NEURAL STYLE TRANSFER

A SEMINAR REPORT

Submitted by

AKSHAY PRAKASH (SNG22CS026)

To

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering

Department of Computer Science and Engineering

Sree Narayana Gurukulam College of Engineering,

Kadayiruppu, 682311

August 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING,
KADAYIRUPPU, 682311
(Affiliated to APJ Abdul Kalam Technological University & Approved by A.I.C.T.E)

CERTIFICATE

This is to certified that the seminar report, **"Neural Style Transfer"** submitted by **Akshay Prakash** to the Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bona fide record of the project work carried out by them under our guidance and supervision .This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. (Dr.) Smitha Suresh	**Asst. Prof. Soumya A V**	**Asst. Prof. Archana P S**
HEAD OF THE DEPARTMENT	**SEMINAR GUIDE**	**SEMINAR COORDINATOR**

Submitted for the University Evaluation on:

University Register Number :

Internal Examiner External Examiner

DECLARATION

We undersigned hereby declare that the seminar report **"Neural Style Transfer"**, submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Asst. Prof. Archana P.S** . This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu

Date : Akshay Prakash

ACKNOWLEDGEMENT

Dedicating this seminar to the Almighty God whose abundant grace and mercy enabled its successful completion, I would like to express our profound gratitude to all the people who had inspired and motivated me to undertake this seminar .

I wish to express our sincere thanks to our Head of the Department, **Prof. (Dr.) Smitha Suresh**, for providing us the opportunity to undertake this seminar. I am deeply indebted to our project coordinator **Asst. Prof**. **Archana P.S** and project guide **Asst. Prof**. **Soumya A.V** in the Department of Computer Science and Engineering for providing us with valuable advice and guidance during the course of the project.

Finally, I would like to express my gratitude to Sree Narayana Gurukulam College of Engineering for providing me with all the required facilities without which the successful completion of the seminar work would not have been possible.

COURSE OUTCOMES AND PROGRAM OUTCOMES

CO PO PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	1	1		2	1					3	3	2
CO2	3	3	2	3		2	1					3	3	2
CO3	3	2			3			1		2		3	3	3
CO4	3				2			1		3		3	3	2
CO5	3	3	3	3	2	2		2		3		3	3	3

CO-PO-PSO MAPPING JUSTIFICATION

Mapping	Points Attained	Justification
CO1-PO1	2	Students apply core computer-vision and deeplearning concepts (feature maps, convolution responses) to understand how content and style are represented; this moderately supports foundational engineering knowledge.
CO1-PO2	2	Interpreting how feature correlations and loss terms affect outputs develops analytical skills in diagnosing model behaviour and reasoning about problem statements.
CO1-PO3	1	Provides a basic awareness of network layer roles, such as which VGG layers capture texture versus semantics, offering only a slight link to system design understanding.
CO1-PO4	1	Encourages small investigative exercises like observing the effects of loss tweaks or layer choices, creating an introductory connection to investigative methods.
CO1-PO6	2	Raises moderately important ethical and societal considerations, including potential misuse and aesthetic bias, prompting students to reflect on such implications.
CO1-PO7	1	Promotes a minor awareness of computational efficiency and lightweight model choices during experimentation, relating slightly to sustainability.

CO1-PO12	3	Demands continuous reading of literature and hands-on experimentation, strongly supporting lifelong learning and staying current with methods.
CO1-PSO1	3	Builds practical computer-vision and deeplearning skills, such as feature extraction and representation, which strongly support the program-specific outcome.
CO1-PSO2	2	Helps students moderately relate neural methods to software and creative applications, for example applying models to produce stylized outputs.
CO2-PO1	3	Requires strong mathematical and engineering grounding to analyse Gram matrices, innerproduct computations, and spatial transforms.
CO2-PO2	3	Demands high-level analytical and problemsolving skills to evaluate improved loss formulas and reason about their effects on visual artifacts.
CO2-PO3	2	Translates analytical improvements into architectural choices or algorithmic changes, representing a moderate design-level activity.
CO2-PO4	3	Involves systematic experimentation, ablation studies, and validation using metrics such as PSNR and SSIM, strongly supporting investigative competence.
CO2-PO6	2	Considers ethical and societal implications of generated images, giving moderate emphasis to responsible use and potential harms.
CO2-PO7	1	Shows slight awareness of sustainability concerns, such as computational costs during iterative experiments.
CO2-PO12	3	Encourages the adoption of new loss ideas and recent literature, strongly supporting lifelong learning and continuous upskilling.
CO2-PSO1	3	Strengthens core computer-vision and deeplearning skills through improved loss functions and experiments.
CO2-PSO2	2	Moderately supports applying these research advances to real-world software and creative image tasks.
CO3-PO1	3	Requires strong foundational deep-learning knowledge to implement NST with pretrained networks and manipulate feature maps.
CO3-PO2	2	Involves selecting and adapting architectures or components, such as deciding which layers to use for content and style representation, which is a moderate design activity.

