Sketch Colorization Using Diffusion Models and Photo-Sketch Correspondence

Axel Delaval, Adama Koïta
École Polytechnique — Telecom-Paris
Email: axel.delaval@gmail.com, adama.koita@telecom-paris.fr

GitHub Repository HuggingFace

***Abstract*—Sketch colorization has gained significant attention in recent years, particularly in the context of anime and manga artwork. Recent advancements in diffusion models and deep learning have led to substantial improvements in automated colorization techniques. This project explores diffusion-based approaches to anime sketch colorization.**

Our final model draws inspiration primarily from AnimeDiffusion [1], MangaNinja [2], and photo-sketch correspondence models [3]. We also explored ideas from feature matching and robust visual representation models [4]–[10].

Due to computational constraints, we first trained our model on a subset of the Danbooru dataset [11]. However, we also curated an extensive dataset using AnimeDiffusion [1] enriched with augmented deformation flows to generate diverse training samples, which we used in a second step.

***Index Terms*—Sketch Colorization, Diffusion Models, Anime Art, Deep Learning, Feature Matching, Image Processing**

Contents

1 Introduction

1.1 Traditional approaches to sketch colorization

Early sketch colorization was a manual, labor-intensive process by artists. Initial computational methods eased this task using heuristic color propagation and transfer. For example, Welsh et al. [12] introduced exemplar-based colorization by matching luminance and texture statistics from a grayscale image to a colored reference, while Ironi et al. [13] enhanced this approach with a texture classifier for more localized transfer. Other methods, such as those by Levin et al. and Sýkora et al. [14], used minimal user inputs like scribbles or flood-fill, but these rule-based techniques, reliant on low-level cues, struggled with complex details and semantic understanding.

1.2 CNN-Based and GAN-Based Reference Colorization Models

Deep learning significantly advanced colorization quality and automation. Early CNN models, such as Zhang et al. (2016) [15], addressed general colorization but often produced average results. To allow more control, researchers introduced reference images and adversarial training. Isola et al. [16] (2017) demonstrated that conditional GANs (e.g., the pix2pix framework) could directly map edges or sketches to colored images. Later, He et al. [17] (2018) developed a CNN that incorporated both a target sketch/photo and a reference image to apply the reference color scheme more faithfully, while Sun et al. [18] (2019) applied a similar conditional GAN approach for icon drawing colorization.

1.3 Feature Matching Techniques for Reference-Guided Colorization

A major challenge is aligning the reference image to the sketch. To address this, feature matching or correspondence modules are used. Lee et al. [19] (2020) introduced an augmented self-reference training strategy by geometrically distorting an image to create exact correspondences, enabling

the training of a dense semantic correspondence network for precise color mapping.

1.4 Recent Advances: Transformers, Diffusion Models, and Hybrid Approaches

Recent research has begun to replace or augment CNNs with transformers, which capture global color dependencies more effectively. For example, Vision Transformers have been used to propagate user color hints in real-time [20]. Hybrid models, like Ma et al. [21] (2024), combine a ResNet backbone with transformer blocks and superpixel-guided attention for improved fidelity in line art colorization. Diffusion models, such as those used in Stable Diffusion, generate images via iterative denoising. Yan et al. [22] (2024) propose a latent diffusion model that balances the influence of the sketch and reference, while AnimeDiffusion [1] (2023) and MangaNinja [2] (2025) achieve precise, vivid outputs. Other hybrid methods combine latent space interpolation with CNN or GAN decoders to transfer global color tone while preserving sketch structure.

1.5 Benchmarking and Evaluation

With many methods available, standardized benchmarks have emerged. Metrics such as Fréchet Inception Distance (FID) assess realism and diversity, while SSIM, PSNR, and LPIPS evaluate fidelity and perceptual similarity. For example, Lee et al. [19] (2020) showed their correspondence-based model outperformed prior GAN-based methods on FID and introduced a semantic-PSNR metric to verify accurate color transfer. In addition to objective metrics, user studies are crucial for evaluating perceived quality.

1.6 Feature Matching for Reference-Based Colorization

Precise alignment between the sketch and reference is essential. Lee et al. [19] propose a self-reference training strategy to generate synthetic correspondences for color transfer, while advanced techniques like SuperGlue [4], LoFTR [5], and DinoV2 [6] further enhance the robustness of color mapping.

