Reality Sketch: Edge-Aware Image Stylization for Cartoon-like Rendering

Muhammet Batuhan Doğan
Hacettepe University
Department of Artificial Intelligence Engineering
Ankara, Turkey
doganmuhammetbatuhan@gmail.com

Alper Mert
Hacettepe University
Department of Artificial Intelligence Engineering
Ankara, Turkey
alperm76@gmail.com

ABSTRACT

In this project, we present a comparative evaluation of traditional image filtering techniques and generative adversarial network (GAN)-based models for the task of image cartoonization. While classical methods such as bilateral filtering and guided filtering offer efficient and interpretable pipelines, GAN-based models like White-box CartoonGAN and AnimeGANv2 provide data-driven stylization with high perceptual quality. Our dataset consists of 40 images collected from various public code repositories focused on image stylization and cartoonization, including sources such as [8, 10, 12, 15, 16]. We evaluate each method with both qualitative visual inspection (performed by the authors) and quantitative metrics including PSNR, SSIM, edge density, and inference time. Furthermore, we explore fusion-based approaches that combine traditional preprocessing with GAN-based stylization to enhance overall performance. Our results reveal the strengths and limitations of each approach and highlight potential directions for future hybrid cartoonization systems.

1 INTRODUCTION

The transformation of real-world photographs into stylized, artistic renderings, particularly those resembling sketches or cartoons, represents a fascinating and active research area within computational photography and computer vision. This project, titled "Reality Sketch: Edge-Aware Image Stylization for Cartoon-like Rendering," focuses on a comparative study of different methodologies capable of achieving such artistic transformations effectively.

The core interest in this problem stems from the diverse set of techniques available, each presenting its own unique characteristics and, crucially, distinct trade-offs between visual appeal, computational load, and user control. As stylized imagery continues to gain popularity in various applications like social media filters, digital art, and game assets, a systematic comparison and understanding of these trade-offs become increasingly important. This project aims to provide such a comparison by exploring two primary paradigms.

The first paradigm involves Traditional Edge-Aware Filtering techniques. This approach leverages established image processing methods, often implemented using libraries such as OpenCV [11], including edge-preserving filters like the Bilateral Filter and the Guided Filter, used alongside edge detection algorithms like Canny. These methods are primarily noted for their computational efficiency and interpretability, serving as a strong baseline for comparison.

Conference'17, July 2017, Washington, DC, USA
2025.

In contrast, the second paradigm explores Deep Learning-Based Techniques. For instance, we utilize the pre-trained White-box Cartoonization model [17, 18]. This model builds upon concepts from earlier works like CartoonGAN [5]. We also employ AnimeGAN, specifically implementations based on AnimeGANv2 [4, 14]. These methods leverage complex neural networks trained to learn intricate mappings between photographs and artistic styles. While often capable of producing highly dynamic and visually compelling results, they typically involve higher computational costs.

Beyond comparing these two main paradigms in isolation, we also investigate the potential of sequentially fused models. This involves using the output of one method as the input for another – for example, applying a Bilateral filter to an image before processing it with AnimeGAN. This allows us to explore whether pre-processing or post-processing can enhance or modify the results in beneficial ways.

The central motivation for this project is, therefore, to conduct a systematic comparison. To achieve this, we employ both objective quantitative metrics (like SSIM, PSNR, Edge Density, and Time) and, crucially, qualitative evaluation to assess aesthetic appeal. Recognizing the practical constraints of conducting formal user studies, the qualitative assessment was performed by the authors. Our approach was guided by principles of subjective image quality assessment, such as Pair Comparison (PC) [1], where we compared the outputs from different methods side-by-side for each original image, focusing on criteria like cartoon-likeness, edge quality, and artifacts. Our key questions remain: How do traditional filtering techniques measure up against pre-trained deep learning models? And can sequentially fused models offer unique advantages?

By implementing, evaluating (both quantitatively and through author-based qualitative comparison), and comparing these distinct strategies, this work seeks to provide valuable insights into the relative strengths and weaknesses of different cartoonization approaches, offering a clearer picture for practitioners choosing tools for stylized image generation.

2 RELATED WORK

The transformation of real-world photographs into cartoon-like styles, often termed image cartoonization or stylization, is a vibrant research area in computational photography and computer vision. Existing approaches generally fall into three main categories: traditional image processing (IP) techniques, deep learning (DL) methods, and hybrid strategies combining elements of both. Our project aims to compare these paradigms, drawing upon a range of relevant studies.

2.1 Traditional Image Processing Approaches

Several methods rely solely on classical image processing pipelines. For instance, Balaji et al. (2024) [2] present an efficient, purely edge-aware filtering approach. Their method performs parallel edge enhancement (detection and dilation) and color quantization on separate channels, subsequently combining them to produce the final cartoon image without using deep networks [2]. This highlights how strong edge outlines and reduced color complexity can be achieved with low computational cost through traditional techniques [2]. Similarly, other studies discussed by Kulkarni and Agrawal [9], as well as Joshitha [7], detail OpenCV-based pipelines using sequences of filters like Bilateral Filtering, edge detection (e.g., Canny), and color simplification steps to achieve stylization. These methods are often valued for their efficiency and interpretability.