CO3-PO5	3	Makes extensive use of machine learning frameworks and tools such as PyTorch or TensorFlow, strongly mapping to modern tool competency.
CO3-PO8	1	Encourages ethical use of models and datasets, including respecting licenses and attribution, but this plays a limited role in implementation.
CO3-PO10	2	Improves technical communication through documentation of implementation steps and explanations of technical workflows.
CO3-PO12	3	Promotes continuous learning and iterative improvement through hands-on development and experimentation.
CO3-PSO1	3	Builds professional competence in image processing implementations expected in the program.
CO3-PSO2	3	Strongly supports the creation of domainspecific software solutions, such as stylization modules, using AI methods.
CO4-PO1	3	Relies on strong domain knowledge to interpret quantitative metrics like PSNR and SSIM and combine them with perceptual judgement.
CO4-PO5	2	Uses analysis and visualization tools, such as plotting scripts and metric computation programs, to compare outputs.
CO4-PO8	1	Promotes limited ethical awareness, such as avoiding misrepresentation of generated media.
CO4-PO10	3	Requires clear presentation of evaluation outcomes, including figures, tables, and reasoned discussion, strongly supporting communication skills.
CO4-PO12	3	Encourages ongoing learning and refinement of methods through review of recent literature and evaluation techniques.
CO4-PSO1	3	Strengthens applied evaluation and comparative skills relevant to professional computer-vision and deep-learning tasks.
CO4-PSO2	2	Moderately supports assessing whether generated outputs meet the needs of specific software or creative applications.
CO5-PO1	3	Applies engineering concepts accurately in the preparation of a detailed technical report.
CO5-PO2	3	Demonstrates strong analytical ability through critical assessment of methods, results, and limitations.
CO5-PO3	3	Shows strong design understanding by documenting architecture choices, algorithmic trade-offs, and system flows.

CO5-PO4	3	Strongly supports investigative skills through detailed reporting of experimental setups, metrics, and inferences.
CO5-PO5	2	Moderately enhances tool competency through the use of document creation and visualization tools such as LaTeX, Word, and plotting libraries.
CO5-PO6	2	Discusses societal and ethical implications of the work, giving moderate attention to these aspects.
CO5-PO8	2	Maintains ethical writing practices, including proper citation and acknowledgment of datasets and prior work.
CO5-PO10	3	Strongly supports communication skills through clear written and visual presentation of technical findings.
CO5-PO12	3	Encourages ongoing learning and research interest through reflective project writing.
CO5-PSO1	3	Develops professional technical documentation skills expected in the domain.
CO5-PSO2	3	Strongly supports clear articulation of domain specific AI and software concepts for various audiences.

ABSTRACT

Neural Style Transfer (NST) is a deep learning technique that generates artistic images by combining the content of one image with the style of another. Using Convolutional Neural Networks (CNNs) such as VGG-19, the model extracts content features from the content image and style features from the style image. These features are compared using a special loss function that balances content preservation with style application. The style is often represented using a Gram matrix, which captures texture information, while content loss ensures the original structure is maintained. An improved version of NST enhances the style loss function by incorporating spatial information, resulting in clearer textures, reduced distortions, and better visual quality. This technology has wide applications in digital art, photography, film production, and virtual reality, showcasing the creative potential of artificial intelligence in image processing.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

Sl. No.	Abbreviation	Full Form
1	AI	Artificial Intelligence
2	CNN	Convolutional Neural Network
3	NST	Neural Style Transfer
4	VGG-19	Visual Geometry Group Network-19 Layers
5	GPU	Graphics Processing Unit
6	Gram Matrics	Gramian Matrix(used in style representation)
7	SGD	Stochastic Gradient Descent
8	TV Loss	Total Variation Loss
9	DL	Deep Learning
10	ML	Machine Learning
11	CVPR	Conference on Computer Vision and Pattern Recognition
12	MSE	Mean Squared Error
13	ReLU	Rectified Linear Unit
14	PyTorch	Python Torch(Deep Learning Framework)
15	TensorFlow	TensorFlow(Deep Learning Framework)

CHAPTER 1
INTRODUCTION

1.1 Neural Style Transfer: A Deep Learning-Based Artistic Image Generation Framework

Neural Style Transfer (NST) is a computer vision technique that uses deep learning to merge the content of one image with the artistic style of another. By leveraging Convolutional Neural Networks (CNNs) such as VGG-19, NST extracts content features from one image and style features from another, then blends them into a single output image. This process is guided by a loss function that balances content preservation and style transformation.
An improved version of NST enhances the style loss function by incorporating spatial maps alongside the traditional Gram matrix, allowing the output to retain clearer textures and reduce distortions. This advancement makes NST suitable for applications in digital art, media production, virtual reality, and other creative fields, showcasing the synergy between artificial intelligence and human creativity.

1.2 Key Features

- **Content-Style Fusion**
  Combines structural details from the content image with artistic patterns from the style image to produce visually appealing results.
- **Feature Extraction Using CNNs**
  Uses pretrained networks like VGG-19 to detect and separate content and style features across multiple convolutional layers.
- **Enhanced Style Loss Function**
  Incorporates spatial transformation maps in addition to the Gram matrix to better capture texture placement and reduce distortion.
- **High Visual Quality Output**
  Generates stylized images with smoother textures, clear details, and faithful

reproduction of style patterns.

