A Comparative Study of Neural Style Transfer Methods

Adam Sattout
Hacettepe University

Ezel Bayraktar
Hacettepe University

Figure 1: Example on Neural Style Transfer for a Cat Image

ABSTRACT

Neural Style Transfer (NST) enables the synthesis of artistic images by blending the content of one image with the style of another. In this study, we conduct a comparative analysis of three recent NST models—StyleID, StyleShot, and StyTr²—evaluating them across multiple axes, including content preservation, style fidelity, and aesthetic quality. To complement standard quantitative metrics such as LPIPS, SSIM, and PSNR, we introduce a novel evaluation method, LLM as a Judge, which leverages GPT-4o Vision to provide structured subjective rankings based on human-like artistic judgment. Experiments on 50 diverse content-style pairs from the Viktov dataset show that StyTr² consistently preserves content and is favored for aesthetic appeal, while StyleShot excels in style reproduction. StyleID lags behind on both fronts. Our findings highlight the limitations of existing metrics in capturing perceptual quality and demonstrate the potential of multimodal large language models as scalable, human-aligned evaluators.

1 INTRODUCTION

Neural Style Transfer (NST) aims to render content from one image in the artistic style of another, blending visual aesthetics and structural fidelity. This interdisciplinary task intersects computer vision and computational creativity, allowing for the creation of stylized imagery that mimics renowned art forms. However, producing high-quality stylizations that balance content preservation, style fidelity, and aesthetic appeal remains a significant challenge. These three objectives are often in tension: style transfer models may emphasize artistic texture at the expense of structure or oversimplify style to maintain spatial fidelity.

As the field progresses from optimization-based and convolutional methods to modern transformer and diffusion-based techniques, the importance of fair and comprehensive evaluation becomes more pronounced. Traditional metrics such as LPIPS and SSIM offer partial insight, focusing primarily on perceptual similarity or structural preservation. However, they fall short in capturing the human perception of aesthetic quality or overall coherence.

To bridge this gap, we introduce a novel evaluation strategy, LLM As a Judge, which utilizes GPT-4o Vision's multimodal capabilities to rank stylized outputs based on artistic criteria. This complements our quantitative benchmarking, offering a human-aligned, scalable judgment tool. Our study compares four representative methods—CycleGAN, StyleID, StyTr², and StyleShot—on a diverse set of content-style pairs from the Viktov dataset, evaluating both objective metrics and subjective preferences.

2 RELATED WORKS

There are many studies with different techniques proposed so far in the field of single-image style transfer. The early-stage techniques mainly focus on non-neural texture synthesis and patch-based methods, such as statistical sampling [1], image quilting [2], and multiscale histogram matching [3]. However, these approaches are mainly limited to homogeneous textures, fail to capture global semantic features, and require extensive manual tuning, making them inefficient and less effective for complex style representations.

With the rise of deep learning and convolutional neural networks (CNNs), feature-based neural style transfer models began to dominate the state-of-the-art (SOTA) performance due to their ability to extract high-level content and style features from pre-trained networks [4, 5, 6, 7]. Although neural methods significantly improved style fidelity and visual quality, early optimization-based approaches were computationally expensive and impractical for real-time applications, while feedforward models and conditional networks often lacked generalization capabilities, being restricted to a fixed number of trained styles. Furthermore, universal style transfer techniques introduced trade-offs, leading to occasional content

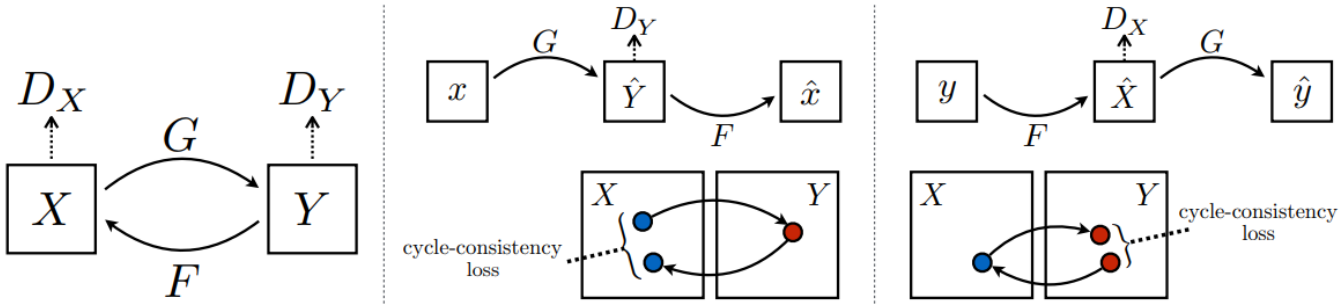


Figure 2: CycleGAN cycle-consistency loss.

distortion and lower stylization quality compared to optimization-based baselines.

