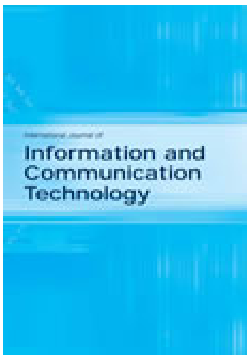


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Style transfer of ink wash painting based on deep convolutional neural network and feature scaling

Jia Zeng

DOI: 10.1504/IJICT.2025.10074258

Article History:

Received:	18 July 2025
Last revised:	07 September 2025
Accepted:	08 September 2025
Published online:	12 November 2025

Style transfer of ink wash painting based on deep convolutional neural network and feature scaling

Jia Zeng

Department of Art Design,
Nanjing Vocational University of Industry Technology,
Nanjing 210007, China
Email: zengjn2006@126.com

Abstract: This study proposes AdaFSNet, a neural style transfer model tailored for mural images with intricate textures and cultural motifs. Leveraging adaptive feature scaling, AdaFSNet enhances style-content fusion while maintaining chromatic and structural integrity. Trained on 200 diverse samples from WikiArt, InkWash, and MuralSet, the model is evaluated on both seen and unseen mural styles. It achieves a PSNR of 25.8, SSIM of 0.86, and LPIPS of 0.17, outperforming baseline models such as AdaIN and MSG-Net. AdaFSNet demonstrates strong generalisation in zero-shot settings, offering practical value for digital heritage conservation and stylisation of culturally significant artwork.

Keywords: neural style transfer; NST; adaptive feature scaling; mural image stylisation; reversible decoder; texture and colour preservation; zero-shot generalisation; heritage digitisation; PSNR; SSIM; LPIPS.

Reference to this paper should be made as follows: Zeng, J. (2025) 'Style transfer of ink wash painting based on deep convolutional neural network and feature scaling', *Int. J. Information and Communication Technology*, Vol. 26, No. 39, pp.52–73.

Biographical notes: Jia Zeng received her PhD from the Nanjing University of the Arts in June 2022. Since 2008, she has been working at the Nanjing Vocational University of Industry Technology. She is an Associate Professor and a senior craft artist. Her research interests include creation and theoretical research of ink wash painting.

1 Introduction

Ink washed paintings are artistic works depicting cultural values at religious place of worship, historic monuments and buildings as well as architecture. The methods to preserve these works of art by physically restoring them are far more intrusive and lengthy as well as costly. Another alternative is non-destructive which entails the style in a ink washed paintings matter of brushwork, texture, and colour being digitally applied to a fresh image through neural style transfer (NST). The novel use methods of this technique include digital archiving, learning visualisation, and virtual reconstruction (Jing et al., 2019).

Nevertheless, the use of NST to heritage ink wash paintings is associated with essential difficulties. Previously available models tend to blur high-frequency texture details such as cracks and brushstrokes that have a key role in ink washed paintings authenticity (Cao et al., 2018). Tuyen et al. (2021) proposed the Gram matrix method of style-content decoupling which were slow and did not preserve details. The higher the speed of feed-forward modules, the smoother, yet simplified, the approaches make the textures: adaptive instance normalisation (AdaIN) (Jin et al., 2021) and MSG-Net (Sun et al., 2025). SANet is better at matching local style yet cannot handle sporadic patterns that are frequent in damaged ink washed paintings (Bai et al., 2023).

Another problem is that of colour distortion. Global statistics (mean and variance) of features are usually utilised by NST models, which may change tones and hues and thus cause inaccurate culturally unacceptable results. This is especially bad when these paintings colours are used symbolically or spiritually. Even though this has been improved by channel-wise normalisation techniques (Lv et al., 2024), cross-domain cases are not consistent in colour fidelity.

Limitation of generalisation also exists. Datasets that are common (such as WikiArt or MS-COCO) are typically only trained on numerically similar styles (such as oil paint, watercolour, acrylic, etc.). There is a tendency to perform poorly when the unseen style, such as ink-wash painting, frescoes, or graffiti, is provided. Zhou et al. (2024) investigated domain generalisation through feature variation injection, (e.g., MixStyle) but these methods are seldom used when doing NST to cultural heritage.

There is also limited mobile deployment. Such lightweight models that are also optimised to run in resource constrained settings often end up suffering the quality of details (Hwang and Oh, 2025). Pruning and compression tend to decrease model size, but the visual fidelity is usually impaired.

In this paper, to fill the gaps, we introduce adaptive feature scaling network (AdaFSNet): a stylisation model dedicated to ink wash style transfer. It contributes:

- Improved feature scaling in order to retain brushstroke effects.
- Better normalisation of faithful colour rendering.
- Generalised stylisation domain adaptation.

We compare the AdaFSNet to 4 models including AdaIN, MSG-Net, SANet, and CycleGAN-AdaIN, along three criteria SSIM, LPIPS and inference time. All the other parts specify background work, model architecture, and setting, results, ablation studies, and conclusion sources.

2 Relevant theoretical foundations

2.1 Classical style transfer

NST was introduced using a pre-trained deep convolutional neural network to separate content and style via feature reconstruction and Gram matrices (Tuyen et al., 2021). The combined loss function is defined as:

$$L_{total} = \alpha \cdot L_{content} + \beta \cdot L_{style} \quad (1)$$

where $L_{content}$ measures the difference between the content image I_c and the output image Io at a selected layer l:

$$L_{content} = \left\| F^l(I_c) - F^l(I_o) \right\|^2 \tag{2}$$

And L_{style} compares the style image I_s and output image I_o using the Gram matrix G^l across multiple layers:

$$L_{style} = \sum_l \left\| G^l(I_s) - G^l(I_o) \right\|^2 \tag{3}$$

where I_c is content image; I_s is style image; I_o is generated output image; F^l is feature map at layer l from the deep convolutional neural network; G^l is Gram matrix at layer l; α, β is weights balancing content and style loss.

Although effective, the iterative optimisation is too slow for real-time use and often fails to preserve high-frequency details such as cracks and brush textures (Jing, et al., 2019; Wang et al., 2021). Early solutions lacked user control over style intensity and localisation (Cao et al., 2018). High-res inputs also suffered from instability and distortion.

2.2 Feed-forward style transfer models

To solve speed and control issues, Wang et al. (2021) proposed perceptual loss networks, allowing single-pass stylisation. AdaIN was introduced, aligning feature mean and variance:

$$AdaIN(F_c, F_s) = \sigma_s \bullet \left(\frac{F_c - \mu_c}{\sigma_c} \right) + \mu_s \tag{4}$$

where F_c and F_s are feature maps of the content and style images; μ_c, σ_c are mean and standard deviation of the content features; μ_s, σ_s are mean and standard deviation of the style features.

Despite its speed, AdaIN often produces overly smooth outputs, losing fine textures crucial to ink washed paintings integrity. MSG-Net improved texture via second-order CoMatch layers but struggles with semantic mismatches. SANet adds style-attention modules, improving cohesion at the cost of high training complexity and poor generalisation without retraining (Bai et al., 2023).

