworks. Deep learning techniques like cGANs require large amounts of paired training data, which may be impractical to obtain in some scenarios. Additionally, unsupervised approaches like CycleGAN may struggle with preserving fine details and texture in the generated images.

By building upon these previous works, this project aims to explore the potential of Pix2Pix in anime sketch colorization. Specifically, by leveraging the Pix2Pix model's capability for image-to-image translation, this project seeks to automate and enhance the process of anime sketch colorization, offering a viable solution to the challenges faced by traditional manual methods.

3 Methods

Generative Adversarial Networks (GANs) are a generative models that learn a mapping from random noise vector z to output image y, $G : z \rightarrow y$ [2] [4]. It consists of generator and the discriminator, which are trained simultaneously in a competitive manner. The generator network learns to generate images by mapping random noise to the data distribution of interest. On the other hand, the discriminator network learns to distinguish between real data samples and those generated by the generator. Through adversarial training, where the generator aims to deceive the discriminator while the discriminator strives to accurately classify real and fake samples, GANs learn to produce high-quality synthetic data that closely resemble real examples.

Conditional Generative Adversarial Networks (cGANs) [3], on the other hand, extend traditional GANs by incorporating additional information, or conditioning, into both the generator and discriminator networks. This conditioning allows for more controlled generation, as the generator learns to produce samples based on specific attributes provided as input. For tasks like image-to-image translation, cGANs can generate outputs consistent with the given inputs. cGANs learns a mapping from observed image x and random noise vector z, to y, $G : \{x, z\} \rightarrow y$ [4]. The mapping can be seen in Figure 1.

3.1 Pix2Pix Architecture

The generator G utilizes a U-Net architecture, characterized by its encoder-decoder framework with integrated skip connections facilitating information flow between encoder and decoder layers. On the other hand, the discriminator D uses a PatchGAN design, which evaluates N × N patches within an image to classify them as either real or fake [17].

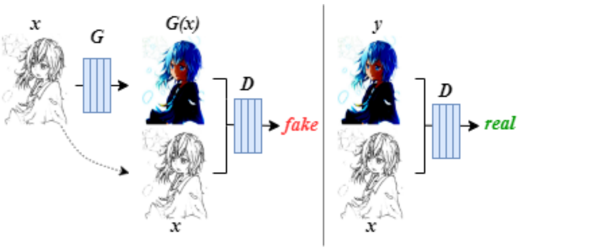

Fig. 1. Mapping of input (x), ground truth (y) and the predicted image (G(x)) with generator G and discriminator D

3.1.1 Generator Architecture

Pix2pix generator's architecture is important in image-to-image translation, particularly in converting high-resolution input sketches into corresponding high-resolution colored images. At higher level, The generator G is a U-Net architecture [6], which is an encoder-decoder with skip connections from encoder layers to decoder layers as shown in Figure 2.

In conventional encoder-decoder networks, all information must pass through each layer, including the narrowest point. However, in many image translation tasks, there's shared basic information between input and output images. To efficiently utilize this shared data and avoid congestion, pix2pix incorporates skip connections inspired by the ”U-Net” model [6]. U-Net is known for capturing detailed features and context simultaneously. In the contracting path, convolutional layers and max-pooling extract key features and shrink spatial dimensions, while the expansive path enlarges the image using upsampling layers. This framework empowers U-Net to understand intricate input images and generate precise output translations. Skip connections establish direct connections between each layer ’i’ and its corresponding counterpart at layer ’n - i’, where ’n’ represents the total number of layers. By merging all channels at layer ’i’ with those at layer ’n - i’, this design facilitates seamless information flow across different abstraction levels within the network. This enhances the generator's ability to capture fine details and subtleties during the colorization process [4].