Therefore, in recent years, researchers have focused on developing more flexible, efficient, and generalizable style transfer models, leveraging adversarial learning, attention mechanisms, and transformer-based designs. Until the methods that will be analyzed and compared in this study, namely CycleGAN [8], StyTr² [9], StyleID [10], and StyleSHOT [11], were proposed, CycleGAN [8] addresses the lack of paired training data by proposing an unpaired image-to-image translation framework, combining adversarial learning with a cycle-consistency loss to enable domain mappings without requiring paired datasets. StyTr² [9] introduces a transformer-based style transfer architecture that enhances the modeling of long-range dependencies and improves content-style disentanglement through self-attention mechanisms, outperforming traditional CNN-based methods. StyleID [10] takes a novel one-shot approach by encoding both the style and the input image separately and injecting them directly into a pretrained Stable Diffusion model, achieving stylization without the need for additional fine-tuning. StyleSHOT [11] further refines this idea by designing a dual-encoder framework, where a style encoder extracts a style embedding and a content encoder extracts content features, both of which are fused and injected into a Stable Diffusion model to generate high-quality, stylized outputs.

Despite these innovations, evaluation remains underdeveloped. Metrics like LPIPS, SSIM, and PSNR address different facets of quality but cannot capture nuanced artistic judgments. While simple user surveys can be expensive to operate and unscalable. A small body of work has explored human-in-the-loop or prompt-based evaluations, but these remain informal or non-reproducible. Our proposed GPT-4o Vision-based judging protocol addresses these limitations with repeatable, human-aligned assessments of content, style, and aesthetics.

3 METHODS

In this study we will compare between the upcoming methods using the approach that we will explain. But first, let's explain the used methods

3.1 CycleGAN

CycleGAN is a framework proposed to solve the problem of image-to-image translation when paired training data is not available. Traditional Neural Style Transfer (NST) techniques and supervised image translation models require datasets where each input image has a corresponding output image (aligned pairs), which are often expensive, impractical, or impossible to obtain. CycleGAN addresses this limitation by introducing a model capable of learning a translation between two domains using unpaired data.

$$\mathcal{L}_{\text{identity}}(G,F) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(y)-y\|_1] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(x)-x\|_1] \quad (1)$$

The core idea behind CycleGAN is to learn mappings between two image domains, X and Y, without requiring paired examples. This is achieved by using a dual generator-discriminator architecture. Specifically, two generators are trained: $G : X \rightarrow Y$ maps images from domain X to domain Y, and $F : Y \rightarrow X$ maps images from domain Y to domain X. Correspondingly, two discriminators D_Y and D_X are trained to distinguish between real and generated images in domains Y and X, respectively.

The adversarial loss is used to ensure that the stylized images are indistinguishable from real images in the target style domain. For the generator G and discriminator D_Y, the adversarial loss is defined as in equation 1,

$$\mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1-D_Y(G(x)))] \quad (2)$$

A similar adversarial loss is formulated for F and D_X.

However, merely using adversarial losses can lead to mode collapse or unrealistic translations. This is because the generator can adapt to a specific image that can always fool the discriminator, resulting in all the images in our X domain producing the same stylized image regardless of its content. To address this, CycleGAN introduces the cycle-consistency loss (figure 4), which enforces that an image translated to the other domain and then back to the original domain should closely resemble the original image. Formally, the cycle-consistency loss is:

$$\mathcal{L}_{\text{cyc}}(G,F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x))-x\|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y))-y\|_1] \quad (3)$$

This constraint prevents the generators from making arbitrary transformations that lose important content.

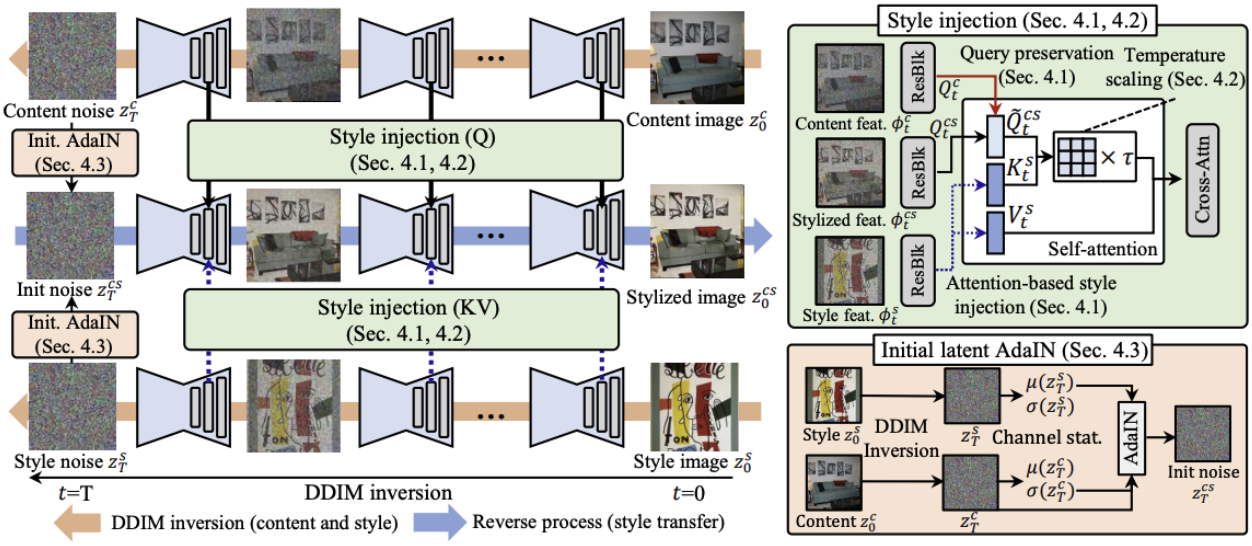


Figure 3: StyleID Pipeline.