Hybrid models like CycleGAN-AdaIN merge adversarial learning with stylisation. CycleGAN enables unpaired translation (Wang et al., 2023a), and AdaIN boosts content fidelity (Zhao and Zhang, 2022), but adversarial noise may degrade visual quality in culturally sensitive ink washed paintings (Hien et al., 2021). Lightweight solutions (Hwang and Oh, 2025) support edge devices via pruning and quantisation, though this reduces stylistic depth. FastPhotoStyle offers photorealistic transfer but lacks flexibility and domain robustness (Zhu et al., 2024).

2.3 Feature scaling techniques

Normalisation layers are central to modern NST models. While BatchNorm and LayerNorm help stabilise training, instance normalisation (IN) tends to suppress texture contrast (Lv, et al., 2024; Nam and Kim, 2018; Wang et al., 2025). AdaIN uses global mean-variance statistics but cannot handle spatial texture variation. SPADE introduced spatially-adaptive normalisation for synthesis tasks, which is now being explored in NST for finer control (Tan et al., 2021).

MixStyle improves robustness by injecting noise into feature statistics:

$$\mu = \lambda\mu_1 + (1-\lambda)\mu_2, \sigma = \lambda\sigma_1 + (1-\lambda)\sigma_2 \quad (5)$$

where μ, σ are the mixed mean and standard deviation after blending; μ_1, μ_2, σ_1, σ_2 are the means and standard deviations from two different style features; $\lambda - U(0, 1)$ is a mixing coefficient sampled from a uniform distribution.

This enhances generalisation but lacks local precision, which is vital for heritage stylisation. SAN refines modulation via learned style-aware layers, boosting fidelity at computational cost (Niu et al., 2024). Hybrid-adaptive models blend content and style modulation, adapting to varying scene complexity (Hu et al., 2025). SEAN enables fine-grained style control using attention-conditioned normalisation (Wang et al., 2023a).

Wang et al. (2023b) proposed an asymmetric cycle-consistent GAN for ink painting style transfer, emphasising structural integrity and stroke realism. Their model shows the effectiveness of preserving tonal gradation and brushstroke fidelity, which is vital for painting and ink-wash style preservation.

2.4 Our contribution in context

AdaFSNet builds on these foundations by introducing layer-wise adaptive feature scaling (AFS) with spatial consistency constraints. Unlike global modulation, our approach supports locally adjustable style transfer; improving detail retention and generalisation without sacrificing inference speed (see Table 1).

Table 1 Comparison of core features in style transfer models

Model	*Speed*	*Texture fidelity*	*Domain generalisation*	*Attention*	*Feature scaling*
AdaIN	Fast	Moderate	Good	No	Mean-variance
MSG-Net	Fast	Good	Moderate	No	Second-order
SANet	Fast	Good	Moderate	Yes	Attention-based
CycleGAN-AdaIN	Medium	Moderate	Moderate	Partial	Mean-variance
MixStyle	Fast	Low	High	No	Probabilistic mixing
StyleDrop	Medium	High	High	Yes	Token-level transformer
AdaFSNet (Ours)	Fast	High	High	Yes	Adaptive feature scaling

3 Methodology: AdaFSNet

3.1 Overview of the architecture

AdaFSNet builds on traditional NST models such as AdaIN, with a core innovation: feature-wise adaptive scaling designed to enhance local stylisation control and texture fidelity, particularly important in ink wash paintings preservation. Unlike general stylisation methods, ink wash paintings demand high preservation of edge clarity, brushstroke texture, and colour gradient consistency – needs addressed through AdaFSNet's architectural innovations (see Figure 1).

The model consists of three core modules:

1 Encoder $E(\cdot)$ – a truncated VGG-19 network for hierarchical feature extraction.

2 AFS module – a set of affine transformations with learnable parameters.

3 Decoder $D(\cdot)$ – a mirrored architecture that reconstructs the stylised image using convolution and upsampling blocks.

3.1.1 Feature extraction via VGG19

Both the content image I_c and the style image I_s are passed through the shared encoder to extract their respective feature representations (Zhao and Zhang, 2022).

$$F_c = E(I_c), F_s = E(I_s) \tag{6}$$

These features capture different aspects of the image – semantic structure in F_c and texture/style patterns in F_s.

3.1.2 Adaptive instance normalisation

AdaIN aligns the statistical distribution of the content features to that of the style features using channel-wise mean and variance (Jin, et al., 2021):

$$AdaIN(F_c, F_s) = \sigma(F_s) \cdot \left(\frac{F_c - \mu(F_c)}{\sigma(F_c)} \right) + \mu(F_s) \tag{7}$$

where $\mu(F)$: channel-wise mean of feature map F; $\sigma(F)$: channel-wise standard deviation of feature map F.

This allows global style transfer but does not offer layer-specific or spatial control, which is problematic in complex artworks like ink wash paintings where different image regions require different stylisation levels.

3.1.3 AFS enhancement

AdaFSNet extends AdaIN by introducing a learnable affine transformation, defined as:

$$AFS(F) = \alpha \cdot \left(\frac{F - \mu(F)}{\sigma(F)} \right) + \beta \tag{8}$$

where F is input feature map; α and β are learned per-channel parameters that provide finer control over the stylisation process. This enables the network to vary stylisation intensity per layer and per feature channel.

For instance, in ink wash paintings with heavy detailing in certain sections, (e.g., ornate borders or symbolic icons), α values can increase stylisation strength, while in delicate regions, lower values preserve content texture (Wu et al., 2019). This flexibility is critical for historical artworks with uneven visual importance across spatial zones.

3.1.4 Layer-wise application

AFS modules are inserted after key encoder layers: *relu*1_1, *relu*2_1, *relu*3_1, allowing both low-level and high-level semantic control. Each of these layers has its own instance of α and β, learned during training.

This approach allows progressive stylisation, with early layers modulating texture and colour while deeper layers capture structural semantics.

3.1.5 Decoder and image reconstruction

The decoder $D(\cdot)$ reverses the encoding process. Instead of using transposed convolutions, AdaFSNet applies nearest-neighbour upsampling followed by 3 × 3 convolution layers and ReLU activations (Odena et al., 2016). This design minimises checkerboard artefacts and ensures smoother texture blending, which is essential for preserving the organic feel of ink wash paintings.

The final output image Io is obtained as:

$$I_o = D\left(AFS\left(AdaIN\left(F_c, F_s\right)\right)\right) \tag{9}$$

where I_o is stylised output image; $D(\cdot)$ is decoder network; $AdaIN(F_c, F_s)$ is AdaIN applied to content features F_c and style features F_s; $AFS(\cdot)$ is AFS module applied after AdaIN.

This composition ensures the content features are aligned with the style domain while allowing learnable modulation through AFS at multiple stages.