The encoder section of the generator undertakes the crucial task of compressing input sketches into a latent space representation, thereby facilitating efficient feature extraction. Consider an input sketch of dimensions 256x256 pixels with 3 color channels (RGB). As the sketch traverses through the encoder blocks, its dimensions gradually diminish while the number of channels increases. Initially,

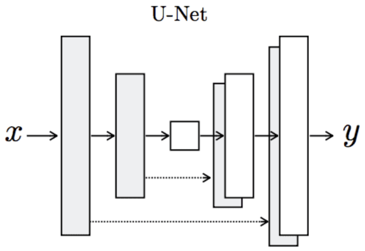

Fig. 2. The “U-Net” [6] is an encoder-decoder with skip connections [4]

the image is reduced to 128x128 pixels, effectively doubling the number of channels to capture more intricate features. Subsequent encoder blocks further decrease the spatial dimensions, downsampling to 64x64 pixels and 32x32 pixels while enriching the representation with additional channels. This iterative process condenses essential information from the input sketch into a compact latent space, encapsulating the image's salient features efficiently.

Conversely, the decoder section of the generator focuses on reconstructing the output image from the compact latent space representation obtained from the encoder. Each decoder block contributes to the gradual upsampling process, aiming to restore the original dimensions and enhance the reconstructed image's fidelity. For example, if the latent space representation has dimensions of 4x4 pixels with an increased number of channels, the decoder starts by applying transposed convolutional layers to upscale it to 8x8 pixels, progressively expanding the spatial dimensions. Batch normalization and ReLU activation are then applied to refine the feature maps, ensuring the faithful reconstruction of the output image. Dropout may be incorporated in specific decoder blocks to prevent overfitting and maintain the reconstruction's accuracy.

If $\texttt{CE}_k$ represents an encoder block composed of Conv2d-BatchNorm-LeakyReLU layers with k channels in the output layer, then the encoder block of our generator is structured as follows:

$$\texttt{CE}_{64} - \texttt{CE}_{128} - \texttt{CE}_{256} - \texttt{CE}_{512} - \texttt{CE}_{512} - \texttt{CE}_{512} - \texttt{CE}_{512} - \texttt{CE}_{512}$$

It's worth noting that $\texttt{CE}_{64}$ doesn't undergo batch normalization, and the last $\texttt{CE}_{512}$ doesn't undergo batch normalization and utilizes ReLU instead of LeakyReLU.

Similarly, if $\texttt{CD}_k$ represents a decoder block comprised of ConvTranspose2d-BatchNorm-Dropout-ReLU layers with k channels in the output layer, then the decoder block of our generator is structured as follows:

$$\texttt{CD}_{512} - \texttt{CD}_{1024} - \texttt{CD}_{1024} - \texttt{CD}_{1024} - \texttt{CD}_{512} - \texttt{CD}_{256} - \texttt{CD}_{128} - \texttt{CD}_{3}$$

The rationale behind the doubling of feature sizes or channels in the decoder is due to the use of skip connections in U-Net, where the output of the i^{th} layer (encoder) is mapped to the $(n-i)^{th}$ layer (decoder). It's important to note that only the first three decoder blocks employ dropout, and $\texttt{CD}_3$ employs Tanh as an activation function without batch normalization. This final layer produces the colored image with 3 channels.

In the encoder, each layer $\texttt{CE}_k$ reduces the dimension size by half, starting from 256×256 and ending at 1×1, through 8 encoders. Conversely, in the decoder, the dimension of the image is doubled from 1×1 to 256×256, through 8 decoders block $\texttt{CD}_k$.

3.1.2 Discriminator Architecture

The pix2pix discriminator relies on a convolutional neural network (CNN) structure, ideally suited for tasks involving image classification. Unlike conventional discriminators within GANs, the pix2pix discriminator generates a spatial map of classification scores rather than a single value, enabling discrimination at the pixel level.

A notable feature of the pix2pix discriminator is its adoption of the PatchGAN architecture, which concentrates on local image patches, emphasizing high-frequency details rather than the entire image. This approach furnishes nuanced feedback to the generator, facilitating the synthesis of realistic textures and structures. The PatchGAN architecture, depicted in Figure 3, simplifies computation and parameterization, rendering it applicable to images of varying sizes.