- **Versatile Artistic Applications**

  Can be applied to photographs, videos, and 3D scenes for creative or commercial purposes.

1.3 Objective

The main objective of Neural Style Transfer is to develop an AI-powered image processing framework that can seamlessly combine the content of one image with the artistic style of another while maintaining high visual quality and structural accuracy. This involves:

- Preserving the structural details and spatial layout of the content image.
- Accurately transferring the color patterns, textures, and brushstroke-like effects from the style image.
- Enhancing the style transfer process using improved style loss functions that incorporate spatial maps alongside the traditional Gram matrix.
- Minimizing distortions and ensuring smooth, natural-looking textures in the output image.
- Enabling creative and commercial applications in digital art, media production, photography, and virtual reality.

1.4 Technological Framework

The NST framework operates using deep learning models optimized for image feature extraction and transformation.

- **AI Models:** Convolutional Neural Networks (VGG-19) for feature extraction, optimization algorithms (e.g., gradient descent) for loss minimization.
- **Loss Functions:** Content loss for preserving structure, style loss (Gram matrix + spatial maps) for accurate texture transfer.
- **Data Handling:** Input images processed in multiple resolutions to capture both global structure and fine details.

- **Computational Requirements:** High-performance GPUs for faster training and inference due to the complexity of convolution operations.
- **Interoperability:** Can be integrated into image editing tools, creative software, and real-time rendering pipelines.

CHAPTER 2

LITERATURE SURVEY

[1] This paper presents a Neural Style Transfer (NST) network for image augmentation of corn leaves, proposed in the 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS). The authors design a CNN-based NST framework to generate synthetic variations of corn leaf images, aiming to enhance dataset diversity for training deep learning models in agriculture. By transferring artistic patterns and textures onto original corn leaf images, the augmented dataset helps improve the performance of plant disease detection models. Experimental results show that the approach increases classification accuracy compared to traditional augmentation methods. However, the framework requires significant computational resources, which may limit its use in real-time agricultural applications. This study demonstrates the potential of NST as a creative and effective data augmentation tool for precision agriculture.

[2] This paper, presented at the 2022 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR), provides a comprehensive study of **Neural Style Transfer (NST) methods for images**. The author examines classical optimization-based approaches introduced by Gatys et al., feed-forward generative networks, and more recent transformer-based methods. The study highlights how optimization-based NST achieves high-quality stylizations but suffers from slow processing times, making it less practical for real-time use. In contrast, feed-forward models significantly speed up stylization but often compromise texture richness and fine artistic details. Transformer-based techniques are noted for their ability to capture long-range dependencies, improving global consistency of style transfer, though they introduce high computational costs. The paper compares these methods experimentally, analyzing stylization quality, structural preservation, and efficiency. The results emphasize that no single method is perfect—each involves trade-offs between visual fidelity, speed, and resource requirements. The study concludes by suggesting hybrid models and improved loss functions as promising directions for advancing NST, especially for real-time artistic applications and practical deployment in mobile or embedded systems.

[3] This paper, published in *IEEE Computational Intelligence Magazine* (2025), serves as a concise, accessible introduction to neural style transfer (NST) under the title *"Explaining Neural Style Transfer [AI-eXplained]."* The authors provide an interactive exploration of NST concepts, including how CNNs enable the separation and fusion of content and style features. They also highlight extensions such as color correction models, making this a valuable entry point for readers new to the field. While not presenting new algorithmic contributions, the article excels in demystifying NST processes and helping readers visualize the internal workings of style transfer. However, as a short overview, it lacks in-depth technical or empirical evaluation. Nonetheless, it fills a unique role in bridging technical understanding with visual intuition for educational and explanatory purposes.

[4] This article introduces **PhotoStyle60**, a new photographic style dataset built for authorship attribution and style transfer research, published in *IEEE Transactions on Multimedia* (2024). The authors compile over 5,700 images from 60 photographers, providing labeled examples of distinct photographic styles. Although not an NST method paper, PhotoStyle60 offers a valuable resource—particularly for training and evaluating style transfer models sensitive to photographer-specific characteristics. By enabling fine-grained stylistic analysis, this dataset can help improve NST models' ability to replicate or adapt to particular aesthetic styles. However, its current limitation is that it's constrained to still images and may not include annotations for dynamic or generative stylization workflows. The dataset paves the way for more personalized and artist-aware style transfer applications.

[5] This paper, published in *IEEE Signal Processing Letters* (2022), introduces a novel improvement to neural style transfer (NST) by incorporating an adaptive auto-correlation alignment loss. Traditional NST approaches rely on Gram matrices to represent style, but they often fail to capture fine-grained spatial and textural correlations, which can lead to distortions in stylized outputs. To address this, the authors design a loss function that adaptively aligns the auto-correlations of feature maps between the style and generated images. This refinement allows for more accurate preservation of texture patterns and artistic strokes while maintaining the overall structure of the content image. Experimental comparisons against baseline NST methods show that the proposed approach achieves clearer textures, reduced artifacts, and higher perceptual quality in stylized results. Although the method increases computational cost, making it less suitable for real-time applications, it highlights how enhanced loss functions can push NST closer to producing visually appealing and structurally faithful results. The work underscores the importance of balancing efficiency with fidelity in advancing NST for practical use in art, design, and media applications