2.2 Deep Learning Approaches

The advent of deep learning, particularly Generative Adversarial Networks (GANs), has introduced powerful tools for learning complex artistic styles directly from data. A foundational work in this area is **CartoonGAN by Chen et al. (2018) [5]**. This framework specifically targets the unique characteristics of cartoons—such as simplified textures and clear edges—by employing a GAN trained on unpaired photos and cartoons. It introduced novel loss functions designed to capture these stylistic features effectively, distinguishing it from general-purpose style transfer methods.

Building upon such concepts, Wang and Yu (2020) [18] proposed a "white-box" GAN framework specifically for cartoonization. Their model explicitly learns and disentangles three key cartoon representations (surface shading, color structure, edges/texture) from photographs, allowing for controllable stylization via separate loss functions [18]. This GAN-based approach demonstrated state-of-the-art results compared to previous methods. Other works, as reviewed by Balaji et al. (from the proposal) [3] and Kulkarni & Agrawal [9], also emphasize the capability of GANs like CartoonGAN to produce dynamic and visually appealing cartoon images, though often at the cost of significant computational resources and large datasets.

2.3 Hybrid Approaches

Recognizing the complementary strengths of traditional and DL methods, some research explores hybrid strategies. Raut et al. (2024) [13] propose a method where a GAN (specifically CartoonGAN) first generates an initial cartoon image. This output is then refined using traditional post-processing: K-means clustering segments the image into color regions, which are then stylized (recolored/smoothed) before recombination [13]. They report that this blend of learned generation and edge-aware color simplification yields high-quality results [13]. This aligns with the conclusion from Joshitha's review [7], which also suggests that a hybrid approach combining OpenCV-based filtering/edge detection with deep learning models could potentially achieve optimal outcomes.

2.4 Comparative Context

Broader comparisons, such as the initial study reviewed by Balaji et al. (from the proposal) [3] and the work by Kulkarni and Agrawal [9], explicitly contrast the trade-offs. They reiterate that while traditional methods excel in computational efficiency and preserving certain structural information, deep learning often provides superior artistic fluidity and visual appeal, albeit with challenges related to computational cost, data requirements, and potential overfitting.

This body of work underscores the ongoing exploration of different cartoonization strategies. Our project situates itself within this context by systematically implementing and comparing representative traditional techniques (Bilateral/Guided filters, Canny/Sobel edges) against a prominent deep learning model (CartoonGAN), while also considering the potential insights regarding hybrid approaches suggested by the literature.

3 THE APPROACH

Our comparative study explores two primary paradigms for image cartoonization: traditional image processing techniques and deep learning-based methods. This section details the specific algorithms, their mathematical underpinnings, their parameters, and the pipelines employed within each paradigm.

3.1 Traditional Image Processing Pipelines

We implemented and evaluated two distinct traditional pipelines. These aim to achieve a sketch-like cartoon effect by simplifying color regions and emphasizing edges, primarily using OpenCV functionalities [11].

3.1.1 Bilateral Filter-Based Pipeline. This pipeline represents a well-established approach, utilizing the Bilateral Filter for its edge-preserving smoothing capabilities and the Canny edge detector for outline extraction. The Bilateral Filter computes the output pixel value $BF[I]_p$ at position p as a weighted average of nearby pixels q:

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q$$

Here, W_p is a normalization factor, G_{σ_s} is the spatial Gaussian weight (decreasing with distance $\|p - q\|$), and G_{σ_r} is the range Gaussian weight (decreasing with intensity difference $\|I_p - I_q\|$). This dual weighting allows smoothing while preserving edges where $\|I_p - I_q\|$ is large. Its key parameters are `d` (diameter of the pixel neighborhood), `sigmaColor` (filter sigma in the color space), and `sigmaSpace` (filter sigma in the coordinate space). Increasing `d` considers a wider area for filtering, potentially leading to stronger smoothing but also significantly increasing computation time. A larger `sigmaColor` allows more dissimilar colors to be averaged, resulting in more pronounced smoothing and a more posterized look, but may blur finer edges. A larger `sigmaSpace` means that spatially distant pixels can still influence the result if their colors are also similar, contributing to smoothing over larger regions.

We also use the Canny Edge Detector. Canny involves Gaussian smoothing, gradient computation, non-maximum suppression, and double thresholding with hysteresis, controlled by `low` and `high` thresholds. The `low` threshold determines which weak edges are considered for linking, and the `high` threshold identifies strong edges; adjusting these controls the sensitivity and connectedness of detected outlines. A Median Blur, with kernel size `median_k`, is applied pre-Canny for noise reduction. A larger `median_k` applies

stronger noise reduction but can also blur fine details before edge detection. Such pipelines combining Bilateral filtering and Canny for cartoonization are discussed in works like [9] and [7].

