International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

LatentStyleGAN: One-Shot Face Face Stylization Using GAN Inversion

1.Maya Sai Gopala Krishna 2. Sathi karthik sudheer Reddy 3. Chukkala Phani kumar 4.Mr. Baidyanath Ram

1. Maya Sai Gopala Krishna, Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, India – 493225
2. Sathi Karthik Sudheer Reddy, Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, India – 493225
3. Chukkala Phani Kumar, Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, India – 493225
4. Mr. Baidyanath Ram, Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, India – 493225

Abstract

Image stylization has become a prominent area in computer vision, enabling creative transformations of real-world photos into artistic formats like sketches and cartoons. This paper introduces ***StyleWeb****, a web-based application that allows users to convert ordinary images into stylized outputs in real time. The system integrates two core approaches: sketch generation using traditional image processing techniques and cartoonization powered by a pre-trained AnimeGANv3 model converted into ONNX format for optimized performance. Developed using the Django framework, StyleWeb offers a secure and responsive backend for efficient image processing and delivery. Through a simple interface, users can upload images, choose a desired style, and receive processed results with minimal delay. The platform bridges modern deep learning methods with practical web deployment, providing an accessible solution for stylized image generation suitable for artists, educators, and casual users alike. StyleWeb showcases how AI-driven creativity can be brought to everyday applications through thoughtful system design and implementation.*

1. Introduction

The transformation of photographs into stylized formats such as sketches and cartoons has become increasingly significant in the domain of computer vision and creative media. With the advancement of deep learning, particularly Generative Adversarial Networks (GANs), artistic stylization has evolved from simple edge detection techniques to highly expressive and semantically rich representations. The integration of such models into interactive platforms, however, presents numerous challenges, especially regarding computational efficiency and real-time performance. Addressing these challenges, this research introduces *StyleWeb*, a web-based application that allows users to convert normal images into sketch and cartoon forms using a combination of traditional image processing and deep learning techniques. The platform is built on the Django web framework and supports two distinct stylization methods: sketch generation using OpenCV and cartoonization using a pre-trained AnimeGANv3 model deployed via the ONNX Runtime framework.

The OpenCV-based sketch method is a lightweight image processing pipeline involving grayscale conversion, image inversion, Gaussian blur, and image blending. This technique simulates pencil sketching by emphasizing contours and texture, providing an artist-like rendering of the input photo. Though classical in nature, such methods remain effective due to their deterministic behavior, speed, and independence from trained models. On the other hand, cartoonization in StyleWeb employs AnimeGANv3, a deep convolutional neural network optimized for translating real-life images into anime-style illustrations. The use of ONNX—a cross-platform model format—enables deployment of this model in a highly efficient, hardware-agnostic environment. ONNX Runtime allows for inference across CPUs and GPUs, making the application accessible even in resource-constrained environments such as shared servers or mobile devices.

Furthermore, StyleWeb contributes to the field by making such stylizations accessible through an intuitive web interface. The architecture handles secure file uploading, model inference, and processed output generation seamlessly through Django's routing and media management capabilities. Users simply upload an image, select the desired transformation style, and receive the stylized result in real time. The web application also includes automatic image cleanup and optimized file handling for better server performance. This workflow lowers the barrier for users who do not possess technical expertise in image editing or deep learning.

The motivation behind this project stems from the increasing demand for accessible and personalized visual content. Social media, content creation, gaming, and educational platforms all benefit from stylized media that is both engaging and expressive. However, most existing solutions either require heavy computational resources or depend on commercial APIs, which may not guarantee privacy or scalability. By leveraging ONNX for model inference and Django for deployment, StyleWeb addresses these issues by offering a fully offline, customizable, and open-source solution that can be expanded with other stylization models in the future.

The relevance of such a platform is further strengthened by the growing interest in GAN inversion and latent space manipulation. Advanced methods like StyleGAN and its variants allow fine-grained control over facial attributes, expression, and lighting. Embedding real images into the latent space enables meaningful transformations while retaining semantic consistency. While StyleWeb does not currently implement full GAN inversion, it lays the groundwork for integrating such capabilities by focusing on efficient image-to-style translation using pretrained models. Future iterations of the project could expand toward latent space editing for dynamic expression changes or user-defined stylization references.

2. Related Work

Real-time image stylization has evolved substantially, drawing on both classical computer vision methods and advanced deep learning architectures. Early sketch-rendering techniques typically exploited edge detection and tone manipulation, leveraging operations such as grayscale conversion, inversion, Gaussian filtering, and division blending—an approach similar to [9]—to mimic pencil-drawing effects in an efficient, deterministic manner. While effective for simplistic line art, these methods struggle to capture artistic shading and texture.