An additional identity loss can be optionally included to encourage generators to preserve content when the input already belongs to the target domain. It is defined as:

This identity loss is particularly useful in applications where color preservation or fine structural details are critical.

The overall objective function that CycleGAN optimizes is the weighted sum of these losses:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}}(G, F) + \lambda_{\text{id}}\mathcal{L}_{\text{identity}}(G, F) \quad (4)$$

where λ_{cyc} and λ_{id} are hyperparameters controlling the relative importance of the cycle-consistency and identity losses.

However, this approach can have some downsides. First it is a GAN structure, meaning that maintaining a health training pace for both the generators and descriminators is hard and generators can produce limited diversity in outputs, collapsing to a few modes rather than capturing the full distribution. It also may induce cycle loss missalignment where the cycle-consistency constraint can sometimes force the model to produce unrealistic mappings just to satisfy the cycle. In addition to that it describes a way for turning an image from a real domain into a style domain, requiring a whole new model and a style domain dataset for each style. Which is why we will not include it in our final test as the other methods work with various image-style pairs.

3.2 StyleID

StyleID is a training-free style transfer method built on a frozen latent diffusion model (e.g. Stable Diffusion 1.4). It works in four stages: inversion, attention injection, color alignment, and synthesis (see Figure 3).

1. DDIM inversion and feature collection. We use the diffusion autoencoder encoder E to map images into latent space:

$$z_0^c = E(I_c), \quad z_0^s = E(I_s).$$

Here, z_0^c and z_0^s are the latent representations of the content and style images, respectively. Next, we apply DDIM inversion for T steps to turn each latent back into noise, collecting a full sequence of noisy latents:

$$z_{0:T}^c = \text{DDIMInv}(z_0^c), \quad z_{0:T}^s = \text{DDIMInv}(z_0^s).$$

The operator DDIMInv runs a deterministic backward diffusion pass, yielding a trajectory from image-like latent z_0 to pure Gaussian noise z_T. During this process we record the self-attention queries Q_t^c from the content path and the keys K_t^s, values V_t^s from the style path at each timestep t.

2. Attention-based style injection. We initialize our stylized noise at the final timestep:

$$z_T^{cs} = z_T^c.$$

Then, for each reverse step $t = T, T-1, \ldots, 0$, we modify the self-attention in selected U-Net blocks as follows:

$$\widetilde{Q}_t = \gamma\, Q_t^c + (1-\gamma)\, Q_t^{cs}, \quad \gamma \in [0, 1].$$

This equation linearly blends the original content query Q_t^c with the evolving stylized query Q_t^{cs}, letting γ control how much pure content structure is preserved. Next, we overwrite the stylized key and value with those from the style image:

$$K_t^{cs} \leftarrow K_t^s, \quad V_t^{cs} \leftarrow V_t^s.$$

Finally, we recompute the attention output using a temperature-scaled dot product:

$$\phi_t^{cs} = \text{softmax}\Big(\tau \frac{\widetilde{Q}_t\,(K_t^s)^\top}{\sqrt{d}}\Big)\, V_t^s, \quad \tau > 1,$$

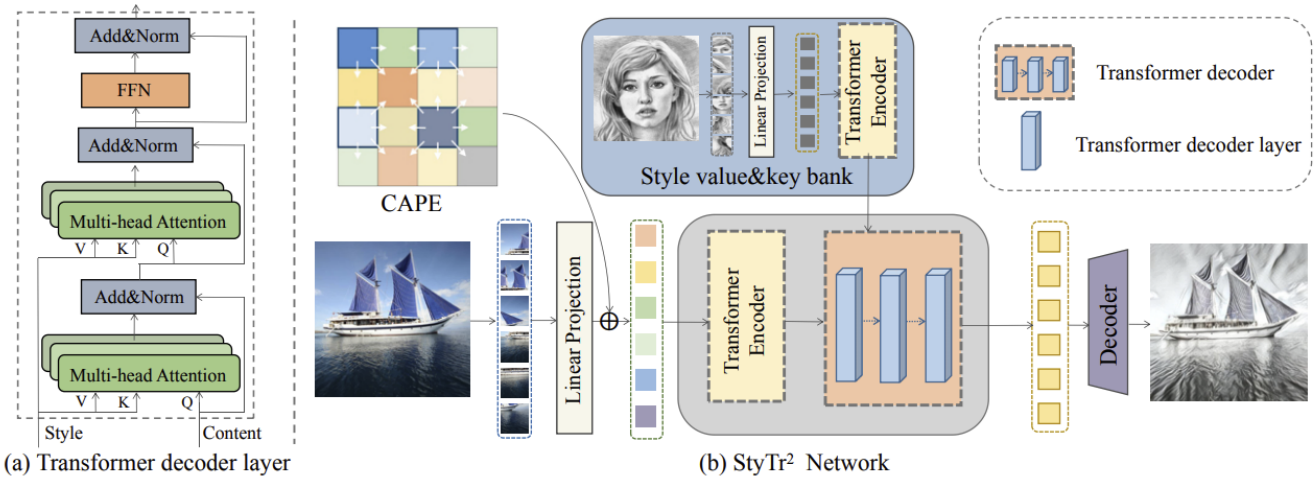


Figure 4: StyTr2 Pipeline.

where dividing by $\sqrt{d}$ normalizes the raw scores, and multiplying by τ sharpens the attention distribution to counteract any blurring from key/value swapping.