The specific steps for our pipeline are as follows:

(1) Apply `cv2.bilateralFilter`.
   (a) Apply `cv2.bilateralFilter`.
   (b) Convert to grayscale, apply `cv2.medianBlur`, and run `cv2.Canny`.
   (c) Invert and convert the edge map to BGR.
   (d) Combine the smoothed image with the inverted edge mask using the `cv2.bitwise_and` operation.

For this pipeline, we adopted the **v10** parameter set ($d = 5$, $\sigma Color = 100$, $\sigma Space = 100$, $low = 80$, $high = 220$, $median\_k = 5$). This choice was based on our preliminary work involving a comparative analysis of 10 variants, using both subjective visual assessment and quantitative metrics, ultimately selecting v10 for its balanced performance across various images.

3.1.2 Guided Filter and K-Means Based Pipeline. As an alternative, we explored a second traditional pipeline incorporating the Guided Filter and K-Means clustering. The Guided Filter, often offering greater computational efficiency, assumes a local linear model in a window w_k:

$$q_i = a_k I_i + b_k, \forall i \in w_k$$

where q_i is the output, I_i is the guidance image pixel, and a_k, b_k are linear coefficients. These coefficients are found by minimizing an energy function $E(a_k, b_k) = \sum_{i \in w_k}((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2)$, where p_i is the input image and ϵ (`eps`) is a regularization parameter preventing a_k from being too large. The `radius` parameter defines the window size. A larger `radius` expands the local window for the linear model, generally leading to more significant smoothing. A smaller `eps` allows the output to more closely follow the guidance image (less smoothing, more detail preservation), while a larger `eps` enforces a stronger smoothing effect by penalizing variations from a flat response.

For color quantization, we use K-Means Clustering. It aims to partition N pixels (x_i) into K clusters (S_j) by minimizing the within-cluster sum of squares (WCSS):

$$J = \sum_{j=1}^{K} \sum_{x_i \in S_j} \|x_i - c_j\|^2$$

where c_j is the centroid (mean color) of cluster S_j. The value of `k` directly controls the number of distinct colors in the output; a smaller `k` produces a more abstract and heavily simplified color palette, whereas a larger `k` preserves more color nuances.

The pipeline for this alternative approach involves the following steps:

(1) Canny edge processing (using parameters consistent with the v10 set).
(2) Smoothing the original color image with `cv2.ximgproc.guidedFilter`, employing parameters `radius=7` and `eps=500`. These were selected based on common practices and initial visual assessments.
(3) Color reduction via `cv2.kmeans`, using `k=16`. This value was chosen over `k=32` for its preferred aesthetic outcome and faster processing speed.
(4) Combining the K-Means processed image with the prepared edge mask using `cv2.bitwise_and`.

The Guided Filter precedes K-Means in this sequence to provide a smoother input for the clustering stage; exploring the reverse order is considered an avenue for future work.

3.2 Deep Learning Approaches

For deep learning methods, we utilized two pretrained models, leveraging their ability to learn complex stylistic transformations from data. Generative Adversarial Networks (GANs) operate via a two-player min-max game, typically formulated as:

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

where the Generator (G) tries to create realistic fakes, and the Discriminator (D) tries to distinguish them from real data.

3.2.1 White-box CartoonGAN Based Pipeline. White-box CartoonGAN [17, 18] represents a deep learning-based pipeline that applies generative adversarial learning to convert natural photos into stylized cartoon-like images. This model builds upon the traditional CartoonGAN architecture [5] but includes several enhancements to improve visual quality, training stability, and interpretability. The architecture consists of a generator network, which learns to map real-world images to the cartoon domain, and a discriminator network, which evaluates the realism of the stylized outputs. What distinguishes White-box CartoonGAN is its "white-box" design approach, often achieved through structured losses and architectural improvements for greater transparency.

Key innovations contributing to its effectiveness include an edge-promoting adversarial loss ($\mathcal{L}_{edge}$), which encourages the model to preserve and emphasize contour lines critical for a cartoon effect, alongside standard GAN losses. A perceptual content loss ($\mathcal{L}_{content}$), often using VGG-19 features, ensures that the output preserves essential semantic information from the input. Furthermore, instance normalization is typically used instead of batch normalization, aiding style consistency and improving training behaviour. Implementations also often incorporate mechanisms like adaptive layer weight decay to enhance training stability and reduce common artifacts such as color bleeding or broken edges. The inference process is fully feedforward: the input image is passed through the pretrained generator, and the resulting cartoonized image is returned.

3.2.2 AnimeGANv2 Based Pipeline. AnimeGANv2 [4, 14] is another GAN-based model, but it is lightweight, efficient, and specifically designed to translate real-world photos into anime-style illustrations, drawing its style from training on anime images. Unlike White-box CartoonGAN, which aims for a more general cartoon look, AnimeGANv2 focuses on replicating the distinct visual aesthetic of Japanese anime, particularly in backgrounds, shading, and character representation.