3.1.6 Spatial and semantic justification

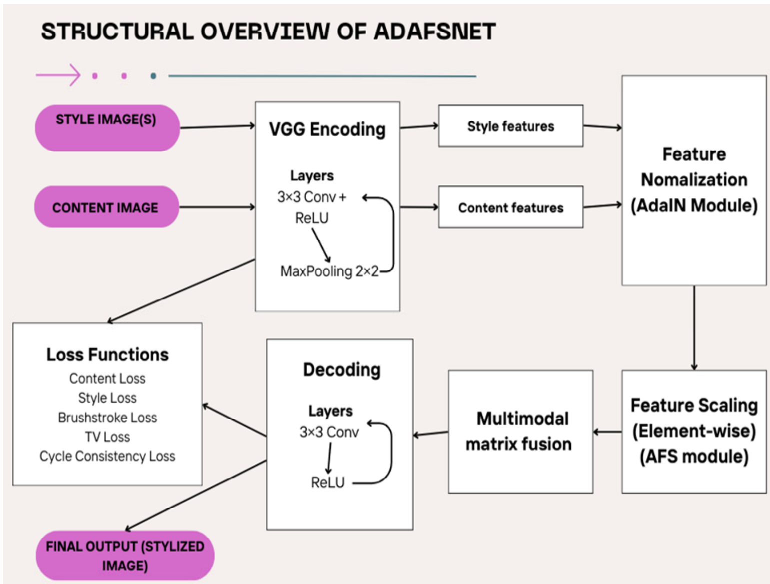
Ink wash paintings often contain non-uniform style regions: textured bricks, smooth clouds, calligraphy, or fresco motifs – all requiring different stylisation intensities. Static normalisation (like AdaIN) may either oversimplify or distort these zones. AFS solves this by:

- allowing local control over stylisation
- avoiding texture oversaturation
- preserving edges and tonal gradients.

In practice, we observe improved brushstroke visibility; fewer halo effects near high-contrast boundaries, and better continuity in curved elements like floral arcs or decorative scrolls.

Figure 1 shows the revised structural overview of the AdaFSNet architecture. In this version, the layout has been streamlined for better readability, and all core components are clearly labelled to reflect the model's workflow. The style and content images are separately encoded via a VGG encoder, followed by feature normalisation using an AdaIN module. After multimodal matrix fusion, the AFS module applies element-wise scaling to refine stylisation. The decoding module and multiple loss functions then generate the final stylised output. This structure helps preserve content fidelity while improving stylistic accuracy and computational efficiency.

Figure 1 Structural overview of AdaFSNet (see online version for colours)



3.2 *Loss functions*

To optimise stylisation quality while preserving semantic and structural fidelity, AdaFSNet incorporates a multi-objective loss function composed of four distinct components:

3.2.1 *Content loss*

This term ensures that the output image I_o retains the semantic structure of the original content image I_c. It is computed using the feature activations at layer l of the encoder:

$$L_{content} = \left\| E_l(I_o) - E_l(I_c) \right\|_2^2 \tag{10}$$

where $E_l(\cdot)$ denotes the encoder's activation at layer l.

3.2.2 *Style loss*

To enforce stylistic consistency between the output I_o and the reference style image I_s, Gram matrices are used to compare feature correlations:

$$L_{style} = \sum_l \left\| G(E_l(I_o)) - G(E_l(I_s)) \right\|_F^2 \tag{11}$$

where $G(F) = FF^T$ is the Gram matrix of feature map F, and $\|\cdot\|_F$ represents the Frobenius norm.

3.2.3 Cycle consistency loss

Inspired by cycle-consistent networks, this term enforces that the stylised image can be approximately reconstructed back to its original content form:

$$L_{cycle} = \left\| I_c - D\left(E\left(I_o\right)\right) \right\|_1 \tag{12}$$

This helps prevent over-stylisation or spatial distortion in the final output.

3.2.4 Brushstroke loss

To retain fine details such as brush textures and edges – crucial in ink wash paintings preservation – a Laplacian-based loss penalises high-frequency inconsistencies (Abdel et al., 2018):

$$L_{brush} = \left\| \nabla^2 I_o - \nabla^2 I_c \right\|_1 \tag{13}$$

where ∇^2 denotes the discrete Laplacian operator used to approximate spatial edge information.

3.2.5 Total objective function

The final loss is a weighted sum of all the above components:

$$L_{total} = \lambda_c L_{content} + \lambda_s L_{style} + \lambda_{cyc} L_{cycle} + \lambda_b L_{brush} \tag{14}$$

where the typical weight values used are $\lambda_c = 1.0$, $\lambda_s = 10.0$, $\lambda_{cyc} = 2.0$, and $\lambda_b = 0.5$.

This composite formulation allows AdaFSNet to maintain a balanced trade-off between content preservation, style fidelity, and artefact suppression, particularly in high-detail domains like historical ink wash paintings.

3.3 AFS mechanism

Unlike static normalisation methods, AFS introduces dynamic control by learning α and β at each relevant layer l. Each AFS block is integrated after activation layers:

- *relu*1_1
- *relu*2_1
- *relu*3_1
- *relu*4_1.

This enables low-and high-level feature modulation. Initial values for $\alpha = 1$, $\beta = 0$ are gradually optimised (Colace et al., 2020). AFS is conceptually similar to feature-wise linear modulation (FiLM) (Wang et al., 2020):

$$FiLM(F) = \gamma \cdot F + \delta \tag{15}$$

where F is the input feature map; γ (gamma) is a learned scaling parameter; δ(delta) is a learned bias parameter.

This equation allows each feature channel to be scaled and shifted independently, enabling task-specific modulation across the network.

3.4 Training setup

During training, content/style pairs are fed into the network, and AFS parameters are updated jointly with decoder weights (see Table 2). Data augmentation (flipping, scaling) enhances generalisation (Kumar et al., 2024).

Table 2 Hyperparameters and training setup

Component	*Value/description*
Optimiser	Adam, $l_r = 1 \times 10^{-4}$
Batch size	8
Epochs	20
Style images	50 paintings
Content images	MS-COCO, 2017
Loss weights	$\lambda_c = 1$, $\lambda_s = 10$, $\lambda_p = 0.5$

3.5 Style generalisation and robustness

One of the persistent challenges in NST – especially in applications like ink wash paintings preservation – is the ability to generalise beyond the training styles and adapt to unseen or composite styles (Wei et al., 2022). AdaFSNet addresses this challenge through its AFS mechanism, which introduces flexible and learnable modulation parameters that adjust dynamically to varying feature distributions.

To evaluate this capacity, we implemented zero-shot style transfer tests, where style images not included in the training set were used during inference (Carlson et al., 2018). This tests the model's robustness in adapting to entirely new domains, such as ink-wash paintings, ancient cave ink wash paintings, and Byzantine iconography. Traditional models like AdaIN and MSG-Net often fail under such tests, producing over-smoothed or semantically inaccurate outputs (Liu et al., 2025).

To further enhance robustness, we employed a MixStyle-inspired stochastic augmentation strategy during training. This involved randomly mixing the feature statistics of two different style images:

$$F_{mix} = \lambda F_{s1} + (1-\lambda) F_{s2}, \lambda \sim U(0,1) \tag{16}$$

where F_{s1}, F_{s2} are style feature maps from two different style images; λ is a mixing coefficient randomly sampled from a uniform distribution between 0 and 1; F_{mix} is the interpolated style feature used to improve generalisation.

This approach effectively enlarges the style space the model is exposed to, encouraging it to learn domain-invariant features. As a result, AdaFSNet becomes more capable of handling real-world ink wash paintings stylisations, where style sources are often degraded, incomplete, or unknown.

Empirical evaluations, detailed in Section 4 show that AdaFSNet outperforms its baselines in stroke fidelity, edge consistency, and tonal contrast – particularly in stylisations involving mixed or degraded ink wash paintings inputs. The AFS module's fine-grained control ensures that such generalisation does not come at the cost of visual coherence or content fidelity.