[6] This journal article presents a novel **patch-based style transfer** technique using **optimal transport for patch matching**, published in *IEEE Transactions on Multimedia* (2022). The authors diverge from conventional neural network–based NST methods by operating directly in the pixel domain using patches, optimized via an optimal transport criterion. This approach avoids the overhead and opacity of deep learning while generating compelling stylized results with rich visual detail. Through experimental comparisons, the paper shows that their method achieves comparable or better style coherence and texture richness than some neural methods, particularly in scenarios where training resources are limited. However, the patch-based optimization can be computationally heavy and may struggle with large image sizes or complex style representations. This study offers a refreshing, mathematically grounded alternative to neural style transfer that emphasizes explainability and control over neural black-box models.

CHAPTER 3

METHODOLOGY

3.1 Data Acquisition and Preprocessing

The Neural Style Transfer (NST) process begins with the acquisition of two input images:

- **Content Image** – Represents the structural layout or subject to be preserved.
- **Style Image** – Represents the artistic patterns, textures, and color schemes to be applied.

Both images undergo preprocessing steps to ensure compatibility with the model:

- **Resizing** – Adjusting both images to a uniform resolution for optimal processing speed.
- **Normalization** – Scaling pixel values to match the input requirements of the pretrained CNN (e.g., VGG-19).
- **Color Space Alignment** – Ensuring both images are in the same color space (RGB) for accurate feature extraction.

This preprocessing ensures that the deep learning model receives clean, standardized inputs, enabling consistent feature comparison and minimizing artifacts in the final output.

3.2 Feature Extraction Using Convolutional Neural Networks (VGG-19)

After preprocessing, the **VGG-19** convolutional neural network is used as the computational backbone for extracting image features. The model processes both the content and style images through its convolutional layers:The network processes each data modality separately in dedicated sub-networks:

- **Content Feature Extraction** – Deeper layers of the network capture high-level structural information, such as object shapes and spatial arrangements.

- **Style Feature Extraction** – Multiple layers capture texture patterns and colors, summarized using a **Gram matrix** to represent correlations between feature maps.

- Wearable sensor sub-network processes time-series physiological signals such as heart rate, respiratory rate, blood oxygen levels, and activity metrics.

By separating the extraction of content and style features, the network ensures that both structural accuracy and stylistic richness are preserved during synthesis.

3.3 Improved Style Loss Function with Spatial Maps

In the original Neural Style Transfer developed by *Gatys et al.*, the style of an image is captured using a **Gram matrix**, which measures how different features in an image relate to each other. This works well for capturing overall color and texture patterns, but it ignores **where** these textures should appear. Because of this, textures in the generated image can sometimes appear **misplaced or distorted**.

The improved method enhances this process by adding **spatial transformation maps** alongside the traditional Gram matrix:

- **Spatial transformation maps** keep track of the position of textures in the style image.
- They ensure that textures are placed in **relevant areas** of the content image (for example, patterns from the sky in the style image are applied to the sky region in the content image)
- This reduces visual distortions and prevents textures from being scattered randomly.

By combining both **global texture patterns** (from the Gram matrix) and **local texture placement** (from spatial maps), the resulting image:

- Shows **clearer and sharper textures**
- Maintains a **more natural artistic look**
- Aligns style details more closely with the original content structure

This improvement is especially useful when working with complex artistic styles that contain **fine details**, such as brush strokes or fabric patterns, where correct positioning greatly improves the final quality.

3.4 Image Optimization Process

After extracting the content and style features, the Neural Style Transfer process focuses on **generating a new image** that blends both sources effectively. This is done through an **iterative optimization process**, where the pixels of the output image are adjusted step-by-step until the desired artistic effect is achieved.

The process begins with an initial image, which can be:

- A **copy of the content image** (for faster convergence)
- A **random noise image** (for more abstract results)

The goal is to make this initial image look like the **content image painted in the style of the style image**.

The optimization follows these steps:

1. Calculate Content Loss
    - Compares the structure of the generated image with the content image to ensure the main shapes and objects remain recognizable.
2. Calculate Style Loss
    - Compares the texture and color patterns of the generated image with the style image.
    - Uses both **Gram matrix** (for global style) and **spatial maps** (for correct texture placement).
3. Combine Losses into Total Loss
    - Balances style and content so that neither overpowers the other.
4. Update the Image
    - Uses optimization algorithms like **Gradient Descent**, **Adam**, or **L-BFGS** to gradually adjust pixel values.

- After each update, the image becomes closer to the target blend of content and style.

5. Repeat Until Convergence

- The process continues for several iterations until the output image reaches the desired quality, showing both the **structural essence** of the content and the **artistic textures** of the style.

By the end of this process, the generated image is a **harmonious combination** of the two source images, ready for use in digital art, design, or creative projects.

3.5 Evaluation and Validation

The quality of the generated images is assessed using both quantitative and qualitative metrics:

- **Quantitative Evaluation** – Metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are used to measure structural fidelity and visual similarity.
- **Qualitative Evaluation** – Human reviewers assess artistic appeal, texture clarity, and content preservation.