Its architecture typically features a ResNet-based generator with residual blocks and upsampling layers for transformation, and a PatchGAN-style discriminator to evaluate local features. The objective function incorporates style loss ($\mathcal{L}_{style}$) and color loss ($\mathcal{L}_{color}$)

terms that explicitly encourage anime-style smoothness and saturation, in addition to adversarial losses. Key characteristics of AnimeGANv2 include its fast inference time due to a relatively shallow and optimized architecture, and its ability to produce stylized textures and color flattening mimicking anime patterns. Its pretraining on large anime datasets ensures that specific stylistic features like thick contours, saturated backgrounds, and simplified shadows are learned. Similar to White-box, inference involves passing an input image through the pretrained generator. However, the output quality is highly stylized and may deviate significantly from the original structure.

3.3 Hybrid Pipelines (Fused Models)

After implementing the individual traditional and deep learning approaches, we explored hybrid pipelines. Our approach involves applying a traditional cartoonization filter as a preprocessing step before feeding the result into a GAN model (either White-box CartoonGAN or AnimeGANv2). This aims to explore potential complementarity between traditional image abstraction and deep generative stylization.

The rationale behind this Traditional-to-DL sequence is twofold. Firstly, we hypothesized that traditional filters could perform noise reduction and structure enhancement, smoothing out fine textures and enhancing boundaries. This preprocessed image might allow the GAN to focus more effectively on high-level stylization. Secondly, we considered style harmonization; since GANs can be sensitive to input characteristics, preprocessing might regularize input statistics and improve the consistency of the stylization.

The fusion process itself is straightforward: an original image is first processed using a chosen traditional method. The resulting cartoonized output is then passed as input to the pretrained GAN model. The GAN further stylizes this filtered input, aiming to combine the structural clarity often associated with traditional filters with the expressive, learned aesthetics of deep models. While this sequential pipeline is inherently more computationally expensive, our initial observations suggest it can produce visually pleasing results with potentially better edge coherence or improved stylization balance in some images. However, its effectiveness can vary depending on the input image's complexity and the specific compatibility between the chosen filter and the GAN model.

4 EXPERIMENTAL RESULTS

In this section, we present a comprehensive evaluation of the cartoonization methods explored in our project. We assess the performance of two traditional image processing pipelines (Bilateral Filter and Guided Filter + K-Means) and two deep learning-based models (White-box CartoonGAN and AnimeGAN), including combinations of these approaches through fusion strategies.

To perform a rigorous comparison, we selected two representative input images from our test set: one portrait photograph captured in an indoor restaurant setting, and one urban outdoor scene containing strong geometrical structures (a bridge architecture). These images were chosen intentionally to highlight the strengths and limitations of each method under different visual characteristics such as human features, texture, lighting conditions, and structural edges. *You can access the results for all images in our dataset from the project's GitHub Repository [6].*

For each method, we compute standard image quality metrics including Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), Edge Density (ED), and Inference Time (sec). PSNR and SSIM measure how structurally similar the stylized image is to the original. SSIM in particular aligns better with human perception. Edge Density evaluates how well edge and line information is preserved or emphasized after stylization. Inference Time assesses how fast each model processes a single image.

In addition to the quantitative metrics, we also provide a qualitative analysis of the outputs by discussing visual properties such as stylization level, color palette transformation, edge retention, and potential distortions. Finally, we address the inherently subjective nature of cartoonization by considering how stylistic preferences and use-case requirements can influence the perceived quality of a result.

Original Input Images

(a) Portrait/Restaurant Scene. **(b) Bridge/Urban Scene.**

Figure 1: The two representative original input images used for evaluation.

4.1 Bilateral Filtering Output

(a) Bilateral: Portrait/Restaurant.

SSIM:	0.77
PSNR:	18.28
Edge Density:	0.08
Time (s):	0.015

(b) Bilateral: Bridge/Urban.

SSIM:	0.79
PSNR:	18.74
Edge Density:	0.07
Time (s):	0.023

Figure 2: Results of the Bilateral Filter pipeline on the two representative images, with corresponding metrics.

4.2 Guided Filtering + K-Means Output

(a) Guided+KMeans: Portrait/Restaurant.

SSIM:	0.67
PSNR:	17.98
Edge Density:	0.10
Time (s):	6.28

(b) Guided+KMeans: Bridge/Urban.

SSIM:	0.73
PSNR:	18.43
Edge Density:	0.08
Time (s):	23.03

Figure 3: Results of the Guided Filter + K-Means (k=16) pipeline on the two representative images, with corresponding metrics.

4.3 White-box CartoonGAN Output

(a) White-box: Portrait/Restaurant.

SSIM:	0.80
PSNR:	21.14
Edge Density:	0.08
Time (s):	6.59

(b) White-box: Bridge/Urban.