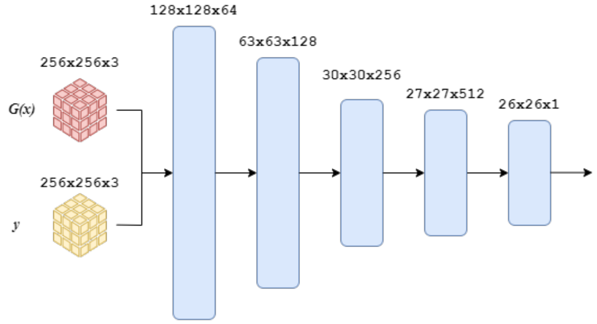

Fig. 3. PatchGAN used as a Discriminator in our implementation of pix2pix

During training, the discriminator is tasked with minimizing the binary cross-entropy loss between predicted classifications and the true labels. The PatchGAN architecture confers several advantages, including reduced parameterization, faster computation, and scalability to accommodate large images. By enforcing high-frequency correctness and capturing local structural intricacies, this discriminator architecture synergizes with GAN training, thereby enhancing the quality of generated images [4].

It comprises multiple layers of convolutional operations followed by optional batch normalization and LeakyReLU activation. If $\texttt{D}_k$ represents an block composed of Conv2d-BatchNorm-LeakyReLU layers with k channels in the output layer, then the block of our discriminator is structured as follows:

$$\texttt{D}_{64} - \texttt{D}_{128} - \texttt{D}_{256} - \texttt{D}_{512} - \texttt{D}_{1}$$

It's worth noting that $\texttt{D}_{64}$ doesn't undergo batch normalization, and the last $\texttt{D}_1$ doesn't undergo batch normalization and LeakyReLU. Dimension of output of each block is shown in Figure 3.

3.2 Loss Function

We have employed the same loss function used in the Pix2Pix framework [4] which is a combination of adversarial loss and additional constraints.

In Pix2Pix, the loss function $L_{cGAN}(G, D)$ is structured as follows:

$$\begin{aligned} L_{cGAN}(G, D) = {} & E_{x,y}[\log(D(x, y))] \\ & + E_{x,z}[\log(1 - D(x, G(x, z)))] \end{aligned} \quad (1)$$

where x is input sketch image, y is output colored image, z is a random noise, and G tries to minimize this objective against an adversarial D that tries to maximize it, i.e.

$$G^* = \arg\min_G \max_D L_{cGAN}(G, D). \quad (2)$$

To investigate the significance of conditioning the discriminator, an unconditional variant is compared where the discriminator does not have access to x. The loss function for this variant, $L_{GAN}(G, D)$, is defined as:

$$\begin{aligned} L_{GAN}(G, D) = &E_y[\log(D(y))] \\ &+ E_{x,z}[\log(1 - D(G(x, z)))] \end{aligned} \tag{3}$$

Previous methodologies have combined the GAN objective with traditional loss functions such as L2 distance [21]. Although the discriminator's role remains consistent, the generator is tasked not only to deceive the discriminator but also to produce outputs close to the ground truth in terms of L1 distance:

$$L_{L1}(G) = E_{x,y,z}[\|y - G(x, z)\|_1] \tag{4}$$

The ultimate objective is to minimize the combined loss:

$$G^* = \arg \min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G) \tag{5}$$

where λ is hyperparameter which controls the relative contribution of the L1 loss term compared to the adversarial loss term in the overall optimization process.

4 Experiments

In this section, we detail the experiments conducted to evaluate the performance of the Pix2Pix model for anime sketch colorization. We describe the dataset used, experimental setup, evaluation metrics, and present results obtained from the experiments.

4.1 Experimental Setup

The experimental setup utilized Kaggle's cloud infrastructure, harnessing the computational capabilities of an NVIDIA Tesla P100 GPU for accelerated training. PyTorch's CUDA backend ensured efficient parallel processing within the GPU-accelerated computing environment. The Pix2Pix model was implemented using the PyTorch framework.

A learning rate of 2e-4 was carefully selected to strike a balance between convergence speed and stability. The batch size was adjusted to 16 to maximize computational resource utilization while maintaining stable training dynamics. Over 150 epochs, the model underwent extensive iteration to glean comprehensive insights from the dataset. Additionally, the L1 lambda parameter, crucial for regulating the significance of L1 loss in the generator's loss function, was set to 100 to achieve a balance between image fidelity and adversarial realism during training.

