IJARETY



International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 13, Issue 6, November - December 2026

Impact Factor: 9.971

www.ijarety.in editor.ijarety@gmail.com

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 9.971|| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 6, November - December 2026 ||

DOI:10.15680/IJARETY.2026.1306003

Neural Style Transfer Using Adaptive Instance Normalization (AdaIN)

Prof. Amerah Zameer Ansari, Mayuri Suresh Pawar, Tanushree Prakash Bharote,

Shravani Narendra Nikam

Department of Information Technology, R. C. Patel Institute of Technology, Near Nimzari Naka, Shirpur, Maharashtra, India

ABSTRACT: Neural Style Transfer (NST) is a deep learning technique that combines the content of one image with the artistic appearance of another image. Traditional optimization-based methods can require considerable computation because the output image is refined iteratively. This paper presents an implementation-oriented approach to arbitrary neural style transfer using Adaptive Instance Normalization (AdaIN). A pretrained VGG-19 network extracts feature representations from the content and style images. The AdaIN operation adjusts the channel-wise mean and standard deviation of content features to match those of style features, and a decoder reconstructs the transformed representation into a stylized image. This architecture supports the use of different style images without training a separate model for each style. The paper describes the proposed workflow, system components, implementation technologies, testing considerations, applications, and limitations. The effectiveness of the implementation should be established through measured image-generation time and systematic visual evaluation; no unverified performance figures are claimed in this paper.

KEYWORDS: Neural Style Transfer, Adaptive Instance Normalization, VGG-19, Deep Learning, Image Processing, Artistic Image Generation.

I. INTRODUCTION

Neural Style Transfer is a computer vision technique that generates an image by combining the semantic content of one image with the visual characteristics of another. For example, a photograph may be transformed to resemble a painting while retaining the main objects and composition of the original photograph. This capability has applications in digital art, photography, graphic design, and creative media.

Early neural style transfer methods used convolutional neural network (CNN) features to represent content and style separately. Although these methods can produce visually rich results, optimization of each output image can be time-consuming. Feed-forward methods improved inference speed, but conventional models often required separate training for different styles. Adaptive Instance Normalization, introduced by Huang and Belongie, provides a flexible alternative by aligning the feature statistics of a content image with those of a style image.

This paper describes a VGG-19 encoder, AdaIN transformation, and decoder pipeline for arbitrary style transfer. The main objective is to present a practical system workflow that allows users to select content and style images and obtain a stylized output while retaining important content structure.

II. LITERATURE SURVEY

Research on neural style transfer has progressed from iterative image optimization to feed-forward networks and feature-statistics-based style adaptation. The following studies form the foundation for the proposed approach.

Paper / Authors	Review / Survey
Gatys, Ecker and Bethge, “A Neural Algorithm of Artistic Style” (2015)	Introduced an optimization-based approach that uses CNN feature activations for content and Gram matrices for style representation. It can produce detailed results but requires iterative optimization for each output.
Johnson, Alahi and Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution” (2016)	Used feed-forward transformation networks trained with perceptual losses to generate results efficiently.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 9.971|| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 6, November - December 2026 ||

DOI:10.15680/IJARETY.2026.1306003

	Conventional implementations may be tied to a particular trained style.
Ulyanov, Vedaldi and Lempitsky, “Instance Normalization: The Missing Ingredient for Fast Stylization” (2016)	Showed that instance normalization can improve feed-forward image stylization. This work helped establish normalization-based techniques in neural style transfer.
Huang and Belongie, “Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization” (2017)	Proposed AdaIN, which aligns channel-wise means and standard deviations of content features to those of style features, enabling arbitrary style transfer without training a separate model for each style.
Li et al., “Universal Style Transfer via Feature Transforms” (2017)	Presented feature transformation methods for transferring style information. The work provides a related direction for flexible style transfer using feature representations.

III. PROBLEM STATEMENT AND OBJECTIVES

Problem Statement: Optimization-based neural style transfer can require substantial processing time, while some feed-forward approaches are designed for predefined styles. A flexible implementation is needed to apply different artistic styles to content images without retraining a separate model for each style.
Objectives:

- To extract content and style representations using a pretrained VGG-19 encoder.
- To apply Adaptive Instance Normalization to align the statistical characteristics of content and style features.
- To reconstruct the transformed features using a decoder network.
- To provide a workflow for selecting input images and generating stylized outputs.
- To evaluate functionality, output appearance, and image-generation time using documented tests.