SSIM:	0.87
PSNR:	18.93
Edge Density:	0.05
Time (s):	5.93

Figure 4: Results of the White-box CartoonGAN pipeline on the two representative images, with corresponding metrics.

4.4 AnimeGAN Output

(a) AnimeGAN: Portrait/Restaurant.

SSIM:	0.23
PSNR:	10.08
Edge Density:	0.04
Time (s):	1.49

(b) AnimeGAN: Bridge/Urban.

SSIM:	0.38
PSNR:	10.86
Edge Density:	0.03
Time (s):	1.14

Figure 5: Results of the AnimeGAN (v2) pipeline on the two representative images, with corresponding metrics.

4.5 Bilateral+White-box CartoonGAN Fuse Output

(a) Bilateral+White-box: Portrait.

SSIM:	0.73
PSNR:	18.47
Edge Density:	0.07
Time (s):	5.83

(b) Bilateral+White-box: Bridge.

SSIM:	0.82
PSNR:	18.87
Edge Density:	0.05
Time (s):	6.03

Figure 6: Results of the Fused Bilateral Filter + White-box CartoonGAN pipeline, with metrics.

4.6 Bilateral+AnimeGAN Fuse Output

(a) Bilateral+AnimeGAN: Portrait.

SSIM:	0.26
PSNR:	9.91
Edge Density:	0.04
Time (s):	1.24

(b) Bilateral+AnimeGAN: Bridge.

SSIM:	0.44
PSNR:	10.61
Edge Density:	0.02
Time (s):	1.20

Figure 7: Results of the Fused Bilateral Filter + AnimeGAN pipeline, with corresponding metrics.

4.7 Guided+White-box CartoonGAN Fuse Output

(a) Guided+White-box: Portrait.

SSIM:	0.73
PSNR:	18.47
Edge Density:	0.06
Time (s):	6.11

(b) Guided+White-box: Bridge.

SSIM:	0.79
PSNR:	18.83
Edge Density:	0.05
Time (s):	5.98

Figure 8: Results of the Fused Guided Filter + K-Means + White-box CartoonGAN pipeline, with metrics.

4.8 Guided+AnimeGAN Fuse Output

(a) Guided+AnimeGAN: Portrait.

SSIM:	0.27
PSNR:	10.01
Edge Density:	0.04
Time (s):	1.15

(b) Guided+AnimeGAN: Bridge.

SSIM:	0.43
PSNR:	10.66
Edge Density:	0.02
Time (s):	1.28

Figure 9: Results of the Fused Guided Filter + K-Means + AnimeGAN pipeline, with corresponding metrics.

4.9 Quantitative Metric Analysis

For the first image (human scene/indoor portrait), White-box CartoonGAN yielded the highest SSIM (0.80) and PSNR (21.14), indicating that it best preserves structural and pixel-level fidelity. It maintains a moderate edge density (0.08), meaning it balances abstraction and realism, though at the expense of relatively high inference time (6.59s). Bilateral Filtering provided the second-best SSIM (0.77), but slightly lower PSNR (18.28). It also had the fastest processing time (0.015s), making it the most efficient traditional filter. The edge density (0.08) is comparable to White-box, suggesting

similarly effective edge-aware smoothing. Guided Filter + K-Means (referred to as Guided Filter in the provided text) performed slightly worse than Bilateral in terms of SSIM (0.67) and PSNR (17.98), with the highest edge density (0.10) among all methods. This suggests stronger edge retention but less fidelity. Its execution time (6.28s) is notably higher than Bilateral due to the guided filtering and K-Means clustering operations. AnimeGAN scored significantly lower on SSIM (0.23) and PSNR (10.08), indicating poor pixel fidelity. However, it produced outputs with the lowest edge density (0.04) and decent computational speed (1.49s), showing stronger stylization at the cost of accuracy. Its output has much more difference compared to other methods, since it was trained with only anime type images. Bilateral+White-box and Guided+Whitebox fusion results reduced the SSIM (0.73) and PSNR (~18.46) compared to pure White-box, while slightly lowering edge density and inference time. This implies that pre-processing by traditional filters smooths out some structures, slightly degrading fidelity while offering minor speedups and edge regularization. Bilateral or Guided Filter preprocess did not affect the White-box CartoonGAN output very much. Fused AnimeGAN results (both Bilateral and Guided) did not offer noticeable improvements over original AnimeGAN, with still low SSIM (~0.26–0.27) and PSNR (~10.00). Their inference time was slightly faster, and edge densities remained low (~0.04).