Comparative experiments are conducted against baseline NST models (e.g., Gatys et al.'s original method) to demonstrate improvements in texture detail and reduction of distortions.

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 Overview

The proposed **Neural Style Transfer (NST) framework** combines the structural features of a **content image** with the artistic features of a **style image** to generate a stylized output. The system is built on a modular deep learning architecture where each component performs a specific role—ranging from preprocessing and feature extraction to loss calculation and iterative optimization.

A key highlight of this architecture is the addition of an **outline loss module** (using GrabCut and distance mapping), which helps preserve the shape and boundaries of the original content image. Together with the traditional **content loss** and **style loss**, this ensures that the generated image maintains structure while achieving high-quality artistic transformation.

The core components of the architecture are:

1. **Input Layer** – Accepts two images: content image and style image.
2. **Preprocessing & Outline Extraction** – Applies GrabCut segmentation and distance matrix to extract and preserve object boundaries.
3. **Feature Extraction (VGG-19)** – Uses a pretrained CNN to capture both style and content features.
4. **Loss Function Module** – Calculates content loss, style loss, outline loss, and total variation loss.
5. **Optimization Engine** – Uses gradient descent to iteratively update pixels of the generated image.
6. **Output Layer** – Produces the final stylized image.

4.2 System Architecture

The system architecture of Neural Style Transfer is designed to transform an ordinary content image into a stylized output by combining it with the artistic features of a style image. The process begins with two input images: the **content image**, which provides the overall subject and structural arrangement, and the **style image**, which provides the artistic textures, colors, and brush stroke patterns. To further preserve the structure of the content image, an additional **outline extraction step** is applied. This involves the use of the GrabCut algorithm with user interaction, which separates the main object from the background, followed by a distance matrix and Laplacian transform to generate an outline image. This outline information is later integrated into the training process to ensure that object boundaries remain sharp and clear in the final stylized image.

Both the content and style images are processed by the **VGG-19 convolutional neural network (CNN)**, which is a pre-trained model commonly used in image classification and feature extraction. The VGG-19 extracts hierarchical image features at different depths of its layers. For the content image, deeper layers capture the high-level structure and semantic details, while for the style image, both shallow and deep layers are used to capture textures, fine details, and overall artistic patterns. These extracted features form the basis for comparing how well the generated image matches both the content and style of the input images.

The extracted features are passed into the **loss function module**, where four types of losses are calculated. **Content loss** ensures that the generated image does not lose the essential structure of the content image. **Style loss** transfers the artistic patterns of the style image, such as brush strokes, color distributions, and texture patterns. **Outline loss**, supported by the GrabCut and distance matrix process, preserves object boundaries and prevents distortions around edges. **Total variation loss** is also included to reduce pixel noise and improve smoothness, making the image visually more natural. These losses are combined to form a total loss function, which serves as the guiding metric for the optimization process.

To refine the generated image, the architecture uses a **gradient descent optimization engine**. The optimization begins either from a random image or a copy of the content image and iteratively updates the pixel values of the generated image in small steps. At each step, the algorithm minimizes the combined losses, ensuring that the image increasingly resembles the content image in structure while adopting the artistic style features. This iterative process continues until the output reaches a visually balanced state.

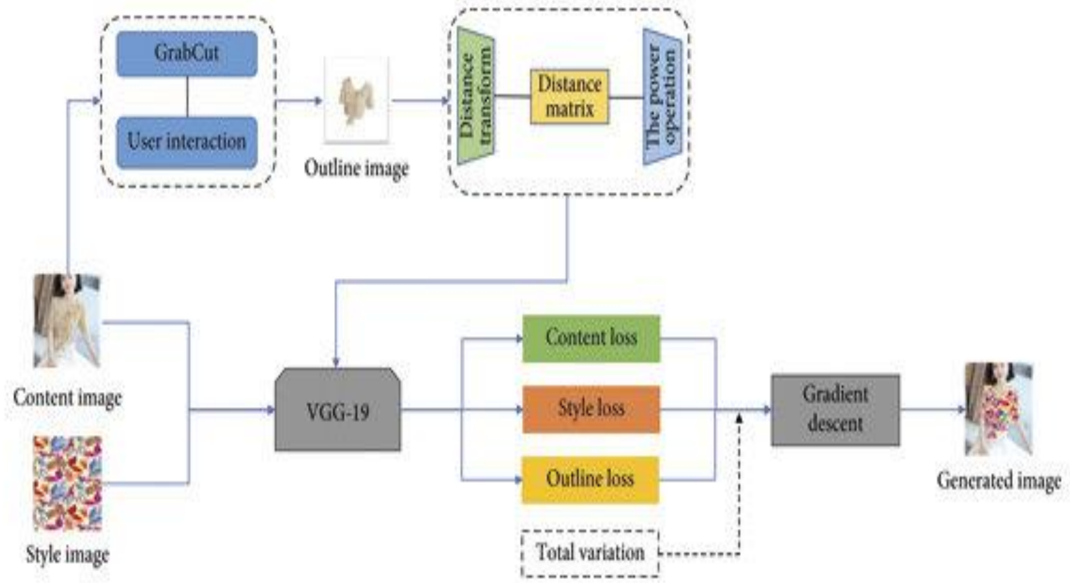

Fig. 4.2.1: System Architecture

The final output is a **stylized image** that successfully merges the subject and structure of the content image with the artistic qualities of the style image. Compared to traditional NST methods, the inclusion of the **outline loss module** provides clearer textures, sharper edges, and reduced distortions, resulting in improved visual quality. The architecture demonstrates how deep learning, feature extraction, and iterative optimization can be combined into a single pipeline to create high-quality artistic transformations.