Both the generator and discriminator were optimized using the Adam optimizer. The selection of appropriate loss functions played a pivotal role in effectively guiding the training process. The discriminator employed binary cross-entropy (BCE) loss, while the generator utilized mean absolute error (L1) loss. These meticulously crafted experimental settings provided a robust foundation for conducting rigorous evaluations and achieving promising performance outcomes.

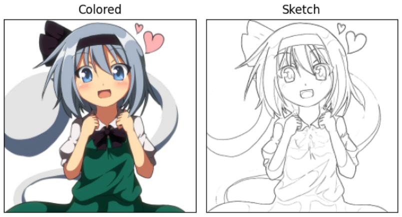

Fig. 4. Example of dataset: Colored/target in left and sketch/input in right

4.2 Dataset

The dataset utilized for the task of anime sketch colorization was obtained from Kaggle under the name Anime Sketch Colorization Pair [7]. This dataset comprises a training set containing $14,224$ pairs of images and a validation set containing $3,545$ pairs. Each image in the dataset has a resolution of 1024×1024 pixels and contains both a colored version and a sketch, with the colored version on the left and the sketch on the right.

Images were separated into colored and sketch components using PyTorch. A custom dataset derived from PyTorch's dataset was created, allowing for the division of each image pair into two halves, one containing the colored image and the other the sketch. In our application, colored image is a target image and the sketch is an input image.

Data augmentation techniques were applied to ensure a diverse training dataset. Following the splitting process, the resulting input and target images were resized from 512×512 pixels to 256×256 pixels. This resolution was deemed sufficient to preserve image details while conserving memory due to the reduced size. Splitted images, input (sketch) and target (colored) images, are shown in Figure 4.

Initially, we randomly flipped the input images horizontally with a probability of 0.5 to further augment the dataset. Additionally, color jittering was applied with a probability of 0.2. Color jittering is a technique that randomizes brightness, contrast, saturation, and hue values to introduce variability into the dataset. However, this data augmentation did not yield better results, so it was subsequently removed.

Finally, normalization was applied to both the input and target images. Normalization is a standard preprocessing step in machine learning tasks that scales the pixel values to a common range, typically [-1, 1], to facilitate more effective learning by the model. By normalizing the images, we aim to enhance the convergence and stability of the training process.

4.3 Evaluation Metrics

To evaluate the performance of the colorization model, we employed the following evaluation metrics:

4.3.1 Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) [9] serves as a metric to assess the likeness in structure between real and generated images. It quantifies similarity based on luminance, contrast, and structure. The SSIM score is calculated using the formula:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6}$$

Here, x and y represent the original and generated images respectively, while μ_x and μ_y denote their mean intensities. Similarly, σ_x and σ_y stand for their standard deviations, and σ_{xy} represents the covariance between x and y. The constants c_1 and c_2 are introduced to prevent division by zero.

The SSIM calculation encompasses both global and local similarities between images, providing a comprehensive measure of their structural correspondence. The SSIM score ranges from -1 to 1, where a score of 1 signifies perfect resemblance, 0 indicates no similarity, and negative values imply significant dissimilarities. Higher SSIM values closer to 1 suggest that the generated image closely matches the original in terms of luminance, contrast, and structure. Conversely, lower scores closer to 0 or negative values highlight substantial differences, revealing flaws in the generated image's quality compared to the original.

4.3.2 Fréchet Inception Distance (FID)

The Fréchet Inception Distance (FID) score [8] serves as a metric for evaluating the quality of generated images. FID evaluates the distance between the feature representations of real and generated images, utilizing a pre-trained Inception-v3 model to extract features. The FID score is calculated based on the mean and covariance of these feature representations.

$$FID = \|\mu_1 - \mu_2\|_2^2 + \text{Tr}\left(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}\right) \tag{7}$$

Here, μ_1 and μ_2 represent the mean vectors of the feature representations of real and generated images respectively, while Σ_1 and Σ_2 denote the covariance matrices of these feature representations.

The FID score complements other evaluation metrics such as SSIM by providing insights into the perceptual quality of generated images from a different perspective. By quantifying the distance between feature distributions, FID offers valuable information regarding the similarity of generated images to real-world data. A lower FID score indicates that the generated images closely resemble the real images in terms of their features.