3. Initial latent AdaIN.. To transfer global color and contrast, we adjust the channel-wise statistics of the starting noise:

$$z_T^{cs} = \sigma(z_T^s)\,\frac{z_T^c - \mu(z_T^c)}{\sigma(z_T^c)} + \mu(z_T^s).$$

Here $\mu(\cdot)$ and $\sigma(\cdot)$ compute per-channel mean and standard deviation. This Adaptive Instance Normalization (AdaIN) operation imposes the style's color distribution onto the content noise while retaining its spatial structure.

4. Sampling and reconstruction. With our modified noise z_T^{cs}, we perform 50 steps of DDIM sampling to obtain z_0^{cs}, then decode back into pixel space with the diffusion decoder D:

$$\hat{I}_{cs} = D\big(z_0^{cs}\big).$$

This yields our final stylized image. On a TITAN RTX, inversion takes 8.2 s and sampling 4.2 s, for a total of 12 s per image.

Hyperparameters. We use $\gamma = 0.75$ (content–style balance), $\tau = 1.5$ (attention temperature), and $T = 50$ DDIM steps. Because no per-style optimization is needed, StyleID runs quickly and preserves fine details, though its quality depends on inversion fidelity and the manual tuning of γ, τ.

3.3 StyTr^2

StyTr^2 is a pure-transformer architecture for arbitrary style transfer. It models both global structure and fine details by attending directly across image patches. Its pipeline comprises: patch embedding, Content-Aware Positional Encoding (CAPE), dual transformer encoders, a cross-attention decoder, and a lightweight CNN upsampler (see Figure 4).

1. Patch embedding. Given an image $I \in \mathbb{R}^{H\times W\times 3}$, we split it into non-overlapping patches of size $m \times m$ (with $m = 8$), producing $L = HW/m^2$ tokens. A learnable linear projection maps each flattened patch into a C-dimensional embedding:

$$E = \{\mathbf{e}_1, \ldots, \mathbf{e}_L\}, \quad \mathbf{e}_i \in \mathbb{R}^C.$$

This converts the 2D image into a sequence suitable for transformer processing.

2. Content-Aware Positional Encoding (CAPE).. Standard sinusoidal encodings only reflect patch distance, not image content or scale. CAPE instead derives position embeddings from the content itself:

$$P_L = F_{\text{pos}}\big(\text{AvgPool}_{n\times n}(E)\big),$$

where we average-pool the token grid into $n \times n$ (here $n = 18$) and run a 1×1 conv F_{pos} to learn a coarse positional map P_L. For each token at grid location (x, y), we interpolate neighboring entries:

$$P_{\text{CAPE}}(x, y) = \sum_{k=0}^{s}\sum_{l=0}^{s} a_{kl}\, P_L(x_k, y_l),$$

with weights a_{kl} from bilinear interpolation. Finally, we add P_{CAPE} to each patch embedding, making the positions both content-aware and scale-invariant.

3. Transformer encoders. We have two separate 6-layer encoders—one for content embeddings $E_c + P_{\text{CAPE}}$, one for style embeddings E_s. In each encoder layer, we compute:

$$Q = ZW_q, \quad K = ZW_k, \quad V = ZW_v,$$

where Z is the input sequence. Multi-head self-attention then fuses these:

$$\text{MSA}(Q, K, V) = \text{Concat}\big[\text{Attn}_h(Q, K, V)\big]\, W_o,$$

followed by residual connections, layer normalization, and a two-layer feed-forward network.

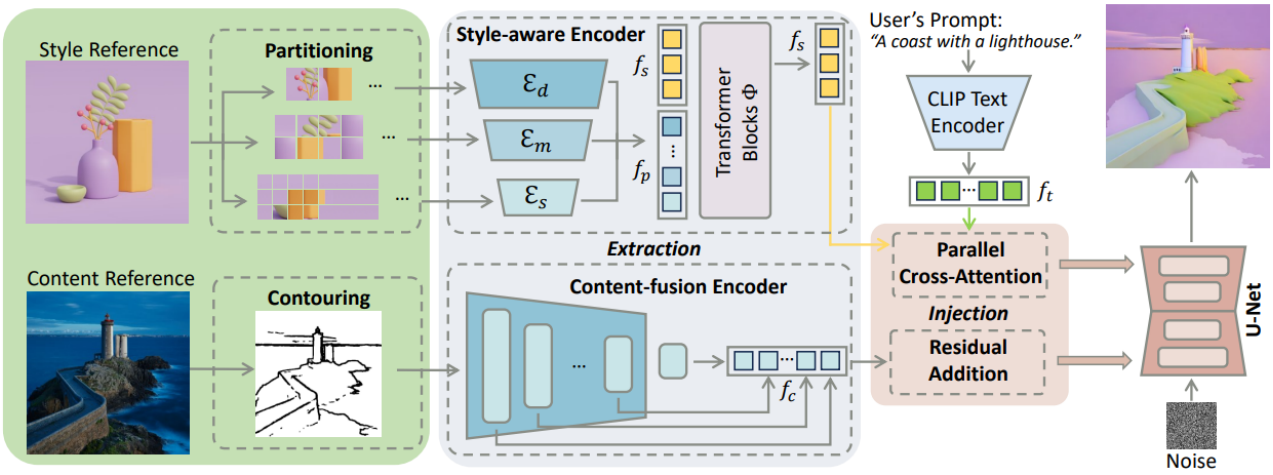


Figure 5: StyleShot High Level Architecture.