CHAPTER 5

UML OVERVIEW

5.1 Overview

The Unified Modeling Language (UML) overview provides a structured representation of the design and functional workflow of the Neural Style Transfer (NST) system. It highlights the interaction between external actors (such as users and datasets) and the system's internal modules that perform style extraction, content preservation, and image synthesis. The diagram and descriptions capture how input images are processed through convolutional neural networks (CNNs) like VGG-19, how losses are computed, and how optimization generates the final stylized output. This section focuses on use case interaction, sequential workflow, and module-level architecture to illustrate the NST system as both a research framework and a practical image stylization tool.

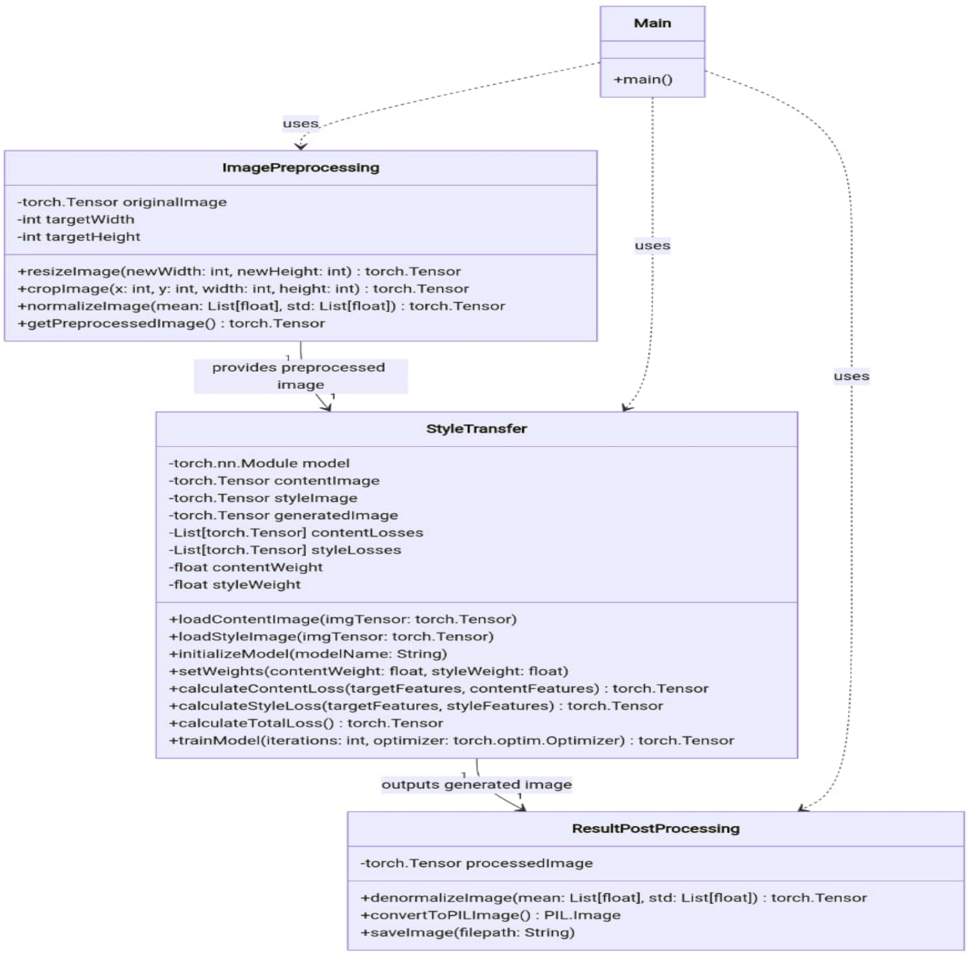

Fig. 5.1.1: UML Overview

5.2 Use Case Representation

The use case model outlines the key interactions between the system and its users. The user acts as the primary actor, providing two input images: a content image (which preserves the structure of the output) and a style image (which provides artistic texture and color). The system extracts content features and style representations using a pre-trained CNN (e.g., VGG-19). The optimization engine then applies loss minimization (content loss, style loss, and total variation loss) to iteratively refine a generated image. The final stylized image is presented back to the user.

Thus, the system enables the user to transform ordinary photos into artwork-like images through AI-powered stylization.

5.3 System Workflow

The workflow of Neural Style Transfer begins with data acquisition, where the user provides the content and style images. These images undergo preprocessing, including resizing, normalization, and tensor conversion. Next, a pre-trained CNN model (VGG-19) is used to extract feature representations at different layers — deeper layers capture content, while shallower layers capture texture and style.