4.3.3 Peak Signal-to-Noise Ratio (PSNR)

The Peak Signal-to-Noise Ratio (PSNR) [10] measures the quality of an image reconstruction by comparing it to a reference image. PSNR is expressed in decibels (dB), and a higher PSNR value indicates better image quality, with infinity representing perfect similarity between the images. PSNR is calculated using the formula:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \tag{8}$$

Where:

- MAX is the maximum possible pixel value (typically 255 for images with 8-bit depth).
- MSE is the Mean Squared Error between the original and generated images, calculated as:

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j) - K(i,j))^2 \tag{9}$$

Here, I represents the original image, K represents the generated image, and $m \times n$ is the dimensions of the images.

4.4 Results

After training the Pix2Pix model for 150 epochs, we monitored three key evaluation metrics mentioned above — PSNR, FID, and SSIM — to evaluate the quality of the generated colorized images. The mean PSNR, FID, and SSIM scores are illustrated in Figure 5.

Mean PSNR: The plot initiates at approximately 13.36 dB, indicating a moderate PSNR score. There is a noticeable decline shortly after, reaching around 8.71 dB at 30 epochs, suggesting lower image quality compared to the target. Subsequently, there is a recovery, with PSNR scores fluctuating between approximately 13 and 16 dB. The highest peaks, surpassing 16.11 dB, denote instances where the reconstructed image quality closely aligns with the target. Overall, there is a progressive improvement in PSNR as the number of epochs increases, indicating enhanced quality in image reconstruction over time.

Mean SSIM: The mean SSIM scores initially register high around 20, fluctuating over epochs. The peak SSIM score of 0.80 is attained at 110 epochs.

FID: Unlike the other metrics, the FID score plot exhibits distinct behavior. Beginning at a high score exceeding 111.87, it undergoes a sharp decline to around 70.54 at 40 epochs, signifying a considerable improvement. Throughout training, the FID score fluctuates dramatically, indicating inconsistent image quality across epochs. Lower FID scores are preferable, with the lowest score observed below 38.58 at 150 epochs, suggesting the closest similarity to the target image distribution at this stage.

Figure 6 illustrates the progression of colorized images at different epochs. Notably, images at 100 or 140 epochs closely resemble the ground truth, consistent with the graphical analysis. Furthermore, observed performance drops suggest background interference affecting evaluation due to background pixel consideration, epoch 120 and 130 for instance.

4.5 Discussion

The training process exhibits significant instability, characterized by varying metrics across epochs. This variability may stem from factors like fluctuations in learning rate, batch size, the inherent stochasticity of training, or the model's struggle to consistently grasp target distribution features.

Despite these fluctuations, a discernible trend towards improvement emerges, particularly evident in decreasing FID scores suggests successful minimization of the gap between real and generated image distributions, thereby enhancing the realism of the generated images. Similarly, the increasing trend in PSNR values signifies improved pixel-wise fidelity compared to the originals.

However, it's crucial to acknowledge the fluctuating SSIM scores, which lack a clear trend, indicating challenges in maintaining consistent structural similarity across epochs. This inconsistency may arise from variations in