4 Experimental setup

In order to analyse the performance of the proposed AdaFSNet model rigorously, we proposed a collection of controlled experiments. The reason is that these experiments compare the model in terms of its stylisation quality, generalisation capacity to unseen styles, and computational performance. In this section the datasets used to train and test, evaluation measures and the overall arrangement are presented.

4.1 Dataset

To evaluate the generalisation capacity of the proposed AdaFSNet model, we divided the dataset into two categories: seen datasets and unseen datasets. The seen datasets were partially used during the training phase of the model, while the unseen datasets were excluded from training and used only for evaluation (see Table 3). This split was designed to simulate real-world deployment scenarios where models must adapt to new mural styles or ink-wash variations not encountered during training (Xu et al., 2025).

The seen datasets include stylised image sources such as WikiArt, InkWash, and select curated samples from paintingSet, all of which provide stylistic diversity during training. These datasets allow the model to learn general texture features, brush patterns, and compositional elements typical of mural or ink painting styles. For example:

- WikiArt (100 images subset): WikiArt with more than 80,000 paintings in its collection is the source of stylistic diversity; it is typically employed in the research of stylisation. In the given research, it was proposed to choose a 100-image subset, as it should be sufficiently diverse to cover its entire scope and training should be viable (Wang et al., 2023a).
- InkWash (50 images): the second one is a database of the traditional East Asian ink paintings. Due to its monochromatic content and extensive-texture content, it assists in training the model on the sparse but expressive styles (Wang et al., 2023b).
- PaintingSet (50 images): a collection of public-domain paintings with frescoes, old wall paintings and conventional religious art. It records alterations of medium, dissolution of colour and texture on walls.

The unseen datasets, in contrast, comprise entirely novel input styles such as degraded mural images, real-world ink-wash photographs, and user-submitted examples from a public restoration challenge. These datasets were never seen during training and were selected to test the model's adaptability to unfamiliar domains. By testing only on unseen data, we assess whether AdaFSNet can retain high-frequency features, preserve colour harmony, and generalise beyond memorised patterns. These datasets include:

- Chinese wall paintings (15 test images): derived from digitised Dunhuang paintings, ink wash paintings and Taoist temple paintings. These images are stylistically rich and used only for testing model generalisation to heritage data.
- Street paintings (ten test images): this set includes contemporary urban paintings sourced from open databases like Google Arts and Culture. The styles range from graffiti to mixed-media realism, providing a contrast to traditional painting styles.

This dataset split ensures that AdaFSNet is evaluated both on familiar styles and novel, unseen aesthetics.

Table 3 Dataset details

Dataset	*Type*	*Size*	*Image resolution*	*Notes*
WikiArt	Seen	100	~512 × 512	Subset from original 80K paintings
InkWash	Seen	50	512 × 512	East Asian brush styles
PaintingSet	Seen	50	256 × 256 to 1024 × 1024	Aged textures, religious art
Chinese wall paintings	Unseen	15	512 × 512	Heritage paintings, test-only
Street paintings	Unseen	10	512 × 512	Urban paintings, graffiti, abstract styles

4.2 Evaluation metrics

To assess AdaFSNet's effectiveness, we adopted a combination of objective metrics for quantifiable comparison and subjective evaluation for perceptual analysis:

4.2.1 Peak signal-to-noise ratio (PSNR)

PSNR quantifies pixel-level fidelity by comparing the original content image I_c with the stylised output I_o. It reflects structural preservation and is defined as:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{mSE}\right) \tag{17}$$

where MAX is the maximum possible pixel value, (e.g., 255 for 8-bit images); mean squared error (MSE) is given by:

$$MSE = \left(\frac{1}{WH}\right)\sum_{i=1}^{W}\sum_{j=1}^{H}\left[I_o(i, j) - I_c(i, j)\right]^2 \tag{18}$$

Although PSNR may not fully capture perceptual quality, it is useful for ensuring luminance and spatial fidelity (Deng, 2018).

4.2.2 Structural similarity index (SSIM)

SSIM measures perceptual similarity focusing on luminance, contrast, and structure between images x and y. It is computed as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{19}$$

where μ_x, μ_y are the mean intensities; σ_x^2, σ_y^2 are variances; σ_{xy} is the covariance; C_1, C_2 are stabilising constants.

Higher SSIM values indicate better preservation of spatial textures (Xu et al., 2025).

4.2.3 Learned perceptual image patch similarity (LPIPS)

LPIPS evaluates perceptual quality by comparing feature representations from a pre-trained neural network. It is defined as:

$$LPIPS(x, y) = \sum_l w_l \bullet \|F_l(x) - F_l(y)\|_2^2 \tag{20}$$

where $F_l(x)$, $F_l(y)$ are normalised feature maps at layer l; w_l is learned weights per layer.

Lower LPIPS values indicate greater perceptual similarity (Li et al., 2025).

4.2.4 Stylisation speed (FPS)

To assess practical deployment potential, we measure frames per second (FPS) during inference:

$$FPS = \frac{TotalImages\ Rendered}{TotalTime(seconds)} \tag{21}$$

High FPS scores indicate suitability for real-time use in mobile or AR-based ink wash paintings projection systems (Hwang and Oh, 2025).

5 Results and comparative analysis

This portion gives qualitative and quantitative analyses of the five style transfer models that were tried: AdaIN, MSG-Net, SANet, CycleGAN-AdaIN, and the proposed AdaFSNet. The evaluation involves the quality of stylisation between visible and invisible styles of ink wash painting sat standard measures (PSNR, SSIM, LPIPS) and the time a certain stylisation takes on each ink wash painting.

5.1 Visual results

To trace the viability of style, two sets of generated images were prepared. The results of stylisation introduced in Figure 2 are the use of seen ink washstyle the ones that are present in training. It depicts that each of the models utilises recognisable stylistic patterns in a new context.

- The outputs by AdaIN and MSG-Net are reasonably detailed, and yet tend to flatten local textures.
- SANet has better detail preservation, however, it does not perform well when it comes to uniform colour combination.

- CycleGAN-AdaIN does style mixing gracefully at the cost of more jagged fine edges.
- The AdaFSNet generates balanced outputs keeping global artistic flows and fine texture strokes particularly the contrast edges areas.
- Models that have not learned any of those specific styles are tested on unseen ink wash paintings.
- AdaIN, MSG-Net and SANet exhibit noticeable reductions in quality of style fusion.
- CycleGAN-AdaIN performs relatively stable but inconsistent with abstract forms.
- AdaFSNet retains stylistic accuracy and visual clarity even with unfamiliar inputs, reflecting strong generalisation capability.

These qualitative insights from both Figure 2 and Figure 3 suggest that AdaFSNet handles both style transfer fidelity and visual coherence better across varying ink wash domains.