4. Cross-attention decoder. To inject style into content, each decoder layer takes the encoded content (plus CAPE) as query and the encoded style as key/value:

$$Q = (Y_c + P_{\text{CAPE}})\, W_q^d, \quad K = Y_s\, W_k^d, \quad V = Y_s\, W_v^d.$$

A cross-attention block computes how each content token should attend to style tokens, followed by a self-attention block (to remix content contexts) and a feed-forward network—all with skip connections and LayerNorm.

5. CNN upsampler. The decoder output is a sequence of length L with dimension C. We reshape it to spatial size $\frac{H}{m} \times \frac{W}{m} \times C$ and apply three Conv(3×3)–ReLU–Upsample×2 layers, yielding a final image of size $H \times W \times 3$.

6. Training losses. We train end-to-end to simultaneously preserve content and replicate style. Let $\hat{I}$ be the output. We use:

- **Content loss on VGG features**: $\mathcal{L}_c = \sum_i \|\phi_i(\hat{I}) - \phi_i(I_c)\|_2^2$.
- **Style loss** matching feature means and variances: $\mathcal{L}_s = \sum_i \left\|\mu(\phi_i(\hat{I})) - \mu(\phi_i(I_s))\right\|_2^2 + \|\sigma(\phi_i(\hat{I})) - \sigma(\phi_i(I_s))\|_2^2$.
- **Identity losses** ensuring that feeding identical content (or style) images reproduces them exactly.

In practice we set patch size $m = 8$, embedding dim $C = 512$, Adam lr = 5e-4, batch = 8, and train 160 K iterations on COCOWikiArt crops of 256^2. At inference time, a single forward pass stylizes arbitrary image sizes in under a second.

3.4 StyleShot

StyleShot is built upon three major technical innovations: the style-aware encoder, the content-fusion encoder, and the injection of resulting embeddings into a standalone Stable Diffusion model (see Figure 5)..

Existing methods often utilize a frozen CLIP image encoder to extract style representations. However, CLIP encoders are primarily optimized for semantic alignment between images and text rather than for pure style modeling. Consequently, features extracted using CLIP often entangle content and style, resulting in unstable and suboptimal style transfer performance.

To address this challenge, StyleShot introduces a specialized style-aware encoder. Unlike traditional single-scale patch partitioning approaches, the style-aware encoder employs a multi-scale patch extraction strategy. Specifically, it divides the input style reference image into patches at three scales: 1/4, 1/8, and 1/16 of the image size. These patches are processed through separate lightweight ResNet blocks corresponding to each scale to extract multi-level style embeddings that capture both low-level features such as colors and textures, and high-level features such as layouts and shading.

The multi-level embeddings are then aggregated using Transformer blocks, and learnable style embeddings are optimized jointly during training. Importantly, positional embeddings are dropped to avoid injecting unwanted spatial information, ensuring the style representations focus purely on stylistic elements.

Once extracted, the style embeddings are introduced into the denoising U-Net of a Stable Diffusion model through a parallel cross-attention mechanism. The style and text conditions are fused at each attention layer, enabling simultaneous conditioning on both style and content. Mathematically, the final latent features after injection are expressed as in Equation 5 where λ controls the balance between text and style attention outputs.

$$f' = \text{Attention}(Q, K_t, V_t) + \lambda \text{Attention}(Q, K_s, V_s) \tag{5}$$

To ensure that content structure is accurately preserved while transferring the style, StyleShot introduces a content-fusion encoder. Rather than directly using content images—which may carry their own stylistic biases—StyleShot preprocesses content references by applying contour detection through a Holistically-Nested