2 Methodology

2.1 Architecture overview

AnimeDiffusion (2023) [1] paved the way for diffusion-based reference-guided sketch colorization. While its results are impressive — especially in terms of facial color accuracy — it shows some limitations when handling intricate details like accessories or clothing. In contrast, MangaNinja (2025) [2] improves on this by incorporating advanced attention mechanisms and deformation-aware modules, leading to finer detail preservation and better generalization across styles, characters, and scenes (including backgrounds).

Given the high computational demand of MangaNinja — reportedly requiring 8 days of training on an A100-80G GPU while AnimeDiffusion required 2 days — we opted to retain its conceptual design while simplifying the architecture to fit within our computational constraints. Our approach balances performance and efficiency, retaining key architectural insights while adopting lighter modules for training feasibility.

2.2 Our architecture

Our model architecture (Figure 1) reuses the core dual-branch design of MangaNinja but replaces heavy components with more efficient counterparts. The system consists of the following key modules:

- **Reference U-Net (8,304,256 parameters)**: Processes the reference image through a dedicated U-Net to extract rich semantic and color features. We apply a 32×32 Progressive Patch Shuffle (PPS) to the input image to enhance robustness against spatial misalignments, helping the model generalize to varied character poses and layouts. The reference U-Net is fully trainable.
- **Denoising U-Net (54,803,347 parameters)**: This is the generative backbone responsible for diffusion-based denoising. It receives a noisy version of the sketch and iteratively predicts the clean colorized output. The denoising U-Net incorporates cross-attention layers that fuse in guidance from the Reference U-Net. Like in AnimeDiffusion, this U-Net is trained to predict noise during pre-training and later fine-tuned using reconstruction loss.
- **Progressive Patch Shuffle (PPS)**: Inspired by data augmentation strategies, PPS randomly shuffles increasingly fine-grained patches of the reference image. This prevents the network from relying on strict spatial correspondences and forces it to learn more robust, local color features, which is critical for generalization to unseen poses or compositions.
- **PSC Model (Photo-Sketch Correspondence) (104,145,030 parameters)**: We use a frozen deformation flow model to estimate dense correspondences between the reference image and the sketch. This flow map warps the reference features to better align them with the sketch's structure. This step enhances the color transfer accuracy, particularly for sketches with large structural deviation from the reference.
- **Cross-Attention Fusion (CA)**: Reference features are injected into the denoising U-Net using cross-attention layers. This allows the model to selectively attend to relevant regions in the reference image and propagate those colors into the generated result.

Compared to the original MangaNinja, our architecture significantly reduces memory and compute requirements while retaining high visual quality. It supports one-shot reference-based colorization and can be extended to multiple references or even animated sequences with temporal coherence.

3 Detailed architecture and implementation

In this section, we provide a detailed breakdown of the implementation of our model components, including the Reference U-Net, Denoising U-Net, and the PSC-enhanced diffusion process. We describe the layer configurations, architectural rationale, and key mathematical operations that govern the system.

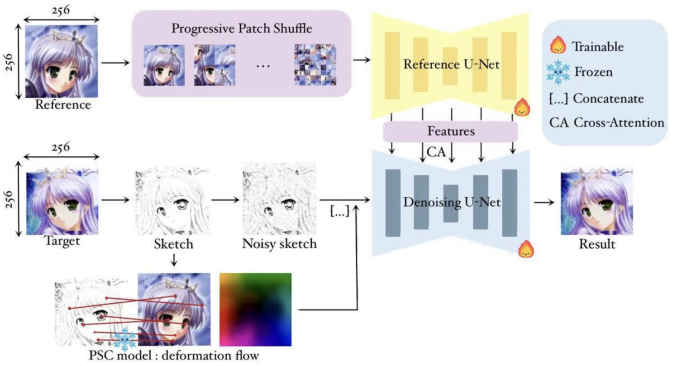

Fig. 1: Architecture diagram of our lightweight MangaNinja-inspired diffusion model.

3.1 *Reference U-Net*

The Reference U-Net processes the reference image to extract color and semantic features that will guide the colorization. It consists of a three-stage encoder with progressively smaller spatial resolutions and more expressive features.

Input: RGB reference (patched) image $I_{\text{patch}} \in \mathbb{R}^{3\times H\times W}$

Structure Overview:

- **enc1 (Shallow Features):** A convolution layer followed by normalization and activation, then several residual blocks to refine low-level details.
- **down1 (Mid-level Features):** The resolution is halved using a downsampling layer. More residual blocks extract intermediate features like edges and textures.
- **down2 (Deep Features):** Another downsampling step leads to deep semantic features. This stage includes more residual blocks for robust abstraction.
- **Skip Connection:** Mid-level features are projected and pooled, then added to the deep features to help preserve structural information.
- **Bottleneck Attention:** Three multi-head self-attention blocks help the model capture global context and match style elements from the reference.
- **Final Output:** A final convolution refines the output feature map $F_{\text{ref}} \in \mathbb{R}^{256\times H/4\times W/4}$, ready to be used by the denoising network.