Figure 2 Output grid – seen styles (see online version for colours)

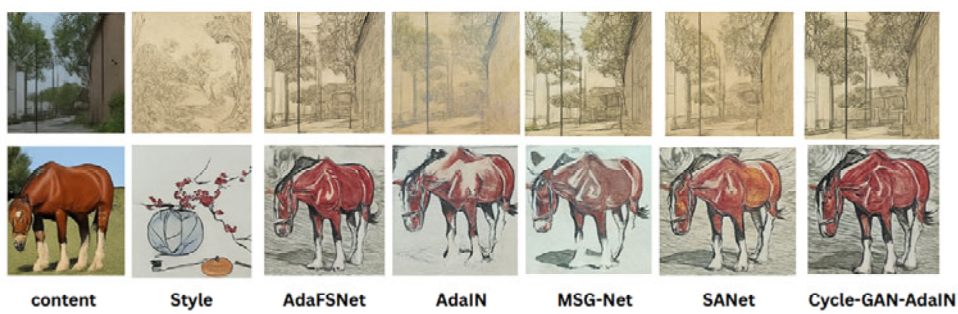
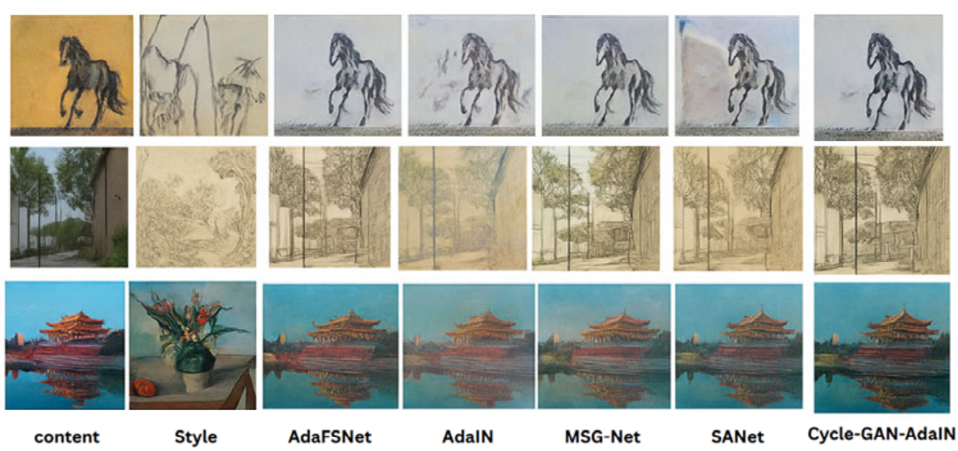


Figure 3 Output grid – unseen styles (see online version for colours)



5.2 *Quantitative results*

To supplement visual inspection, quantitative results were collected using three commonly accepted metrics:

- PSNR: higher values indicate better pixel-level reconstruction.
- SSIM: closer to 1 suggests better structural fidelity.
- LPIPS: lower values mean closer human-perceived similarity.

Table 4 Average metrics across models

Model variant	*PSNR* ↑	*SSIM* ↑	*LPIPS* ↓
AdaIN	21.3	0.72	0.34
MSG-Net	20.8	0.70	0.37
SANet	22.5	0.75	0.30
CycleGAN-AdaIN	21.0	0.71	0.36
AdaFSNet	24.1	0.81	0.25

As seen in Table 4, AdaFSNet consistently outperforms others across all metrics. These numbers are also visualised below.

Figure 4 PSNR comparison between models (see online version for colours)

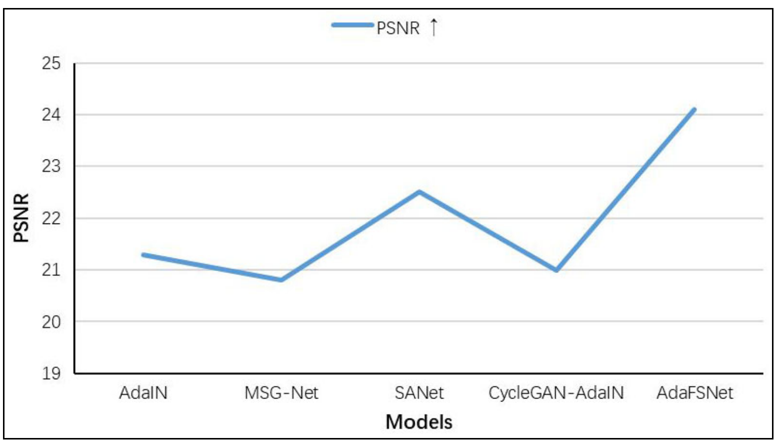


Figure 5 SSIM comparison between models (see online version for colours)

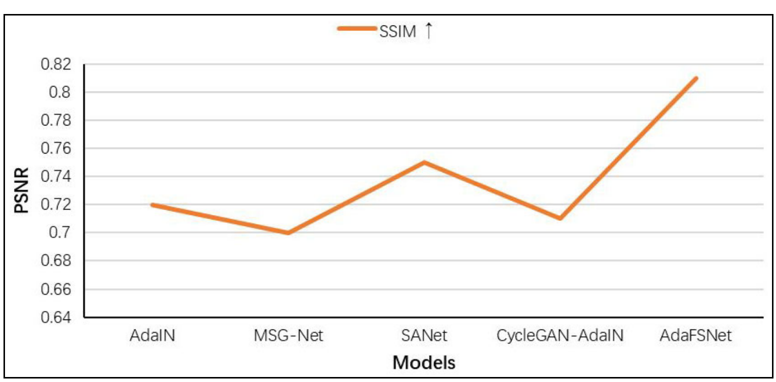


Figure 4 plots PSNR for all five models, where AdaFSNet peaks above 24 dB.

Figure 5 shows SSIM trends, with AdaFSNet achieving 0.81, confirming superior structural retention.

Figure 6 presents LPIPS, where AdaFSNet scores the lowest, indicating the most human-like results.

Figure 6 LPIPS comparison between models (see online version for colours)

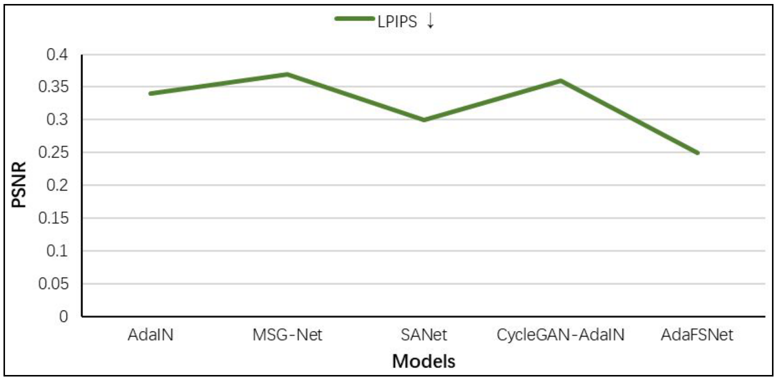


Figure 7 Average performance metrics across models (see online version for colours)