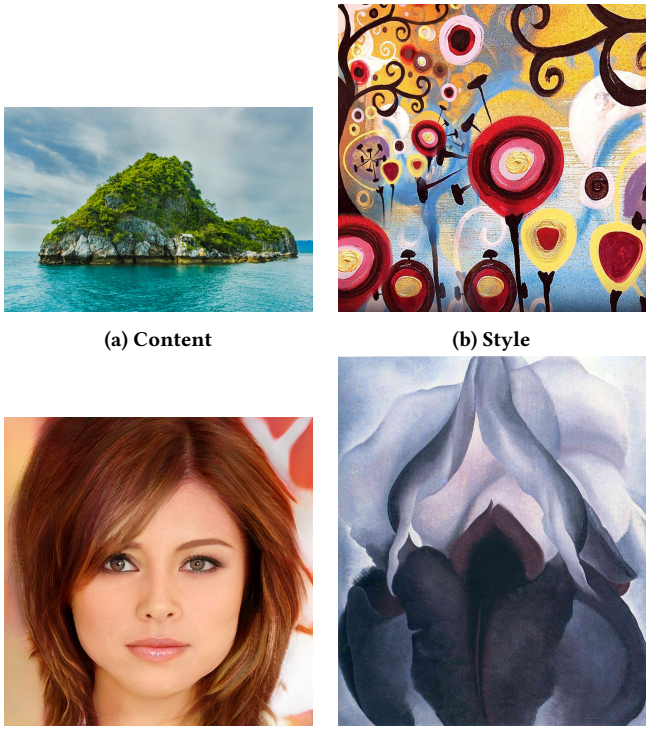
Edge Detection (HED) model combined with thresholding and dilation. This produces clean, stylized-free structural representations of the content.

The content-fusion encoder processes these contours and generates latent features for different layers of the U-Net backbone. These content features are injected into the main diffusion U-Net through residual addition at multiple levels, following a strategy similar to that of ControlNet. This design ensures that the final generated image respects the structural layout of the content input while freely adopting the desired style.

While StyleShot demonstrates substantial improvements over previous approaches, it also presents some limitations. First, although the style-aware encoder is highly effective, the paper acknowledges that alternative designs and architectures for style extraction were not exhaustively explored. Further research could lead to even better style representations. Second, the method's performance is closely tied to the diversity and quality of the StyleGallery dataset. Applications to domains outside those represented in StyleGallery, which is the dataset that they curified from different style/content databases, might require dataset expansion or fine-tuning. Finally, although contour-based content extraction effectively removes most stylistic elements, it may oversimplify highly detailed or complex content structures in rare cases, slightly affecting the realism of generated outputs.

3.5 Testing Dataset

In this study, the Victor Kitov dataset [17] (Figure 6) serves as our test set for evaluation. This dataset includes 50 content-style pairs, each comprising a content image (the scene or subject) and a style image (the reference artwork or texture) with content images varying from wild scenes to human figures, and different style images



(a) Content **(b) Style**

(c) Content **(d) Style**

Figure 6: Samples of Viktov Dataset Pairs

3.6 Evaluation Approach

Our experimental pipeline is designed to generate and assess stylized outputs from three image-style compatible Neural Style Transfer (NST) models—StyTr², StyleID, and StyleShot. The evaluation combines traditional image quality metrics, subjective user feedback, and a novel human-aligned technique we refer to as *LLM as a Judge*, enabling a comprehensive comparison of model performance across multiple dimensions.

Evaluation Setup. For each style transfer instance, the pipeline starts with a content image (original scene) and a style image (reference texture or painting). Each NST model produces a stylized version, resulting in three candidate outputs labeled 1 through 3. These outputs are shuffled and numerically annotated to eliminate positional bias.

LLM-as-a-Judge. For each style-transfer example, the process begins by providing a content image (the original scene), a style image (the reference painting or texture), and three candidate outputs, each generated by different models. These outputs are labeled 1 through 3. The system then prompts a language model (LLM) to act as an art critic and evaluate the candidates based on three criteria: Content Preservation, Style Faithfulness, and Aesthetic Quality. The LLM is strictly instructed to provide its output in JSON format, ensuring consistency and ease of processing.

While giving the results to the multimodel, we have 2 matching approaches: Listwise and Pairwise. In the Listwise Method, all three candidates are presented to the LLM in a randomized order. The LLM is asked to rank the candidates from best to worst. The analysis of the rankings reveals patterns such as how frequently each model appears in first, second, or third place, and the models' average ranks across all triplets. While the other approach involves presenting only two candidates at a time from each triplet, removing the third model to focus on a direct comparison. The LLM is asked to choose which candidate it prefers. Each duel is tallied as a win for the preferred model, and these pairwise preferences are then analyzed to determine which model is consistently favored across the different pairings. This method offers a more granular understanding of the comparative performance between the models in direct competition.

Traditional Metrics. Alongside LLM-based evaluations, we also employ several quantitative metrics to assess technical image quality:

- **PSNR (Peak Signal-to-Noise Ratio):** Measures pixel-level similarity between the stylized and reference images. Higher values suggest greater fidelity in terms of exact pixel content, which helps evaluate content and style preservation. However, it may not align well with human perception. PSNR was calculated for both content and style images.