IV. PROPOSED METHODOLOGY

The proposed approach consists of image preprocessing, feature extraction, feature transformation, image reconstruction, and output display. First, the content image and style image are resized and converted into tensors suitable for the model. The pretrained VGG-19 encoder extracts features from both inputs. The AdaIN module transforms the content features using the channel-wise statistics of the style features. Finally, a decoder maps the transformed features back into image space.

The main operation is expressed as:
AdaIN(x, y) = σ(y) × [(x − μ(x)) / σ(x)] + μ(y)
Here, x represents the content feature map and y represents the style feature map. μ(x) and μ(y) denote channel-wise means, while σ(x) and σ(y) denote channel-wise standard deviations. The operation normalizes content features and then adjusts their statistics to match those of the style features.

The stylized output can be represented as:
I_out = g(AdaIN(f(I_content), f(I_style)))
In this expression, f is the pretrained VGG-19 encoder and g is the decoder. During decoder training, the encoder is kept fixed and the decoder is optimized using content and style objectives. A typical combined objective is L = L_content + λL_style, where λ controls the contribution of the style loss. The actual training configuration should be reported from the implementation.

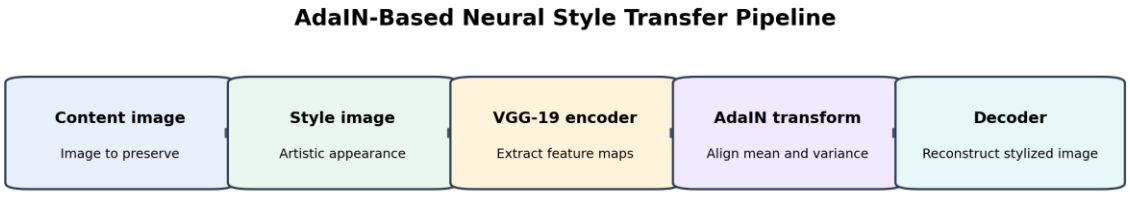


Fig. 1. Main processing pipeline of the proposed AdaIN-based approach.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 9.971|| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 6, November - December 2026 ||

DOI:10.15680/IJARETY.2026.1306003

V. SYSTEM ARCHITECTURE

The system architecture contains the following main stages:
• Input module: accepts a content image and an artistic style image.
• Preprocessing module: resizes images and converts them into the required numerical format.
• VGG-19 encoder: extracts feature maps from both input images.
• AdaIN module: aligns the channel-wise feature statistics of the content representation with those of the style representation.
• Decoder: reconstructs the transformed feature representation into the stylized image.
• Output module: displays the generated image and provides an option to save it.

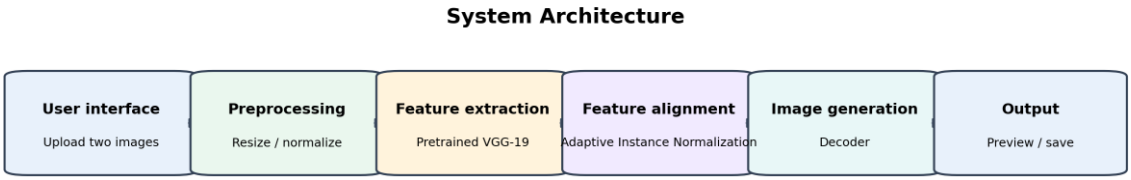


Fig. 2. System architecture from image upload to stylized output.

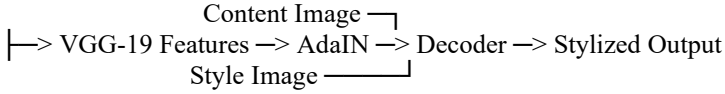


Fig. 1. Functional flow of the proposed AdaIN-based neural style transfer system.

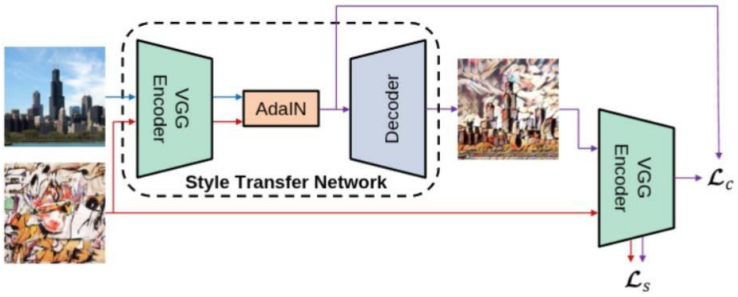


Fig. 3. An overview of our style transfer algorithm.