The extracted features are used to compute **content loss** (to preserve structure) and **style loss** (to replicate textures and colors via Gram matrices or enhanced spatial maps). A **total loss function** is defined as the weighted sum of content and style losses, and an optimization algorithm (e.g., L-BFGS or Adam) iteratively updates the pixels of a generated image. The process continues until convergence, producing a final stylized image that combines the content of the original with the artistic style of the reference image.

5.4 Module-Level Architecture

The Neural Style Transfer system is organized into several interdependent modules, each serving a key role:

- **Input Module**: Accepts user-provided content and style images.
- **Preprocessing Module**: Handles image normalization, resizing, and tensor preparation for CNN input.
- **Feature Extraction Module**: Uses a pre-trained CNN (VGG-19) to extract content and style features.

- **Loss Calculation Module**: Computes content loss, style loss (via Gram matrices or spatial maps), and total variation loss.
- **Optimization Module**: Iteratively updates the generated image using gradient descent until the loss function converges.
- **Output Module**: Produces the final stylized image for the user.

In summary, the UML overview highlights how Neural Style Transfer transforms raw input images into artistic outputs through structured modules. By combining CNN-based feature extraction, custom loss functions, and iterative optimization, the system enables users to generate visually appealing, stylized images that blend structure and artistic texture seamlessly.

CHAPTER 6

ALGORITHMS

The main algorithm used in Neural Style Transfer is Gatys' Algorithm (2015). This method uses Convolutional Neural Networks (CNNs) such as VGG-19 to mix the *content* of one image with the *style* of another.

The CNN extracts two things:

- **Content features** – the shapes and structure of the content image.
- **Style features** – the colors, textures, and patterns of the style image.

To capture the style, the algorithm uses a Gram Matrix, which measures how features are related to each other.

The algorithm then creates a new image by minimizing a **loss function** made up of:

- **Content Loss** – keeps the structure of the content image.
- **Style Loss** – applies the patterns of the style image.
- **Total Variation Loss** – smooths the image and reduces noise.

By repeating this process, the algorithm produces an image that looks like the content image but painted in the style of the chosen artwork.

6.1 Convolutional Neural Network (CNN) Architecture

At the heart of NST lies a pre-trained **VGG-19 model**, chosen for its hierarchical feature extraction capability. CNNs process images through multiple layers, where:

- **Early layers** capture low-level features such as edges, textures, and color gradients.
- **Intermediate layers** capture shapes, contours, and structural patterns.
- **Deeper layers** capture high-level semantic content, such as object identity.

In the NST framework:

- **Content features** are extracted from deeper layers to preserve the overall structure and spatial layout of the content image.
- **Style features** are extracted from multiple shallower and intermediate layers to capture textures, strokes, and artistic details.

The CNN itself remains frozen (weights unchanged), serving purely as a feature extractor. Optimization is performed on the **generated image**, not the network.

Advantages in the framework:

- Hierarchical representation of both style and content.
- Strong transferability due to ImageNet pre-training.
- Separation of style and content features across layers.

6.2 Content Representation and Loss

The **content representation** ensures that the generated image retains the semantic structure of the input content image.

1. Content is extracted from deeper CNN layers (e.g., conv4_2).
2. **Content loss** measures the mean squared error (MSE) between the feature maps of the content image and the generated image.
3. This loss penalizes large deviations from the original structure, ensuring that the output preserves spatial alignment and recognizable forms.

Advantages in the framework:

- Maintains recognizability of objects and layout.
- Prevents distortion of content during stylization.

6.3 Algorithm Representation Diagram

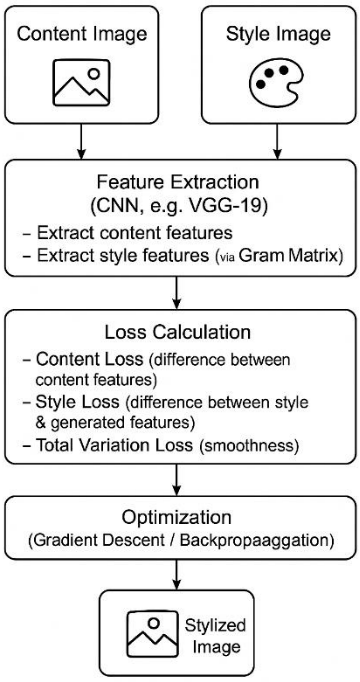

Fig. 6.3.1: Workflow Algorithm

6.4 Algorithmic Synergy

The power of NST comes from the **synergy of its components**:

- CNN provides robust feature extraction for both style and content.
- Content loss preserves structure, style loss transfers artistic texture, and TV loss ensures smoothness.
- Gradient descent optimization refines the generated image iteratively, balancing all constraints.

Together, these algorithms enable NST to merge structural fidelity with artistic creativity, creating outputs that are both visually compelling and semantically meaningful.

CHAPTER 7

TOOLS USED

The development and implementation of Neural Style Transfer (NST) requires a combination of deep learning frameworks, programming tools, and visualization utilities to achieve effective style blending, optimization, and output generation. The chosen tools support image preprocessing, convolutional neural network (CNN) feature extraction, loss function computation, and iterative optimization, while also enabling experimentation and result visualization. The selection of these tools was based on their proven performance in computer vision research, ability to handle large image data, and suitability for deep learning tasks involving convolutional networks such as VGG-19.