For the second image (bridge scene/urban architecture), White-box again achieved the highest SSIM (0.87) and PSNR (18.93), showing consistent strength in preserving detail even in structured outdoor scenes. Bilateral was second in SSIM (0.79), with slightly lower PSNR (18.74) and the fastest runtime (0.023s), proving its effectiveness in simple smoothing tasks. The Guided Filter + K-Means pipeline had decent SSIM (0.73) and PSNR (18.43), but its runtime was the longest (23.03s), significantly limiting its practical usability. AnimeGAN continued to lag behind in SSIM (0.38) and PSNR (10.86), confirming that it favors strong stylization over structural accuracy. Bilateral+White-box retained strong scores (SSIM: 0.82, PSNR: 18.87) close to original White-box with a negligible drop. Edge density and runtime were also consistent. Guided+White-box performed slightly worse than bilateral fusion in terms of SSIM (0.79), but similar PSNR (18.83). Fused AnimeGAN outputs remained far behind (SSIM ~0.43–0.44, PSNR ~10.61–10.66), with edge densities further reduced (~0.02), confirming aggressive abstraction.

To sum up the evaluation metric results, White-box CartoonGAN consistently achieved the highest fidelity (SSIM, PSNR) while maintaining moderate edge stylization. Bilateral filtering was efficient and effective as a traditional baseline. The Guided filtering pipeline had a heavier computational cost with limited fidelity gains. AnimeGAN provided strong stylization with lower structural accuracy. Fusion with traditional filters marginally impacted performance: it was slightly beneficial for White-box, but not meaningful for AnimeGAN.

4.10 Qualitative Analysis

To complement the quantitative evaluation, we conducted a detailed qualitative analysis on the two representative images: a human-centered indoor scene (Figure 1a) and an outdoor architectural photograph (Figure 1b). Each input was processed with all eight pipelines—including traditional filters, two GAN-based models, and four fusion variations—to evaluate subjective quality across several visual dimensions, as performed by the authors. The evaluation criteria for qualitative analysis are:

- Color Saturation & Tonal Palette
- Edge Preservation & Sharpness
- Style Consistency
- Surface Smoothness
- Realism vs. Stylization Trade-off
- Aesthetic Impression

4.10.1 Qualitative Analysis for the Human Image. **Color Saturation:** Among all models, White-box exhibits the most vibrant yet realistic color tones. The blonde hair, red dress, and wooden textures in the background remain expressive without being exaggerated. In contrast, AnimeGAN introduces a slightly pale and unnatural tone, which flattens the overall perception of the scene. Guided filter outputs tend to desaturate slightly due to edge emphasis, while bilateral-based outputs maintain natural saturation.

Edge Sharpness: Bilateral-based pipelines, particularly Bilateral+Whitebox, excel in retaining edge sharpness without artifacts. The edges around the subject's face and the glass are crisp. Guided filter outputs, despite having higher edge density, sometimes suffer from noise or broken lines. AnimeGAN consistently produces the smoothest outputs with minimal edges, resulting in a loss of structure in detailed areas like the background shelves.

Style Consistency: White-box model provides a stylistically coherent cartoonization across different image areas. The textures on skin, hair, and food are rendered with uniform abstraction. AnimeGAN, while stylistically pleasing in smooth regions, often fails to maintain consistency in textured zones like hair and background. The Guided+Whitebox combination somewhat over-flattens these textures, losing the sense of depth.

Surface Smoothness: AnimeGAN stands out for generating the smoothest surfaces, which is ideal for applications targeting minimalistic cartoon aesthetics. However, it sometimes leads to oversmoothing in facial features. White-box provides a balanced approach—sufficient smoothness while preserving critical structure. Guided outputs occasionally introduce blotchy textures.

Realism vs Stylization: Bilateral and White-box models best preserve the recognizable structure of the scene. AnimeGAN sacrifices realism for stylization, leading to less interpretability in complex scenes. Guided+AnimeGAN is the most abstract variant, visually deviating from the original intent.

Aesthetic Impression: White-box outputs often align best with Western cartoon styles, offering sharp details and vivid coloration. AnimeGAN leans more towards Japanese anime aesthetics with soft tones. The fused methods (Bilateral+Whitebox) appear to blend the strengths of both worlds, offering a compelling aesthetic for commercial illustration.

4.10.2 Qualitative Analysis for the Bridge Image. **Color Saturation:** The bridge image reveals more pronounced differences. The Bilateral and White-box fusion (Bilateral+White-box) maintains a rich palette—particularly in sky blues and stone grays—without oversaturation. Guided-based outputs tend to slightly wash out the scene. AnimeGAN strongly simplifies color variation, flattening the sky and removing subtle gradients. **Edge Sharpness:** Bilateral

methods preserve architectural outlines with high fidelity, making them suitable for structured scenes. White-box output is similarly clear but slightly softer at roof boundaries. Guided+AnimeGAN again underperforms, often losing the distinct borders of towers.

Style Consistency: White-box once again achieves stylistic uniformity across sky, water, and structural elements. In contrast, AnimeGAN introduces inconsistent smoothness levels—over-flattening water while preserving too much in towers.

Surface Smoothness: AnimeGAN again produces the smoothest results, but at the cost of fine texture. Bilateral+White-box is a good compromise, offering smooth but not plasticky surfaces. Guided outputs exhibit minor noise artifacts.