VI. SOFTWARE IMPLEMENTATION

The system can be implemented using Python and a deep learning framework such as PyTorch. VGG-19 is used as the feature encoder, while the AdaIN operation performs feature-statistics alignment. Image libraries such as PIL or OpenCV can support image loading and preprocessing. If a web interface is included, HTML, CSS, and JavaScript can provide the user interface, and Flask can connect the interface to the model.

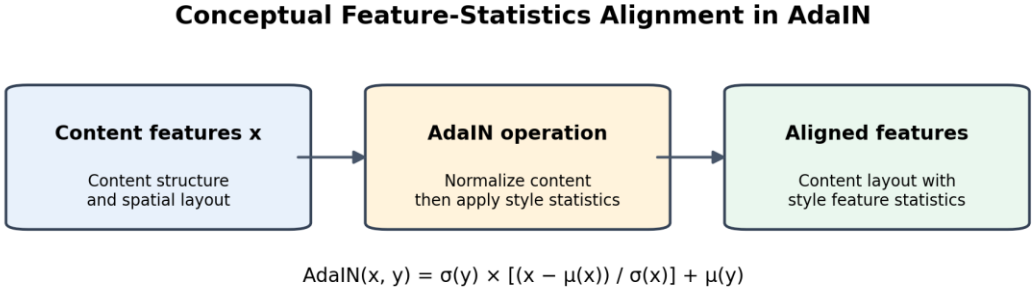


Fig. 4. Conceptual view of feature-statistics alignment performed by AdaIN.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 9.971|| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 6, November - December 2026 ||

DOI:10.15680/IJARETY.2026.1306003

VII. RESOURCES

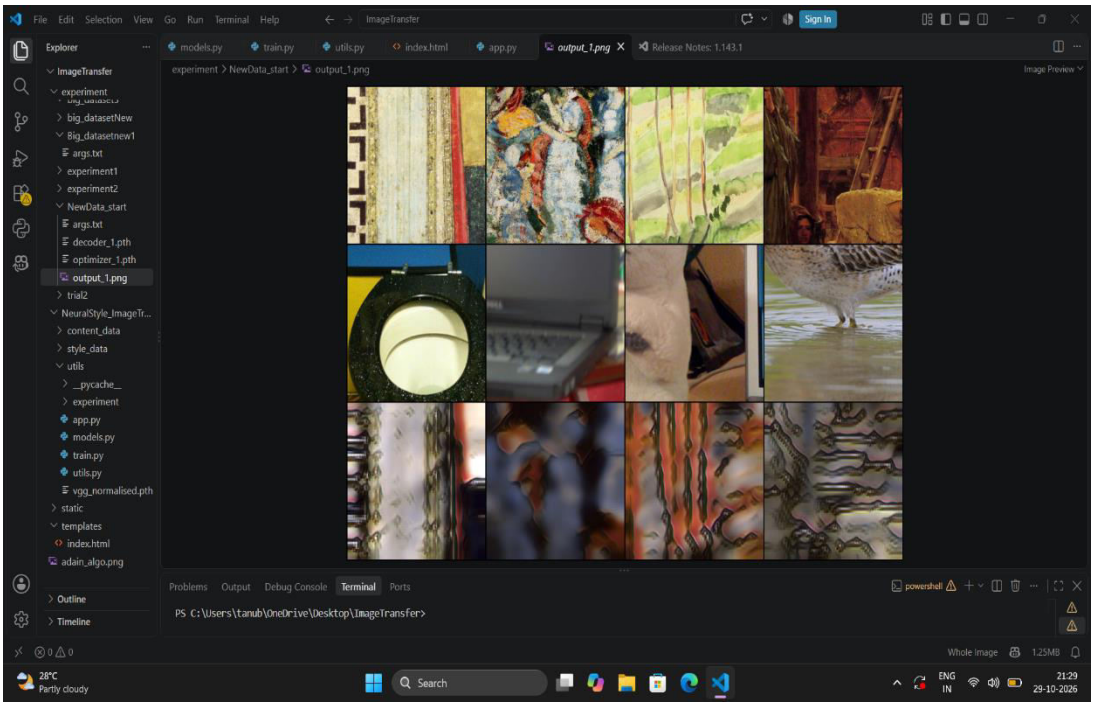
Component	Purpose
Python / PyTorch	Model implementation and tensor operations
Pretrained VGG-19	Extraction of content and style features
AdaIN module	Alignment of content and style feature statistics
Decoder network	Reconstruction of the stylized image
PIL / OpenCV	Image loading, resizing, and preprocessing
HTML, CSS, JavaScript / Flask (if implemented)	User interface and communication between interface and model