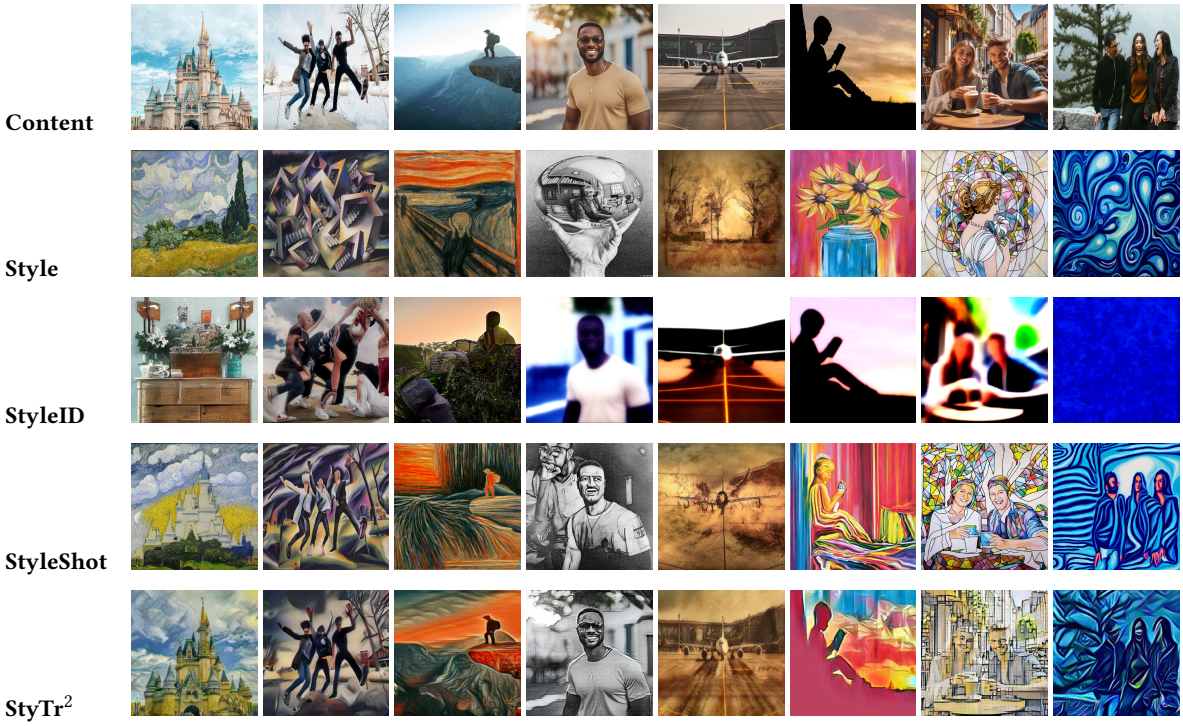


Figure 7: Comparison grid of content, style, and different stylization methods

- **SSIM (Structural Similarity Index Measure):** Assesses structural similarity by factoring in luminance, contrast, and texture. It is particularly useful for evaluating content preservation, as it considers more perceptually meaningful structural features than raw pixel values.

- **LPIPS (Learned Perceptual Image Patch Similarity):** A perceptual metric that compares deep feature activations from pre-trained networks (e.g., VGG, AlexNet). It measures how similar two images appear at a semantic level, with lower values indicating better perceptual alignment. It was also calculated for both content and style images.

User Reviews. To further incorporate human judgment, we include feedback from actual users. Participants are shown the content image, style reference, and the three stylized outputs, and are asked to select the one that best balances the three key dimensions. This subjective input provides an important layer of evaluation that captures perceptual preferences not reflected in automated metrics.

4 RESULTS

Observations from Figure 7 reveal several clear trends in model performance. First, StyleID consistently underperforms relative to the other two methods. Its outputs often exhibit either excessive content distortion or overly literal style application, sometimes resulting in a strong halo effect or, in other cases, an almost unrecognizable transformation of the input. This lack of balance between content preservation and style transfer makes StyleID the least effective across most examples.

In contrast, StyTr demonstrates stronger content preservation, which is particularly evident in examples such as the castle, the airplane, and the jumping people. In these cases, fine structural and semantic details—like architectural lines, body postures, and object contours—remain more intact, making the output more faithful to the original content.

Meanwhile, StyleShot excels in style fidelity. It captures and applies the stylistic features of the reference image more vividly, as seen in the couple and airplane examples. These outputs tend to reflect the texture, color palette, and painterly quality of the style image more convincingly than StyTr, even if some content details are sacrificed in the process.

Model	Preserves Content	Transfers Style	Aesthetically Pleasing
Style_ID	23 votes	4 votes	3 votes
StyleShot	16 votes	51 votes	24 votes
StyTr^2	41 votes	25 votes	**52 votes**

Table 1: User Voting Results

Across the quantitative metrics (Table 2), StyTr emerges as the top-performing method overall, consistently outperforming the others. It dominates in content-related metrics, clearly excelling in preserving the structural and semantic features of the original images. Even in style-related metrics, such as LPIPS-Style and PSNR-Style, StyTr edges out StyleShot, though the margin is relatively small. It is also interesting to see that SSIM score is better for StyleID than StyleShot, matching with user reviews

Model	LPIPS_C	LPIPS_S	1 - SSIM	PSNR_C	PSNR_S
Style_ID	0.7488	0.7770	0.6946	8.60	8.19
StyleShot	0.7780	0.6835	0.8359	7.84	9.47
StyTr^2	0.6088	0.6786	0.5897	11.40	10.21

Table 2: Quantitative Results, Lower LPIPS, 1-SSIM, and Higher PSNR are Better.