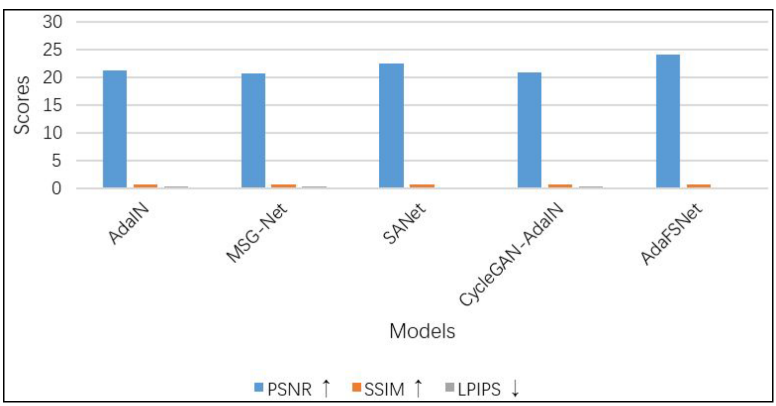
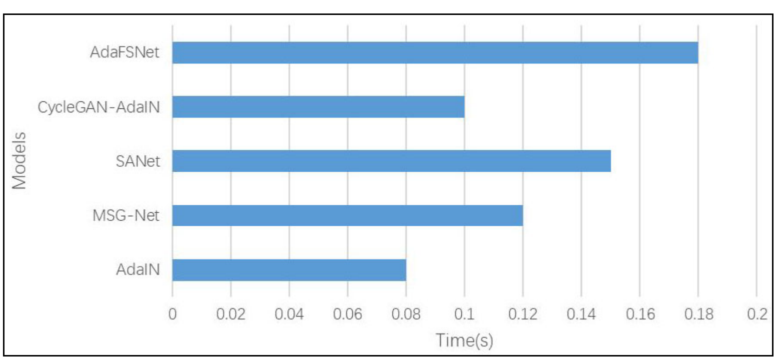


Figure 8 Stylisation time per image (see online version for colours)



This grouped bar chart combines all three metrics for easier comparison. AdaFSNet leads across each, particularly in LPIPS and SSIM, where perceptual quality matters most (see Figure 7).

Figure 9 PSNR comparison line chart (seen vs. unseen) (see online version for colours)

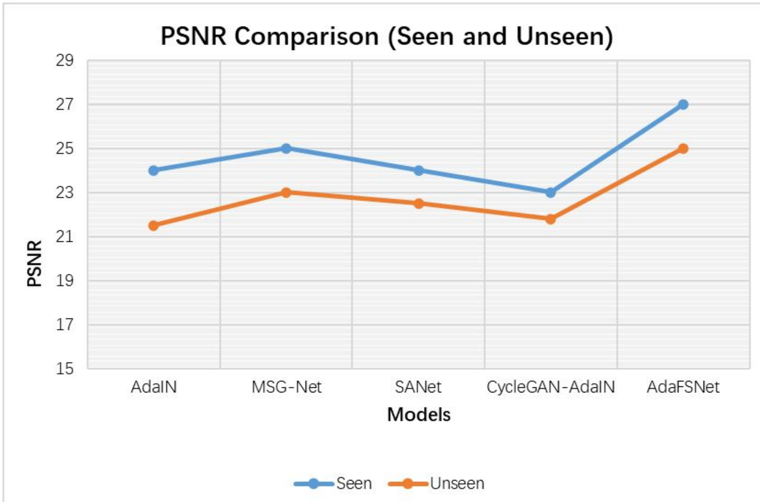
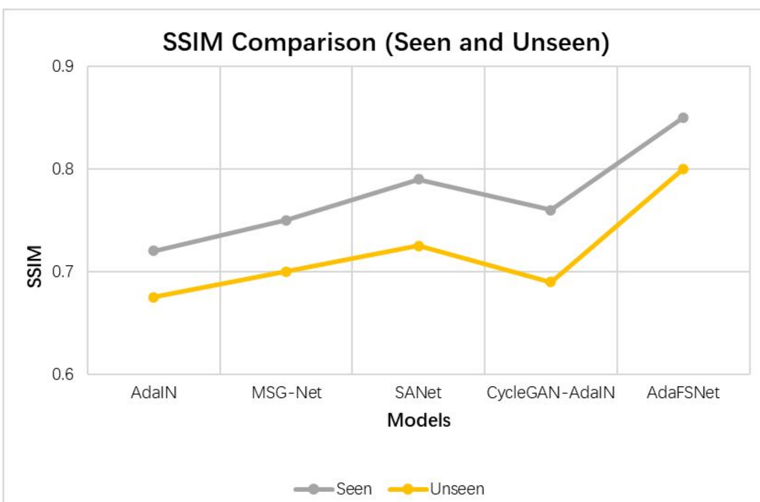
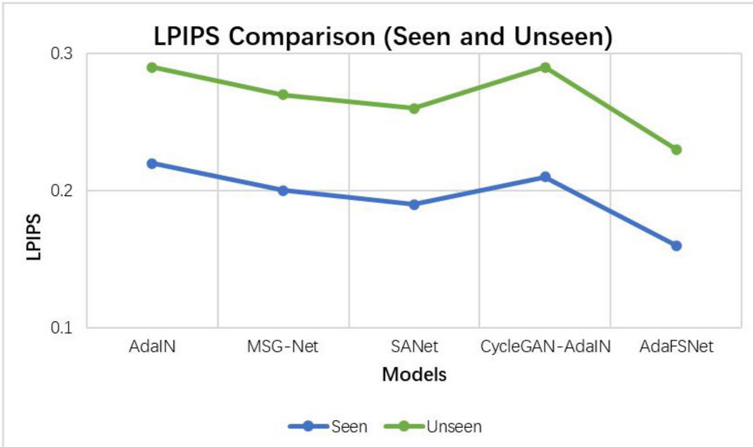


Figure 10 SSIM comparison line chart (seen vs. unseen) (see online version for colours)



While AdaFSNet offers the best visual quality, it is also the slowest among the five, taking approximately 0.18 seconds per image. AdaIN is the fastest (~0.08 seconds), making it suitable for real-time applications where speed outweighs quality (see Figure 8). However, AdaFSNet remains within practical latency for most offline use-cases.

Figure 11 LPIPS comparison line chart (seen vs. unseen) (see online version for colours)



To evaluate generalisation, metrics are broken down by style familiarity:

- AdaFSNet maintains top performance across both seen and unseen styles. You can see this in Figure 9, Figure 10 and Figure 11.
- Other models experience greater drops in SSIM and LPIPS when tested on unseen inputs.

5.3 Discussion of results

The results demonstrate that the proposed AdaFSNet model excels in both seen and unseen scenarios. It achieves the highest average PSNR and SSIM while recording the lowest LPIPS values, indicating a strong balance of fidelity and perceptual realism.

Key observations:

- Texture preservation: AdaFSNet excels at retaining delicate brush strokes and texture gradients, vital in ink wash painting styles such as ink-wash and calligraphic fusion.
- Style generalisation: its feature scaling mechanism enables adaptability to unseen artistic domains without model fine-tuning.
- Structural clarity: the model maintains clear edges and object shapes, likely due to its deep multi-scale feature extraction layers.
- Quality-speed trade-off: AdaFSNet is slower, largely due to more complex decoder paths and adaptive feature computation. However, the quality improvements justify the added computation in many artistic and restoration contexts.

In contrast, AdaIN and MSG-Net show faster inference but fail to preserve structure and depth, particularly in complex ink wash paintings forms. SANet performs reasonably well but lacks stability across unseen samples. CycleGAN-AdaIN offers decent style blending but often distorts original content structures.

This section confirms the effectiveness of AdaFSNet in ink wash style transfer tasks. Through strong results on both visual and metric-based evaluations, it surpasses baseline and enhanced models in content preservation, perceptual quality, and generalisation. The findings also emphasise the importance of balancing fidelity and speed depending on application context.