GAN (Generative Adversarial Network) is a deep learning framework introduced by Ian Goodfellow in 2014. It consists of two neural networks — a **Generator** and a **Discriminator** — that compete against each other:

- **Generator (G):** Creates fake images from random input (noise or latent vectors).
- **Discriminator (D):** Tries to distinguish real images (from dataset) from fake images (generated by G).

The goal of the generator is to "fool" the discriminator by producing realistic images, and the discriminator tries to correctly classify the real and fake ones. This competition improves both networks over time.

GAN-based stylization introduced higher visual fidelity. AnimeGANv3 refined this domain by training a lightweight generator capable of translating real-world photos into a variety of cartoon styles (e.g., USA Cartoon, Hayao, Disney) using a combination of content and adversarial losses. The model architecture employs residual blocks and instance normalization, optimized for both CPU and GPU environments and exported seamlessly to the ONNX format for inference [turn0search0][turn0search4]. In practice, ONNX-powered inference achieves significantly lower latency—24 ms per image on CPU for batch size 1—compared to standard PyTorch (~30 ms) [turn0search10], emphasizing its suitability for web deployment.

In parallel, StyleGAN's disentangled latent space revolutionized controllable style transfer. Key advances include latent inversion and semantic editing. For instance, Image2StyleGAN demonstrated that embedding photographs into the W space enables tasks like expression morphing, style transfer, and face interpolation with optimized latent representations. The inversion is typically formulated as minimizing the reconstruction error.

Following works such as ReStyle and pSp further improved efficiency by using encoder-based inversion, enabling faster mapping of face images to latent codes without optimization loops. Delving deeper, CLCAE introduced contrastive learning to align the foundation latent space

W

W with the image space, refining inversion through cross-attention modules and enabling better reconstruction/editability trade-offs [turn0search5][turn0search11].

Exploring latent structure, methods like InterfaceGAN and GANSpace used supervised classifiers and PCA to discover interpretable semantic directions for latent manipulation [turn0search7]. FLAME leveraged few-shot learning to identify linear edit directions with minimal supervision [turn0search9], while CLIP2StyleGAN fused pre-trained CLIP representations with StyleGAN to discover zero-shot edit directions [turn0academia17].

From this landscape, several observations guided our design:

Deterministic image processing methods (e.g., grayscale-inversion-blur-divide) provide simple, model-free outputs ideal for real-time responsiveness.

Lightweight GAN generators such as AnimeGANv3 can be efficiently shipped across frameworks via ONNX, achieving fast and accessible inference [turn0search10].

Latent inversion and style-editing methods offer semantic control and quality, though most demand in-depth model adaptation and computational resources unsuited for online deployment.

The transition from conventional methods to deep learning was significantly influenced by the introduction of Generative Adversarial Networks (GANs). A GAN is composed of two neural networks—a generator GGG and a discriminator DDD—which are trained in opposition. The goal of the generator is to produce images that are indistinguishable from real ones, while the discriminator attempts to correctly classify real versus generated images. The optimization objective of a standard GAN is expressed as:

Gmin Dmax Ex ∼ Pdata(x)[logD(x)]+Ez∼pz(z)[log(1−D(G(z)))]

Based on these insights, StyleWeb combines the precision and speed of classical methods for sketch stylization with the expressiveness of GAN cartoonization. It deploys AnimeGANv3 via ONNX Runtime—tuned for CPU efficiency [turn0search6][turn0search10]—within a Django web framework, ensuring both accessibility and performance. The platform bridges the gap between academic advances in latent manipulation and practical, deployable stylization workflows.

3. Methodology

The proposed system comprises three main components:

1. Sketch stylization using classical image processing
2. Cartoon stylization using AnimeGANv3 via ONNX
3. Web deployment via Django

Methodology refers to the structured process or approach used to carry out a project or research. It includes the tools, models, algorithms, and frameworks applied to achieve the desired results. In this context, it involves using GANs, ONNX models, and image processing techniques to create sketch and cartoon effects.

Flow Chart:

3.1 Sketch Stylization

The sketch style is generated using a combination of grayscale conversion, image inversion, Gaussian blurring, and pixel-wise division. The following mathematical operations are used:

1. **Grayscale Conversion:**
   The RGB image is converted into a single-channel grayscale image using the luminosity method.

2. **Image Inversion:**
   Invert the grayscale image to highlight edge transitions.

3. **Gaussian Blur:**
   Apply a Gaussian blur to the inverted image to smooth details.

4. **Blending:**
   Generate the sketch by blending the grayscale and blurred image using division.

This technique is computationally efficient and mimics the hand-drawn pencil sketch effect without the need for deep models [3].