However, these results do not entirely align with our earlier qualitative observations, nor with the user study outcomes, which arguably carry the most weight in this domain, given the inherently subjective and perceptual nature of style transfer. In the user reviews, which directly reflect human aesthetic judgment, a more nuanced picture emerges—one that mirrors our visual inspection. Participants consistently preferred StyTr for content preservation, recognizing its ability to retain critical image details. At the same time, they favored StyleShot for its more vivid and faithful style transfer,

This discrepancy between metric-based rankings and human perception underscores a key limitation of existing quantitative evaluations. While metrics like LPIPS and PSNR provide useful approximations, they may not fully capture the subtleties of artistic quality and style perception, which are central to the goals of Neural Style Transfer.

Model	1st Place
StyleShot	48
Style_ID	0
StyTr^2	2

Table 3: Listwise Ranking (1st Place Votes)

Model	Win Rate
Style_ID	0.32
StyleShot	0.57
StyTr^2	0.61

Table 4: Pairwise Duelling (Win Rate)

When examining our LLM-as-a-Judge evaluations, the results present an interesting contrast depending on the ranking method used. The Listwise Ranking approach suggests that StyleShot overwhelmingly dominates, frequently being placed first with little competition from the other models. However, this outcome appears to overstate StyleShot's performance, especially when considered alongside the quantitative metrics and user reviews, which indicate a more balanced performance. Given that StyTr consistently outperforms in content preservation and holds its own in style fidelity, the Listwise results likely reflect a bias or inconsistency in judgment aggregation when comparing all three outputs simultaneously.

On the other hand, the Pairwise Ranking results offer a more nuanced and credible assessment. In this setup, StyTr achieves the highest win rate, reflecting its consistent strength in preserving meaningful content and having better aesthetics across diverse examples. At the same time, StyleShot remains competitive, particularly in pairings where strong stylistic transformation is visually favored. StyleID, by contrast, falls behind in most pairwise comparisons, which aligns with both user sentiment and qualitative observations.

Overall, while Listwise judgments may exaggerate one model's perceived dominance, the Pairwise method proves more reliable, capturing the complex trade-offs between content and style in a way that better matches both human preferences and our broader evaluation metrics.

5 CONCLUSION

This study compared three neural style transfer methods—StyleID, StyleShot, and StyTr²—using human evaluation, quantitative metrics, and a novel GPT-4o-based evaluation framework, LLM as a Judge. StyleID consistently underperformed, producing outputs that often lacked structural coherence or convincing style application. StyleShot excelled at style fidelity, frequently capturing vivid textures and color palettes from the style image, but sometimes at the cost of distorting content. StyTr² emerged as the most balanced method, achieving the best content preservation scores and also being the top choice in aesthetic quality according to user reviews, suggesting it produces the most visually pleasing results overall.

The introduction of LLM as a Judge proved to be a valuable contribution to the evaluation pipeline. The framework offered a scalable, perceptually informed assessment of stylization quality that aligns in part with human preferences. However, our findings also highlight limitations in its current formulation. While the listwise rankings heavily favored StyleShot, they arguably overstated its advantage, especially when juxtaposed with metric scores and user judgments that indicated a closer competition. In contrast, the pairwise judgment mode produced more balanced results, aligning better with both quantitative and qualitative assessments. This suggests that while LLM-based evaluations are promising, the structure of the input and the format of the comparison significantly influence outcomes.

Overall, our study shows that no single model dominates across all aspects of neural style transfer. Each method embodies a different trade-off between style fidelity and content preservation, and evaluation requires both rigorous metrics and perceptual insight. We recommend the use of hybrid evaluation strategies—including both metric-based and language model-based approaches—to better reflect the subjective and multifaceted nature of stylization quality. In future work, refining the prompting, calibration, and aggregation techniques of LLM as a Judge could further improve its reliability and make it a valuable standard in artistic image generation assessment.

Future work can include more NST methods or explore fine-tuning GPT-4o prompts to better balance attention between content and style when judging outputs. Additional directions include integrating multiple LLM evaluations for consensus, extending the framework to video or temporal coherence in stylization, and developing hybrid models that dynamically adjust between content and style priorities based on user preference or downstream task.

REFERENCES

[1] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of ICCV*, 1999.
[2] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of SIGGRAPH*, 2001.
[3] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of SIGGRAPH*, 1995.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of ECCV*, 2016.
[6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *Proceedings of ICLR*, 2017.
[7] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Proceedings of NeurIPS*, 2017.
[8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of ICCV*, 2017.
[9] Yuxin Deng, Xiaowei Tang, Jiangmiao Pan, Chen Change Loy, and Xiaoou Liu. Stytr^2: Unifying style transfer and image translation with transformer. In *Proceedings of CVPR*, 2022.
[10] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
[11] Junyao Gao et al. Styleshot: A snapshot on any style. *arXiv preprint arXiv:2407.01414*, 2024.
[12] The flickr image dataset. https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset.
[13] Ms-coco image dataset. https://cocodataset.org/.
[14] Wikiart style image dataset. https://github.com/cs-chan/ArtGAN/tree/master/WikiArt%20Dataset.
[15] Journeydb style image dataset. https://journeydb.github.io/.
[16] Stylegallery style image dataset. https://github.com/open-mmlab/StyleShot/blob/main/DATASET.md.
[17] Victor Kitov. Viktov style transfer dataset. https://github.com/victorkitov/style-transfer-dataset/tree/main, 2023. Accessed: 2025-10-05.