Component	# Parameters
Stage 1 (enc1)	224,256
Stage 2 (down1)	1,018,496
Stage 3 (down2)	4,068,608
Skip connection (skip_conv)	33,024
Bottleneck attention	2,369,280
Final refinement	590,592
Total (Reference U-Net)	**8,304,256**

TABLE I: Parameter breakdown of the Reference U-Net.

3.2 *Denoising U-Net*

The Denoising U-Net is the generative core of our model. It removes noise from a corrupted sketch to reconstruct the clean, colorized version. The process is guided by both the sketch and the reference features, using a symmetric encoder-decoder structure with attention at the bottleneck.

Input: A tensor $x = [x_t, x_{\text{cond}}] \in \mathbb{R}^{9\times H\times W}$:

- x_t: the noisy image at timestep t, sampled as:

$$x_t = \sqrt{\gamma_t}\, x_0 + \sqrt{1-\gamma_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

- $x_{\text{cond}} = [x_{\text{sketch}}, x_{\text{warp}}]$: 6-channel conditioning tensor combining the sketch and the deformation-warped reference.

Structure Overview:

- **Encoder:** Four stages with increasing depth and downsampling. Each stage includes residual blocks with temporal embeddings that incorporate timestep t:

$$h_l = \text{ResBlock}(h_{l-1}, \text{Embed}(t))$$

Feature maps are stored at each stage to be reused in the decoder.

- **Bottleneck:** At the lowest resolution, we inject reference features via cross-attention:

$$h' = \text{CrossAttn}(h,\, F_{\text{ref}})$$

where F_{ref} are features from the Reference U-Net. This allows the model to align and borrow colors from the reference image.

- **Decoder:** Mirrors the encoder. At each level, feature maps from the encoder are concatenated with the decoder features:

$$h'_l = \text{Upsample}(\text{ResBlock}([h_l, h_{\text{skip}}], \text{Embed}(t)))$$

- **Output:** A final convolution predicts the noise estimate $\tilde{\epsilon}_\theta$, used in the diffusion loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \tilde{\epsilon}_\theta(x_t, t, x_{\text{cond}}, F_{\text{ref}})\|^2\right]$$

Component	# Parameters
Input conv	640
Time embedding	82,432
Encoder blocks	11,662,464
Bottleneck block 0	4,853,248
Cross-attention block	3,186,688
Bottleneck block 2	4,853,248
Decoder blocks	30,147,856
Output head	323
Total (Denoising U-Net)	**54,803,347**

TABLE II: Parameter breakdown of the Denoising U-Net.

3.3 *Attention modules*

- **RefUNetAttentionBlock:** Transformer-like block using MHSA on spatial tokens (flattened $H \times W$) with LayerNorm, MLP, and residual connections.
- **CrossAttentionBlock:** Multi-head attention where queries come from the main branch and keys/values from the reference features, projected to same dimension.

3.4 Residual block

The Residual Block (introduced by He et al. in 2015 [23]) is a core building unit used in both the Reference U-Net and the Denoising U-Net. It enables the model to learn transformations while preserving input information through skip connections, improving gradient flow and training stability.

Input: A feature map $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$, and optionally a time embedding vector $t \in \mathbb{R}^{d_t}$.

Structure:

- **First stage:** GroupNorm $\rightarrow$ SiLU $\rightarrow$ Conv2D$(C_{\text{in}}, C_{\text{out}})$
- **Time embedding (optional):** If provided, t is projected via a linear layer and added to the intermediate tensor:

$$h = h + \text{Linear}(t)[:, :, None, None]$$

- **Attention (optional):** A CBAM block can be applied to the intermediate feature map to enhance focus on important spatial and channel locations.
- **Second stage:** GroupNorm $\rightarrow$ SiLU $\rightarrow$ Dropout $\rightarrow$ Conv2D$(C_{\text{out}}, C_{\text{out}})$
- **Skip connection:** If $C_{\text{in}} \neq C_{\text{out}}$, the input is projected using a 1×1 convolution.