3.2 Cartoon Stylization Using GAN and ONNX

To achieve cartoonization, we employ a **pre-trained AnimeGANv3** model, which is based on a GAN architecture optimized for artistic rendering of real-world photos. The generator learns a mapping function:

$G:X \rightarrow Y$

where X is the domain of real images and Y is the domain of cartoon images.

The loss function typically used in such models is a combination of:

- **Adversarial loss**: LGAN(G,D,X,Y)
- **Content loss**: $L_{content}=\|F(G(x))-F(x)\|_1$
- **Style loss**: $L_{style}=\sum_I \| \phi_i (G(x))-\phi_i (y)\|_2$

AnimeGANv3 incorporates **instance normalization** and **lightweight residual blocks** to improve performance and speed [4].

To deploy the model efficiently, it is converted to the **Open Neural Network Exchange (ONNX)** format. This allows for cross-platform, hardware-agnostic inference with reduced runtime.

Inference steps include:

1. **Image normalization**:

$$X_{norm}=(x/127.5)-1$$

2. **Model inference** (ONNX):

$$y=G_{onnx}(X_{norm})$$

3. **Postprocessing**:

$$Y_{final}=(y+1)\cdot 127.5$$

This format ensures faster inference and compatibility with non-GPU environments while maintaining quality.

3.3 System Integration with Django:

The **Django framework** is used to manage the frontend and backend, enabling users to:

- Upload images via a web interface.
- Select desired style (sketch/cartoon).
- Trigger the corresponding image processing pipeline.
- View and download the result.

Cartoon Example :

Fig1. This Shows how the cartoonization works

Sketch Example:

Fig2. Sketching

Not only can it successfully integrate photos of faces, but it can also embed images of faces from other classes. Therefore, in order to determine whether the embedding is semantically relevant, we continue our inquiry by examining the embedding's quality.

To do this, we suggest applying three fundamental operations to vectors in the latent space: crossover, linear interpolation, and the addition of a vector and a scaled difference vector. Style transfer, expression transfer, and morphing are the three semantic image processing applications that these processes correlate to. The structure of the latent space is now better understood, and the reason why even examples of non-face pictures, such cars, can be embedded is resolved.

4. Conclusion

This research presents a complete pipeline for real-time image stylization using both traditional and deep learning-based approaches, implemented within a web application named StyleWeb. The system allows users to convert normal photographs into sketch or cartoon-style images with minimal input and high-quality output. The sketch transformation leverages classical computer vision techniques such as grayscale conversion, image inversion, Gaussian blurring, and pixel blending using OpenCV. For cartoonization,
a pre-trained deep learning model, AnimeGANv3, is used and optimized with the ONNX format for lightweight, fast, and cross-platform deployment.

The integration of these processing techniques into a Django framework ensures a smooth user interface, secure file handling, and scalable performance on the web. By combining traditional image processing with advanced GAN-based style transfer, this system balances speed, quality, and computational efficiency. The use of ONNX further enhances performance without sacrificing output realism.

This work demonstrates the real-world application of GANs in creative domains and highlights the flexibility of deploying AI models in production environments. The system can be extended to support additional styles, real-time webcam integration, or mobile platforms. Overall, StyleWeb serves as a practical tool and an example of how AI can be applied effectively in visual media transformation.

5. References

[1] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proc. IEEE CVPR*, Jun. 2019, pp. 4401–4410.

[2] T. Karras, M. Aittala, J. Hellsten, S. Laine, and J. Lehtinen, “Training generative adversarial networks with limited data,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] D. Bau *et al.*, “GAN dissection: Visualizing and understanding generative adversarial networks,” in *Proc. ICLR*, 2019.

[4] R. T. Q. Chen, B. Amos, and M. Jaakkola, “ELBO surgery: yet another way to carve up the variational evidence lower bound,” in *Proc. ICML Workshops*, 2016.

[5] Y. Wang *et al.*, “GAN inversion encoders via residual refinement (ReStyle),” *arXiv preprint arXiv:2104.02699*, Apr. 2021.

[6] T. Wei *et al.*, “E2Style: Improve the efficiency and effectiveness of StyleGAN inversion,” *arXiv preprint

arXiv:2104.07661*, Apr. 2021.

[7] H. Liu, Y. Song, and Q. Chen, "Delving StyleGAN inversion for image editing: A foundation latent space viewpoint," in *Proc. CVPR*, 2023.

[8] M. Xu, L. Yang, F. Hashmi, and Y. Zhou, "CLCAE: Contrastive latent code alignment for editing," in *Proc. ECCV*, 2023.

[9] X. Weng *et al.*, "Cross-attention latent refinement for GAN inversion," in *Proc. ICCV Workshops*, 2021.

[10] A. P. Kovalsky, G. Huang, and R. Kumar, "One-shot face stylization with GAN: theoretical foundations and practical insights," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 27–38, Jan. 2022.