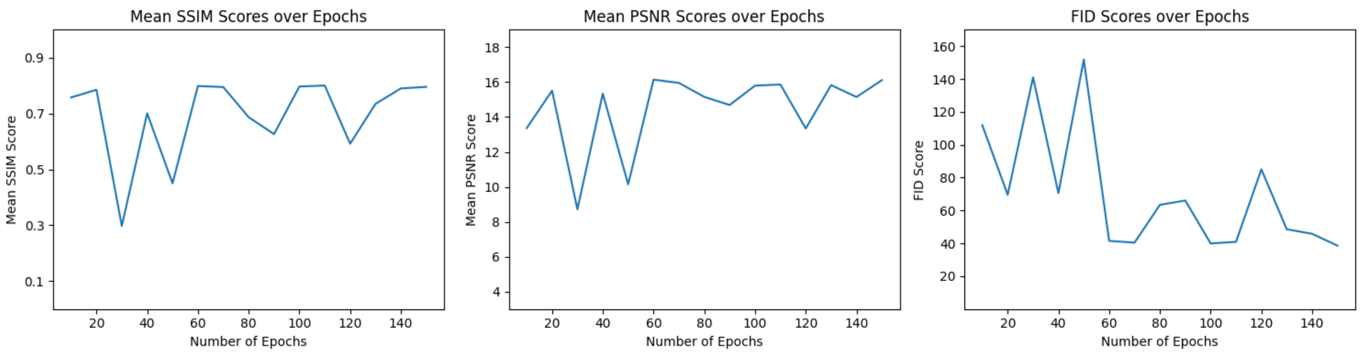

Fig. 5. Graphical representation of evaluation metrics (SSIM, PSNR and FID) over the epochs

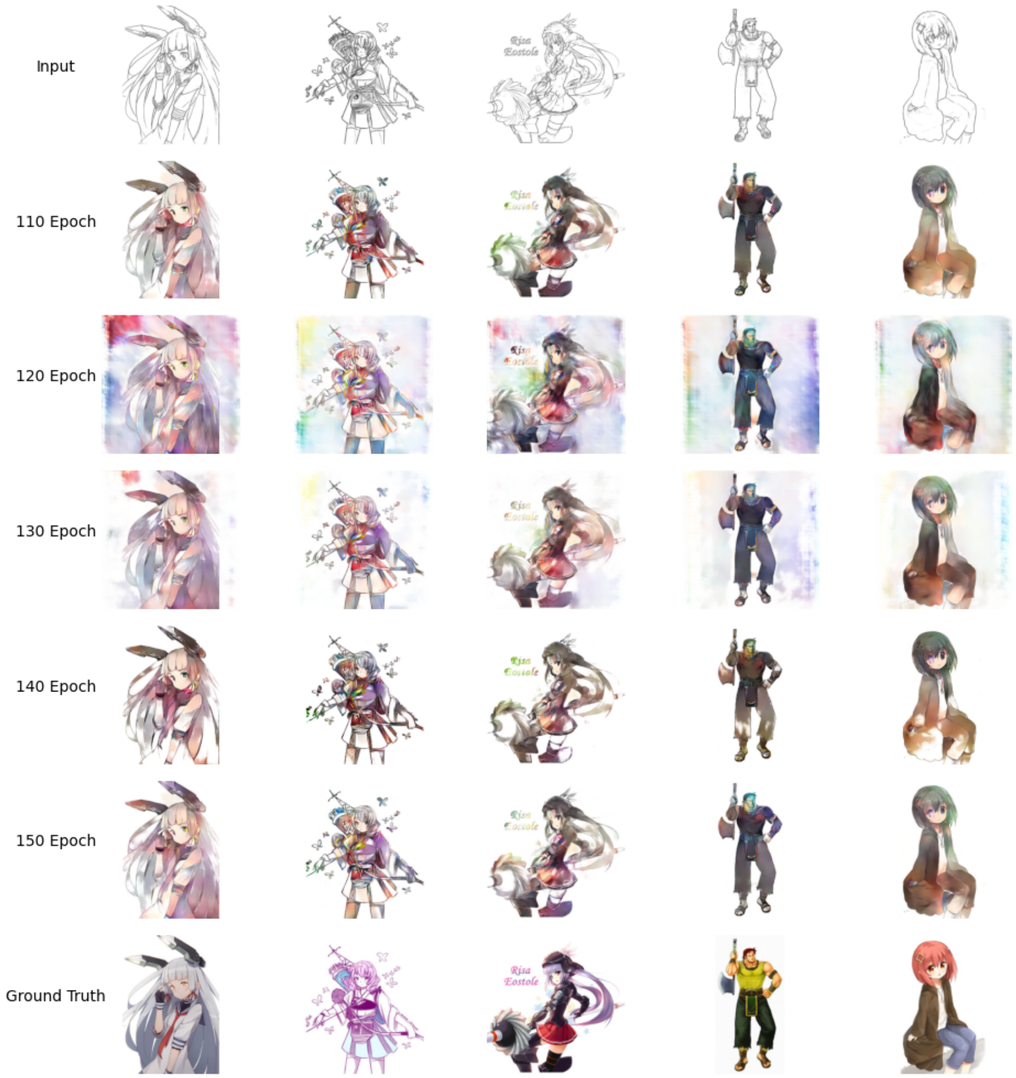

Fig. 6. Progression of predicted images over the epochs.

input sketches and the model's handling of diverse image characteristics.