Output: The final output is the sum of the processed feature map and the skip connection:

$$\text{ResidualBlock}(x, t) = h + \text{Proj}(x)$$

This design allows the block to incorporate temporal information (used in diffusion), focus attention where needed (via CBAM), and maintain architectural flexibility across channel dimensions.

4 Dataset and Experimental Setup

Our training dataset is a subset of the Danbooru dataset [11], enhanced AnimeDiffusion [1]. The dataset is available on HuggingFace. Augmentations include:

- Sketch generation via contrast-inverted line extraction.
- Deformation flow augmentations for structural diversity.

We trained models on 1-5 GPU P100 nodes, dynamically allocating resources based on availability.

5 Experimental Results

5.1 Training Performance

Training was conducted for 199 epochs, with initial results appearing around epoch 39. We trained our model first with a smaller dataset (1.86GB) until the epoch 170. Then we switch the training on the full dataset (8.3GB) from the epoch 171 to the epoch 199. The loss curve illustrates the training dynamics of the model starting from epoch 35. Initially, the loss decreases rapidly and then stabilizes around a relatively low value, indicating that the model is effectively learning from the initial dataset and converging to a consistent performance level. However, around epoch 170, a sharp spike in the loss is observed, which coincides with the switch to a larger or more complex dataset. This sudden increase suggests that the new data introduced a distribution shift or higher variability, making it more difficult for the model to generalize immediately. Following this jump, the loss remains high but relatively stable, indicating that while the model struggles with the new dataset, it does not deteriorate further (see Figure 2).

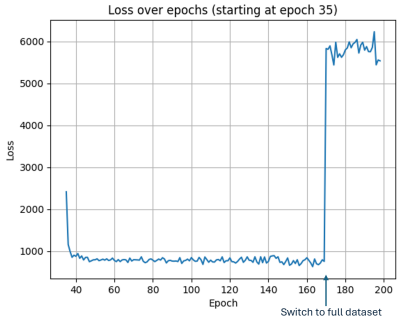

Fig. 2: Loss over epochs. We start the training with a small portion of the dataset then at epoch 170 we trained the model with the full dataset.

5.2 Quantitative Evaluation

FID	SSIM	PSNR	LPIPS
381.78	0.1038	2.5230 dB	0.9525

TABLE III: Evaluation Metrics

The evaluation results indicate that the generative model's performance is quite poor. A FID of approximately 382 suggests a significant discrepancy between the distributions of generated and real images. The very low SSIM (≈ 0.10) and the extremely low PSNR (≈ 2.52 dB) demonstrate that the generated images lack structural similarity and suffer from severe signal degradation compared to the ground truth. Additionally, an LPIPS score near 0.95 confirms that the perceptual similarity between the generated images and the reference images is unsatisfactory.

5.3 Qualitative Results

We compared generated images at different training stages:

- **After 50 epochs:** Colors were inconsistent and lacked detail.
- **After 122 epochs:** More coherent and sharper images with better structure.
- **After 199 epochs:** The colors assigned are more accurate.

Example results are shown in Figure 3.

Discussion: *The evaluation metrics indicate that while the results improve through the epochs – particularly after augmenting the dataset size around epoch 170 – there is still significant room for improvement. For instance, a FID of 381.78 suggests that the distribution of generated images is still far from that of the real images. Additionally, the very low SSIM of 0.1038 and the PSNR of only 2.5230 dB highlight that the structural similarity and signal quality between the generated outputs and the reference images are poor. The LPIPS score of 0.9525 further confirms that the perceptual similarity*