VIII. TESTING AND PROJECT OUTCOME

Testing should verify that the system accepts valid input images, processes the content and style features, generates an output image, and handles invalid inputs appropriately. Functional testing can cover image upload, preprocessing, style transformation, output display, and saving. Performance testing can record image-generation time and memory usage under a consistent setup. Visual evaluation should examine content preservation, style similarity, visible artifacts, and output consistency across different image pairs.

The intended project outcome is an image stylization workflow in which a user supplies a content image and a style image and receives a transformed image. Since quantitative performance results depend on actual execution, generation-time values and claims of accuracy or improvement should only be included after testing. A result table can be completed using measurements from the implemented system.

Test	What to verify	Result to record
Input validation	Valid/invalid image upload	Pass/Fail after test
Style transfer	Style characteristics appear in output	Visual observation
Content preservation	Main objects and structure remain recognizable	Visual observation
Processing performance	Generation time for documented image size	Measured seconds
Output handling	Display and save generated image	Pass/Fail after test



| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 9.971|| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 6, November - December 2026 ||

DOI:10.15680/IJARETY.2026.1306003

IX. APPLICATIONS, ADVANTAGES AND LIMITATIONS

Applications: The approach can support digital art creation, photography filters, graphic design, social-media content creation, creative education, and web-based image editing.

Advantages: AdaIN supports arbitrary style images; the encoder-decoder pipeline can generate outputs through a forward pass after training; and the approach reduces the need for a separately trained model for every style.

Limitations: Fine details may be lost, some styles may introduce artifacts, and output quality depends on the encoder, decoder, and input images. Processing high-resolution images can also require additional memory and computation. Feature-statistics matching does not guarantee exact reproduction of every artistic detail.

IX. CONCLUSION AND FUTURE SCOPE

This paper describes a neural style transfer approach based on Adaptive Instance Normalization. A pretrained VGG-19 encoder extracts feature representations, AdaIN aligns content-feature statistics with those of a style image, and a decoder reconstructs the stylized result. The method provides a flexible foundation for applying different artistic styles without training a separate model for each style. Further work can include systematic benchmarking, user-controlled style strength, blending multiple styles, video style transfer, and optimization for mobile or real-time applications. Experimental claims should be supported by measured results from the implemented system.

REFERENCES

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style,” arXiv:1508.06576, 2015.
[2] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” arXiv:1603.08155, 2016.
[3] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance Normalization: The Missing Ingredient for Fast Stylization,” arXiv:1607.08022, 2016.
[4] X. Huang and S. Belongie, “Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1501–1510.
[5] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal Style Transfer via Feature Transforms,” in Advances in Neural Information Processing Systems (NeurIPS), 2017.
[6] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2414–2423.
[7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” International Conference on Learning Representations (ICLR), 2015.
[8] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Texture Networks: Feed-forward Synthesis of Textures and Stylized Images,” Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016.
[9] L. Sheng, Z. Lin, J. Shao, and X. Wang, “Avatar-Net: Multi-Scale Zero-Shot Style Transfer by Feature Decoration,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[10] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, “Controlling Perceptual Factors in Neural Style Transfer,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3985–3993.
[11] P. Chandran, G. Zoss, P. Gotardo, M. Gross, and D. Bradley, “Adaptive Convolutions for Structure-Aware Style Transfer,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 7972–7981.
[12] “LPAdaIN: Light Progressive Attention Adaptive Instance Normalization Model for Style Transfer,” Electronics, vol. 11, no. 18, article 2929, 2022.
[13] M. Ioannou et al., “Evaluation in Neural Style Transfer: A Review,” Computer Graphics Forum, 2024.
[14] A. Roy Gosthipaty and R. Raha, “Neural Style Transfer with AdaIN,” Keras Code Examples, 2021. Available: https://keras.io/examples/generative/adain/.
[15] “Style Transfer Review: Traditional Machine Learning to Deep Learning,” Information, vol. 16, no. 2, article 157, 2025.





International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 **Impact Factor: 9.971**

www.ijarety.in editor.ijarety@gmail.com