Future research could concentrate on refining model architecture, exploring alternative loss functions, or incorporating additional regularization techniques to further boost Pix2Pix model performance and stability for anime sketch colorization.

5 Conclusion

In conclusion, this study has explored the application of Pix2Pix, for anime sketch colorization. Through a thorough review of related works and an in-depth examination of

Pix2Pix's architecture and methodology, we have demonstrated its efficacy in automating and enhancing the process of generating colorized anime images from sketches. Our experiments, utilizing a dataset of paired anime sketch and color images, have showcased progressive improvements in image quality over training epochs.

Despite encountering challenges such as training instability and fluctuating SSIM scores, our results highlight the promising potential of Pix2Pix in addressing the task of anime sketch colorization. The observed enhancements in FID scores indicate the model's ability to generate colorized images that closely resemble ground truth data. Furthermore, the utilization of evaluation metrics such as SSIM, FID, and PSNR has provided valuable insights into the performance of the model across different dimensions of image quality.

Looking forward, further research can aim at addressing persisting challenges and exploring avenues for refinement and optimization. Subsequent investigations could concentrate on mitigating training instability, enhancing consistency in Structural Similarity Index (SSIM) scores, and broadening the dataset to encompass a wider array of anime styles and characteristics. Furthermore, efforts could be directed towards improving the quality and accuracy of color application by the model to align more closely with the ground truth colors.

In summary, this study which utilizes the Pix2Pix model, presents a promising pathway for automating and refining the process of creating colorized anime images, thus enhancing the visual narrative experience for both enthusiasts and creators.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015. [Online]. Available: http://arxiv.org/abs/1508.06576

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[3] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Y. Fujimoto, M. Tan, N. Okazaki, and Y. Sato, "Animegan: A novel lightweight gan for photo-to-anime translation," *arXiv preprint arXiv:2006.06666*, 2018.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[7] T. Kim, "Anime sketch colorization pair," Dec 2018. [Online]. Available: https://www.kaggle.com/datasets/ktaebum/anime-sketch-colorization-pair

[8] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017. [Online]. Available: http://arxiv.org/abs/1706.08500

[9] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[10] H. Alqahtani, M. Kavakli-Thorne, G. Kumar, and F. SBSSTC, "An analysis of evaluation metrics of gans," in *International Conference on Information Technology and Applications (ICITA)*, vol. 7, 2019, p. 2.

[11] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *CoRR*, vol. abs/1603.08511, 2016. [Online]. Available: http://arxiv.org/abs/1603.08511

[12] K. Nazeri and E. Ng, "Image colorization with generative adversarial networks," *CoRR*, vol. abs/1803.05400, 2018. [Online]. Available: http://arxiv.org/abs/1803.05400

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] B. Li, Y. Lu, W. Pang, and H. Xu, "Image colorization using cyclegan with semantic and spatial rationality," *Multimedia Tools and Applications*, vol. 82, no. 14, p. 21641–21655, Feb 2023.

[15] P. Hensman and K. Aizawa, "cgan-based manga colorization using a single training image," 2017.

[16] Y. Cao, H. Tian, and P. Y. Mok, "Attention-aware anime line drawing colorization," 2023.

[17] T. Vu and R. Yang, "Ganime: Generating anime and manga character drawings from sketches with deep learning." [Online]. Available: http://cs230.stanford.edu/projects_winter_2020/reports/32226261.pdf

[18] G. Bhullar, N. Thangavelu, and R. Marudhachalam, "Coloring manga images with deep learning: A comparative study."

[19] R. Gao, L. Jie, and K. T. U, "Complex manga coloring method based on improved pix2pix model," in *2023 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2023, pp. 582–587.

[20] M. Shi and T. U. Kin, "Color-comics-image sketch-style transformation based on conditional generative adversarial network," in *2020 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, 2020, pp. 110–115.

[21] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *CoRR*, vol. abs/1604.07379, 2016. [Online]. Available: http://arxiv.org/abs/1604.07379