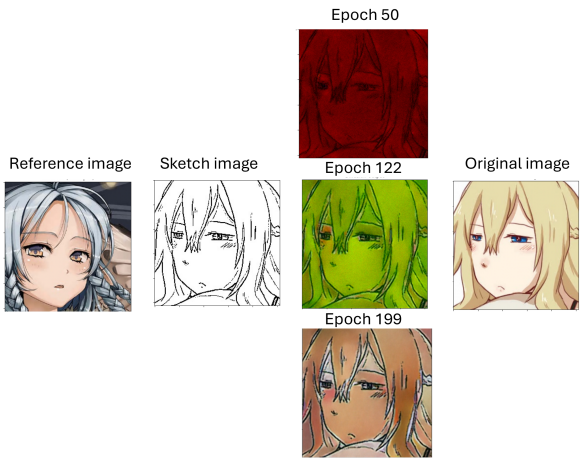

Fig. 3: Result of the diffusion at different epochs accross the training

is unsatisfactory. To achieve optimal results, several aspects must be addressed. First, 199 epochs may not be sufficient for convergence, especially considering that AnimeDiffusion [1] trained for 300 epochs and applied a hybrid two-phase strategy. Specifically, AnimeDiffusion separates its training into (1) a classifier-free guidance pre-training stage focused on noise prediction, and (2) a fine-tuning stage based on image reconstruction loss. This second stage, which is absent from our current training, plays a crucial role in stabilizing color assignments and aligning generated outputs more closely with the reference image. Instead of continuing to rely on noisy inputs during generation, the fine-tuning phase inputs noise derived from the reference image itself, allowing deterministic sampling via DDIM and improved semantic correspondence, particularly in challenging regions like hair or eyes.

6 Conclusion and Future Work

Our approach successfully applies diffusion models and feature matching for anime sketch colorization. While Anime Diffusion [1] introduces a two-phase hybrid training strategy — with Phase 1 focused on denoising and learning structural correspondences, and Phase 2 dedicated to fine-tuning color fidelity through image reconstruction — our current implementation only includes the first phase. As a result, the absence of the second fine-tuning stage may limit the accuracy of the color transfer. Incorporating this second phase in future work could significantly enhance color consistency and overall visual quality.

Future work will include:

- Training on the full dataset for improved generalization.
- Implementing adaptive learning rate strategies.
- Exploring alternative loss functions for enhanced perceptual quality.
- Adding second fine-tuning to improve color fidelity.

References

[1] Y. Cao, X. Meng, P. Mok, X. Liu, T.-Y. Lee, and P. Li, "Animediffusion: Anime face line drawing colorization via diffusion models," *arXiv preprint arXiv:2303.11137*, 2023.

[2] Z. Liu, K. L. Cheng, X. Chen, J. Xiao, H. Ouyang, K. Zhu, Y. Liu, Y. Shen, Q. Chen, and P. Luo, "Manganinja: Line art colorization with precise reference following," *arXiv preprint arXiv:2501.08332*, 2025.

[3] X. Lu, X. Wang, and J. E. Fan, "Learning dense correspondences between photos and sketches," 2023. [Online]. Available: https://arxiv.org/abs/2307.12967

[4] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[5] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[6] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[7] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, "Emergent correspondence from image diffusion," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1363–1389, 2023.

[8] S. Koley, A. K. Bhunia, A. Sain, P. N. Chowdhury, T. Xiang, and Y.-Z. Song, "Text-to-image diffusion models are great sketch-photo matchmakers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 826–16 837.

[9] K. Gupta, V. Jampani, C. Esteves, A. Shrivastava, A. Makadia, N. Snavely, and A. Kar, "Asic: Aligning sparse in-the-wild image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4134–4145.

[10] J. Zhang, C. Herrmann, J. Hur, L. Polania Cabrera, V. Jampani, D. Sun, and M.-H. Yang, "A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence," *Advances in Neural Information Processing Systems*, vol. 36, pp. 45 533–45 547, 2023.

[11] M. O'Neill, "Tagged anime illustrations," 2019, dataset containing tagged anime illustrations from Danbooru. [Online]. Available: https://www.kaggle.com/datasets/mylesoneill/tagged-anime-illustrations/data

[12] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 277–280.

[13] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example." *Rendering techniques*, vol. 29, pp. 201–210, 2005.

[14] D. Sỳkora, J. Dingliana, and S. Collins, "Lazybrush: Flexible painting tool for hand-drawn cartoons," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 599–608.

[15] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073703

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[17] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," 2018. [Online]. Available: https://arxiv.org/abs/1807.06587

[18] T.-H. Sun, C.-H. Lai, S.-K. Wong, and Y.-S. Wang, "Adversarial colorization of icons based on contour and color conditions," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 683–691.

[19] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo, "Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence," 2020. [Online]. Available: https://arxiv.org/abs/2005.05207

[20] G. Lee, S. Shin, T. Na, and S. S. Woo, "Real-time user-guided adaptive colorization with vision transformer," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan 2024, pp. 473–482.

[21] Y. Ma, Z. He, J. Xiang, N. Zhang, and R. Pan, "Transformer-based line sketch colorization assisted by superpixel decomposition," *Available at SSRN 5026259*.

[22] D. Yan, L. Yuan, E. Wu, Y. Nishioka, I. Fujishiro, and S. Saito, "Colorizediffusion: Adjustable sketch colorization with reference image and text," 2024. [Online]. Available: https://arxiv.org/abs/2401.01456

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385