Realism vs Abstraction: For architectural scenes, realism is crucial. White-box and bilateral outputs maintain structural integrity while stylizing. AnimeGAN overly abstracts, making the bridge less legible. Guided+White-box provides a stylized but still recognizable result.

Aesthetic Impression: The aesthetic preference may vary by context. Bilateral+White-box emerges as the best balance between technical fidelity and stylization. AnimeGAN may suit stylized animation or game use, while Guided+AnimeGAN likely fits niche artistic applications.

4.11 Discussion on Subjectivity

Despite the availability of objective image quality metrics such as SSIM, PSNR, and Edge Density, cartoonization is inherently a subjective task. These metrics provide helpful insights into technical fidelity and structural preservation, but they fall short in capturing aesthetic appeal, artistic intention, or viewer preference—all of which are crucial in applications such as digital illustration, animation, or stylized photography. For instance, although AnimeGAN consistently scores lower on SSIM and PSNR, it may still be the preferred output in contexts where a soft, minimalist, anime-style visual is desired. On the other hand, White-box outputs may appeal to audiences seeking a detailed, edge-aware cartoon aesthetic more akin to Western animation styles. Furthermore, user preference can vary not only based on cultural or artistic expectations but also depending on the content of the image. In our experiments, the human portrait image benefited from White-box's balanced abstraction and preserved facial features, while the bridge image showed the strengths of bilateral filtering in architectural clarity. In scenarios involving storytelling or visual branding, subjective choices might prioritize emotional tone or visual impact over technical precision. Thus, there is no "best" output in a universal sense. The definition of quality in cartoonization is heavily guided by the use-case, audience, and contextual intent. Future work may benefit from incorporating user studies or preference-based metrics to more accurately evaluate and tailor cartoonization methods for their intended application domains.

5 CONCLUSIONS

In this project, we explored and evaluated multiple image cartoonization approaches by leveraging both traditional edge-aware filtering techniques and modern deep learning-based generative models. Our experimental pipeline incorporated Bilateral and Guided Filters as classical, edge-preserving methods, and compared them against state-of-the-art GAN-based solutions, specifically White-box CartoonGAN and AnimeGAN. Additionally, we investigated hybrid pipelines combining traditional filtering and GANs to assess whether fusing low-level structural preservation with learned stylistic abstraction could enhance the final visual results.

Through comprehensive experimentation across diverse images, we analyzed both quantitative metrics (SSIM, PSNR, Edge Density, and runtime) and qualitative features such as color flatness, edge sharpness, depth preservation, and visual coherence. Our findings indicate that while White-box consistently achieved the highest structural and perceptual metrics, it came at the cost of significant processing time. AnimeGAN, on the other hand, was computationally lighter but offered weaker structural fidelity and edge retention. Fusion pipelines showed moderate success by attempting to balance the strengths of each component. Although they did not outperform the best individual models quantitatively, they offered stylistic diversity and provided more controllable visual outcomes in some scenarios.

Ultimately, we conclude that there is no universally superior cartoonization model. The optimal choice depends on application-specific constraints and aesthetic preferences. Our results support the notion that image cartoonization is a fundamentally subjective process, where user intent, context, and artistic goals play critical roles in defining what constitutes a "good" cartoonized image.

Limitations

While our study explores a diverse set of cartoonization techniques, ranging from traditional edge-aware filtering to deep learning-based GAN models, there are several limitations to acknowledge. First, the use of pretrained models (White-box and AnimeGAN) restricts our ability to fine-tune outputs for specific visual preferences or target datasets. These models are trained on general-purpose datasets and may not generalize well to specialized domains such as medical imaging, architectural sketches, or animated storytelling. Second, our evaluation relies predominantly on low-level metrics like SSIM, PSNR, and Edge Density, which, while useful, do not fully capture subjective and aesthetic aspects of visual quality. Although we have included a qualitative analysis conducted by the authors, the lack of formal user studies limits our ability to assess human perception more broadly and objectively. Additionally, processing time varied significantly between pipelines. The Guided Filter + K-Means pipeline, for instance, was computationally more expensive than the Bilateral filter pipeline, and the White-box model incurred high latency due to its architectural complexity.

Future Directions

To address these limitations, future work could explore fine-tuning or training cartoonization models on domain-specific datasets, allowing for better adaptability and visual coherence in targeted applications. Incorporating perceptual metrics or user preference studies could also provide more robust evaluations, especially for artistic tasks. Moreover, future research could investigate hybrid architectures that combine traditional filtering with neural stylization more seamlessly, possibly allowing real-time performance without compromising quality. Exploring style-transfer-based methods with controllable parameters might also give end-users more agency in

guiding the cartoonization style. This suggests that future solutions should focus on user-adaptive models, interactive stylization frameworks, and deeper integration of perceptual evaluation, building upon the subjective nature of defining a "good" cartoonized image highlighted in our findings.