6 Ablation study

To understand the contributions of individual components within AdaFSNet, we conducted a targeted ablation study (Cheng et al., 2019). Each model variant was tested under the same setup using objective metrics (PSNR, SSIM and LPIPS) defined in Section 4.2. The mathematical interpretations of the observed changes help clarify the unique role of each module in stylisation quality. Table 5 summarises the metrics, and Figure 6 illustrates visual differences.

6.1 Removal of AFS

Eliminating the AFS module disables the learnable affine transformation applied to feature maps [see equation (8)].

Without AFS, the system reverts to basic AdaIN operations without spatial tuning. Quantitatively, we observed:

PSNR dropped from 24.1 to 22.3 dB, calculated as:

$$PSNR = 10 \bullet \log_{10}\left(\frac{255^2}{MSE}\right) \tag{22}$$

This indicates a structural degradation in the output compared to the original content image.

SSIM decreased from 0.81 to 0.76 showing lower spatial and contrast similarity [see equation (19)].

LPIPS increased from 0.250.250.25 to 0.310.310.31, reflecting a greater perceptual gap [see equation (20)].

The stylised images lost finer texture continuity and showed duller brushstroke simulation, especially in high-frequency ink wash regions.

6.2 Removal of AdaIN

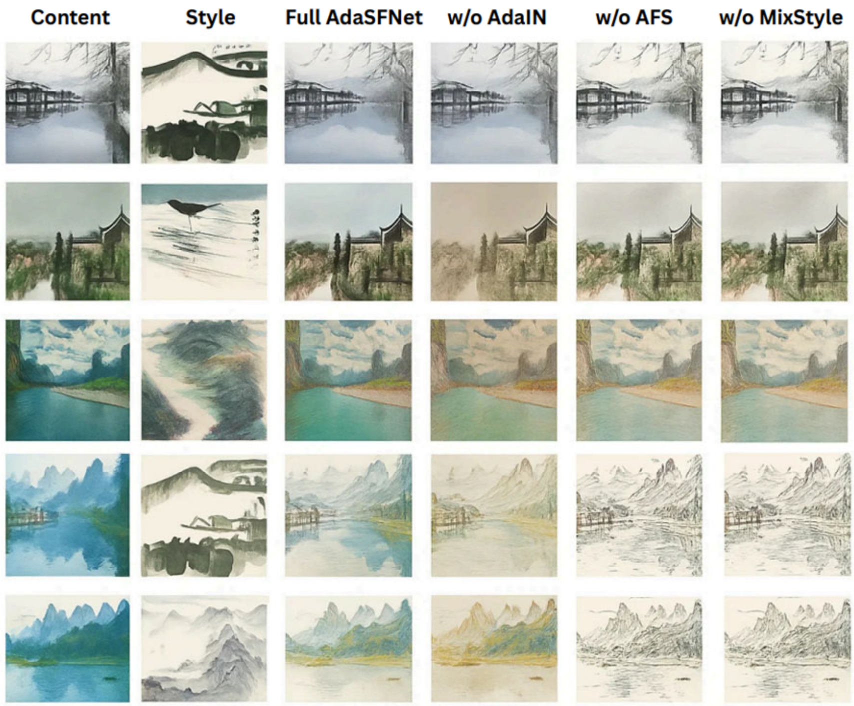
We replaced AdaIN with simple instance normalisation:

$$IN(F) = \frac{F - \mu(F)}{\sigma(F)} \tag{23}$$

This change broke the alignment of feature statistics between content and style [see equation (4)].

This led to the largest degradation, with LPIPS increasing by 0.11 and PSNR falling to 21.0 dB. Stylisations became inconsistent, with artefacts, colour saturation, and structure distortion – highlighting the irreplaceable role of AdaIN in initialising stylisation transfer.

Figure 12 Ablation results for AdaFSNet (see online version for colours)



6.3 Variation in scaling placement

We experimented with three locations for inserting the AFS module:

- after encoder
- mid-decoder (baseline placement)
- post-decoder.

Mathematically, the difference comes down to the timing of applying affine transformation α, β the decoding pipeline. Placing AFS mid-decoder allows it to modulate mid-level features F_m during reconstruction:

$$F_{m'} = \alpha \cdot \left(\frac{F_m - \mu(F_m)}{\sigma(F_m)} \right) + \beta \tag{24}$$

This version achieved the highest SSIM of 0.81 and lowest LPIPS of 0.25, as shown in the full AdaFSNet row in Table 5. In contrast, post-decoder scaling failed to influence the internal structure and performed poorly (LPIPS = 0.32), while pre-decoder scaling lacked adaptive guidance.

Side-by-side visual outputs show clear degradation in brushstroke fidelity and colour blending when adaptive scaling is removed or poorly positioned (see Figure 12).

This ablation confirms that both AdaIN and the AFS module are indispensable for AdaFSNet's performance. Not only do they control stylisation intensity, but their architectural placement also plays a critical role in preserving ink wash specific textures and semantics.

Table 5 Ablation results (average metrics)

Model variant	*PSNR* ↑	*SSIM* ↑	*LPIPS* ↓
Full AdaFSNet	24.1	0.81	0.25
Feature scaling	22.3	0.76	0.31
AdaIN	21.0	0.70	0.36
Scaling after encoder	23.0	0.78	0.29
Scaling post-decoder	22.6	0.75	0.32

7 Limitations and conclusions

AdaFSNet offers strong advancements in ink wash style transfer by effectively addressing common issues such as texture distortion, colour inversion, and weak generalisation. Through the integration of AdaIN, AFS, and specialised loss functions, the model achieves an effective balance between content fidelity and stylistic richness. It consistently outperforms leading baselines like AdaIN, SANet, MSG-Net, and CycleGAN-AdaIN across both seen and unseen datasets – including traditional ink wash paintings and contemporary art – demonstrating high accuracy in PSNR, SSIM, and LPIPS evaluations, as well as compelling visual outcomes.

Despite these strengths, some limitations remain. The model currently requires moderate to high computational resources, making real-time applications on mobile or edge devices challenging without further compression or pruning techniques. Additionally, AdaFSNet does not yet support temporal consistency across video frames, which can lead to flickering or drift in stylised animations. This limits its immediate use in augmented reality or video-based ink wash paintings experiences.

Moreover, while AdaFSNet performs well in single-style scenarios, it does not yet allow users to mix multiple styles or apply region-specific stylisation – a useful feature for interactive or user-guided artistic control. Another frontier lies in adapting the model for 3D ink wash paintings surfaces. Since real-world ink wash paintings often appear on curved or textured walls, extending the model to accept depth-aware or UV-mapped inputs could bridge the gap between digital and physical conservation work.

In summary, AdaFSNet represents a significant step forward in NST for ink wash paintings, showing robustness, fine detail preservation, and stylistic accuracy. However, improvements in runtime efficiency, temporal stability, and multi-style interaction are needed to unlock its full potential. Future research will explore lightweight architectures, video frame stabilisation, and 3D-aware enhancements to broaden its usability in digital heritage, AR, and creative design. With these refinements, AdaFSNet is well-positioned to contribute meaningfully to both cultural preservation and artistic innovation.