REFERENCES

[1] AIN434/BBM444 Mehmet Erkut Erdem. 2025. Lecture 11: Visual Quality Assessment, Slide 10. (2025). AIN434/BBM444 Computational Photography Course Materials. (Accessed: May 28, 2025).

[2] A. Balaji, Kota Deepak Venkatesh, Shaik Mohammad Anwar, Shaik Shabana, and Mangamuri Venkata Mohan. 2024. An Efficient Image Processing Based Image-to-Cartoon Generation Based on Deep Learning. Source mentioned as turcomat.org / ResearchGate. https://www.researchgate.net/publication/379646039 Focuses on purely edge-aware filtering approach. Accessed: 2025-04-26.

[3] A. Balaji, Kota Deepak Venkatesh, Shaik Mohammad Anwar, Shaik Shabana, and Mangamuri Venkata Mohan. 2024. AN EFFICIENT IMAGE PROCESSING BASED IMAGE TO CARTOON GENERATION BASED ON DEEP LEARNING. ResearchGate. https://www.researchgate.net/publication/379646039_AN_EFFICIENT_IMAGE_PROCESSING_BASED_IMAGE_TO_CARTOON_GENERATION_BASED_ON_DEEP_LEARNING Accessed: 2025-04-26.

[4] bryandlee and contributors. 2021. animegan2-pytorch: PyTorch implementation of AnimeGANv2. https://github.com/bryandlee/animegan2-pytorch. Accessed: May 28, 2025.

[5] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. 2018. CartoonGAN: Generative Adversarial Networks for Photo Cartoonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9465–9474.

[6] Doğan, Muhammet Batuhan and Mert, Alper. 2025. Reality Sketch: Project Repository. https://github.com/AlperMRT/Image_Cartoonization. Accessed: May 28, 2025.

[7] Madugula Sai Joshitha. 2023. IMAGE TO CARTOON GENERATION USING MACHINE LEARNING AND OPENCV. International Journal For Relevant Progress Research (IJRPR), Volume 3, Issue 11. https://ijrpr.com/uploads/V3ISSUE11/IJRPR7865.pdf Accessed: 2025-04-26.

[8] juanjaho and contributors. [n. d.]. AnimeArcaneGAN_Mobile Repository (Dataset Source). https://github.com/juanjaho/AnimeArcaneGAN_Mobile. Accessed: May 28, 2025. Used as a source for dataset images..

[9] Srushti Anil Kulkarni and Vinod Agrawal. 2022. IMAGE PROCESSING BASED IMAGE TO CARTOON GENERATION BASED ON MACHINE LEARNING. International Research Journal of Modernization in Engineering Technology and Science (IRJMETS). https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2022/23627/final/fin_irjmets1653626939.pdf Accessed: 2025-04-26.

[10] mnicnc404 and contributors. [n. d.]. CartoonGan-tensorflow Repository (Dataset Source). https://github.com/mnicnc404/CartoonGan-tensorflow/tree/master. Accessed: May 28, 2025. Used as a source for dataset images..

[11] OpenCV Team. 2025. OpenCV 4.x Tutorials and Documentation. https://docs.opencv.org/4.x/d9/df8/tutorial_root.html. Accessed: May 28, 2025.

[12] ptran1203 and contributors. [n. d.]. pytorch-animeGAN Repository (Dataset Source). https://github.com/ptran1203/pytorch-animeGAN/tree/master. Accessed: May 28, 2025. Used as a source for dataset images..

[13] Roshani Raut, Anita Devkar, Pradnya S. Borkar, Radha Deoghare, and Sapna Kolambe. 2024. Generative Adversarial Network Approach for Cartoonifying Image Using CartoonGAN. Journal of Engineering Sciences (JES). https://doi.org/10.52783/jes.666 Source mentioned as journal.esrgroups.org. Accessed: 2025-04-26.

[14] Saturn Cloud. 2023. AnimeGAN: Overview, Architecture, and Applications. https://saturncloud.io/glossary/animegan/. Accessed: May 28, 2025.

[15] SystemErrorWang and contributors. [n. d.]. White-box-Cartoonization Repository (Dataset Source). https://github.com/SystemErrorWang/White-box-Cartoonization. Accessed: May 28, 2025. Used as a source for dataset images..

[16] TachibanaYoshino and contributors. [n. d.]. AnimeGANv3 Repository (Dataset Source). https://github.com/TachibanaYoshino/AnimeGANv3. Accessed: May 28, 2025. Used as a source for dataset images..

[17] vinesmsuic and contributors. 2020. White-box-Cartoonization-PyTorch: A PyTorch implementation of White-box Cartoonization. https://github.com/vinesmsuic/White-box-Cartoonization-PyTorch. Accessed: May 28, 2025.

[18] Xinrui Wang and Jinze Yu. 2020. Learning to Cartoonize Using White-Box Cartoon Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8090–8099. https://doi.org/10.1109/CVPR42600.2020.00811