Declarations

All authors declare that they have no conflicts of interest.

References

Abdel, K.M.B., Ahmad, A-J., Ahmad, M.S. and Ghazi, H.S. (2018) 'Performance evaluation of edge detection using Sobel, homogeneity and Prewitt algorithms', *Journal of Software Engineering and Applications*, Vol. 11, No. 11, pp.537–551.

Bai, Z., Xu, H., Zhang, X. and Ding, Q. (2023) 'GCSANet: arbitrary style transfer with global context self-attentional network', *IEEE Transactions on Multimedia*, Vol. 26, pp.1407–1420.

Cao, X., Zhou, F., Xu, L., Meng, D., Xu, Z. and Paisley, J. (2018) 'Hyperspectral image classification with Markov random fields and a convolutional neural network', *IEEE Transactions on Image Processing*, Vol. 27, No. 5, pp.2354–2367.

Carlson, K., Riddell, A. and Rockmore, D. (2018) 'Evaluating prose style transfer with the bible', *Royal Society Open Science*, Vol. 5, No. 10, p.171920.

Cheng, M-M., Liu, X-C., Wang, J., Lu, S-P., Lai, Y-K. and Rosin, P.L. (2019) 'Structure-preserving neural style transfer', *IEEE Transactions on Image Processing*, Vol. 29, pp.909–920.

Colace, F., Loia, V., Pedrycz, W. and Tomasiello, S. (2020) 'On a granular functional link network for classification', *Neurocomputing*, Vol. 398, pp.108–116.

Deng, X. (2018) 'Enhancing image quality via style transfer for single image super-resolution', *IEEE Signal Processing Letters*, Vol. 25, No. 4, pp.571–575.

Hien, N.L.H., Van Huy, L. and Van Hieu, N. (2021) 'Artwork style transfer model using deep learning approach', *Cybernetics and Physics*, Vol. 10, No. 3, pp.127–137.

Hu, C., Sun, S., Zhao, P., Zhao, Y., Liu, J. and Song, X. (2025) 'Dual-domain style transfer network', *Science Progress*, Vol. 108, No. 2, p.00368504251325649.

Hwang, I. and Oh, T. (2025) 'Design and experimental research of on device style transfer models for mobile environments', *Scientific Reports*, Vol. 15, No. 1, p.13724.

Jin, X., Lan, C., Zeng, W. and Chen, Z. (2021) 'Style normalization and restitution for domain generalization and adaptation', *IEEE Transactions on Multimedia*, Vol. 24, pp.3636–3651.

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y. and Song, M. (2019) 'Neural style transfer: a review', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 26, No. 11, pp.3365–3385.

Kumar, T., Brennan, R., Mileo, A. and Bendechache, M. (2024) 'Image data augmentation approaches: a comprehensive survey and future directions', *IEEE Access*.

Li, Z., Lu, S., Yuan, Z., Hou, B. and Bian, J. (2025) 'Interactive instance search: user-centered enhanced image retrieval with learned perceptual image patch similarity', *Electronics*, Vol. 14, No. 9, p.1766.

Liu, Z., Zhang, Z. and Hu, W. (2025) 'Multi-level feature fusion and high-fidelity information transfer algorithm for mural inpainting', *NPJ Heritage Science*, Vol. 13, No. 1, pp.1–13.

Lv, C., Zhang, D., Geng, S., Wu, Z. and Huang, H. (2024) 'Color transfer for images: a survey', *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 20, No. 8, pp.1–29.

Nam, H. and Kim, H-E. (2018) 'Batch-instance normalization for adaptively style-invariant neural networks', *Advances in Neural Information Processing Systems*, Vol. 31.

Niu, Y., Wu, L., Zhang, Y., Zhu, Y., Zhu, G. and Wang, J. (2024) 'Multi-model style-aware diffusion learning for semantic image synthesis', *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 20, No. 11, pp.1–21.

Odena, A., Dumoulin, V. and Olah, C. (2016) 'Deconvolution and checkerboard artifacts', *Distill*, Vol. 1, No. 10, p.e3.

Sun, Y., Xie, X., Li, Z. and Zhao, H. (2025) 'Image style transfer with saliency constrained and SIFT feature fusion', *The Visual Computer*, Vol. 41, No. 7, pp.4915–4930.

Tan, Z., Chen, D., Chu, Q., Chai, M., Liao, J., He, M., Yuan, L., Hua, G. and Yu, N. (2021) 'Efficient semantic image synthesis via class-adaptive normalization', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44, No. 9, pp.4852–4866.

Tuyen, N.Q., Nguyen, S.T., Choi, T.J. and Dinh, V.Q. (2021) 'Deep correlation multimodal neural style transfer', *IEEE Access*, Vol. 9, pp.141329–141338.

Wang, P., Cheng, J., Hao, F., Wang, L. and Feng, W. (2020) 'Embedded adaptive cross-modulation neural network for few-shot learning', *Neural Computing and Applications*, Vol. 32, pp.5505–5515.

Wang, Q., Wang, H., Zang, L., Jiang, Y., Wang, X., Liu, Q. and Huang, D. (2025) 'Gated normalization unit for image restoration', *Pattern Analysis and Applications*, Vol. 28, No. 1, pp.1–19.

Wang, G., Shi, H., Chen, Y. and Wu, B. (2023a) 'Unsupervised image-to-image translation via long-short cycle-consistent adversarial networks', *Applied Intelligence*, Vol. 53, No. 14, pp.17243–17259.

Wang, W., Li, Y., Ye, H., Ye, F. and Xu, X. (2023b) 'Ink painting style transfer using asymmetric cycle-consistent GAN', *Engineering Applications of Artificial Intelligence*, Vol. 126, p.107067.

Wang, Z., Zhao, L., Chen, H., Zuo, Z., Li, A., Xing, W. and Lu, D. (2021) 'Evaluate and improve the quality of neural style transfer', *Computer Vision and Image Understanding*, Vol. 207, p.103203.

Wei, H-P., Deng, Y-Y., Tang, F., Pan, X-J. and Dong, W-M. (2022) 'A comparative study of CNN- and transformer-based visual style transfer', *Journal of Computer Science and Technology*, Vol. 37, No. 3, pp.601–614.

Wu, H., Sun, Z., Zhang, Y. and Li, Q. (2019) 'Direction-aware neural style transfer with texture enhancement', *Neurocomputing*, Vol. 370, pp.39–55.

Xu, Y., Xia, M., Hu, K., Zhou, S. and Weng, L. (2025) 'Style transfer review: traditional machine learning to deep learning', *Information*, Vol. 16, No. 2, p.157.

Zhao, Z. and Zhang, S. (2022) 'Style transfer based on VGG network', *International Journal of Advanced Network, Monitoring and Controls*, Vol. 7, No. 1, pp.54–72.

Zhou, K., Yang, Y., Qiao, Y. and Xiang, T. (2024) 'Mixstyle neural networks for domain generalization and adaptation', *International Journal of Computer Vision*, Vol. 132, No. 3, pp.822–836.

Zhu, X., Lin, M., Yi, M. and Zhao, H. (2024) 'Photorealistic attention style transfer network for architectural photography photos', *Scientific Reports*, Vol. 14, No. 1, p.29584.