7.1 Programming Languages and Development Environment

- Python:
  Python was the primary programming language used due to its strong support for machine learning and deep learning libraries. It provides efficient implementations for tensor operations, gradient-based optimization, and image handling. Its ease of integration with frameworks like TensorFlow and PyTorch made it ideal for building and testing the NST model.
- JupyterNotebook:
  Jupyter Notebook was used during the experimentation phase, Jupyter Notebook provided an interactive coding environment where preprocessing, model training, and visualization of stylized outputs could be performed efficiently with real-time feedback.

7.2 Deep Learning and AI Frameworks

- TensorFlow:
  TensorFlow was used to implement the feature extraction process with the pre-trained VGG-19 model. Its GPU support and extensive API functions helped in defining custom loss functions (content loss, style loss, outline loss, and total variation loss) and optimizing the generated images.
- PyTorch:
  PyTorch was employed for experimental implementations of NST due to its flexibility

with dynamic computation graphs and easy debugging. It was especially useful for iterative optimization and real-time modifications during research.

- Keras:
  Keras, as a high-level API over TensorFlow, was used for quick prototyping of the NST pipeline, especially in defining network layers and performing transfer learning with pre-trained CNNs.

7.3 Data Handling and Storage

- Pandas:
  Although less critical than in structured datasets, Pandas was occasionally used for managing experiment logs, storing loss values, and comparing performance across multiple runs.
- NumPy:
  NumPy supported matrix operations essential for CNN feature extraction and optimization, enabling fast computation of Gram matrices and spatial transformations.
- HDF5/ImageStorage:
  Processed images and intermediate stylized results were stored in formats like HDF5 or PNG/JPEG, enabling efficient saving and retrieval during optimization.

7.4 Visualization Tools

- Matplotlib&Seaborn:
  These libraries were used to visualize the progression of loss functions, compare training curves, and present before-and-after stylization results in a clear and interpretable format.
- Plotly:
  For interactive visualizations, Plotly was employed to compare style transfer effects under different parameters, enabling deeper exploration of results during experimentation.
- OpenCV:
  OpenCV was used for image preprocessing tasks such as resizing, normalization, and edge extraction (e.g., for outline loss computation). It also supported output rendering of final stylized images.

7.5 Cloud and Computational Resources

- Colab:
  Colab provided free GPU and TPU resources, which were critical for training and testing NST models. Its scalability allowed multiple experimental runs and easy sharing of notebooks.
- AWS:
  AWS cloud services were used to train large-scale NST models and store high-resolution images. Its GPU-powered instances reduced training time significantly compared to local machines.
- Local GPU Workstations:
  For smaller experiments and debugging, local GPU-enabled systems were used to test different configurations of loss functions and learning rates.

7.6 Supporting Tools

- Scikit-image:
  Used for basic image transformations such as grayscale conversion, filtering, and edge detection before feeding images into the neural network.
- **SciPy**:
  Supported optimization functions and image smoothing processes during stylization.
- **OpenSSL & Data Security**:
  In scenarios where datasets or cloud resources were used collaboratively, encryption methods such as OpenSSL ensured the security of experimental files.

CHAPTER 8

BENEFITS AND LIMITATIONS

The proposed Neural Style Transfer (NST) framework represents an important advancement in the field of deep learning–based image processing. By combining convolutional neural networks (CNNs) with improved loss functions, NST enables the generation of visually appealing artistic images that merge the structure of a content image with the textures and patterns of a style image. The addition of an **outline loss module** and **enhanced style loss** makes the system more effective at preserving boundaries, reducing distortions, and improving the clarity of stylized outputs compared to traditional NST approaches. While the system offers numerous benefits for creative industries and digital applications, it also faces certain limitations that must be addressed to improve performance and usability.

8.1 Benefits

1. High-Quality Artistic Transformation

Neural Style Transfer can transform ordinary photographs into visually stunning artworks by applying the patterns, textures, and colors of famous paintings or chosen styles. The enhanced style loss improves the clarity of textures, reduces distortions, and produces results that look closer to real artistic works, making it a powerful tool for creative industries.

2. Preservation of Content Structure

A major benefit of this framework is its ability to maintain the original structure and layout of the content image. With the integration of outline loss and content preservation, the shapes, objects, and edges in the image remain sharp and recognizable. This ensures that even after artistic transformation, the essence of the original image is not lost.

3. Creative Applications Across Industries

NST is not limited to digital art but extends its applications into diverse industries such as advertising, photography, game development, film production, and fashion design. By enabling the transformation of visuals into unique styles, it provides designers and artists with new ways to enhance creativity, design promotional material, and create immersive visual effects.

4. Cultural and Educational Use

The technology also plays a role in cultural heritage preservation by digitally recreating old paintings or artworks with modern techniques. In educational contexts, it serves as a tool to demonstrate the power of artificial intelligence in creativity, making it easier for students and learners to understand the link between AI and art.

5. Flexibility and Customization

One of the strengths of NST lies in its flexibility. Users can adjust the weight of content loss, style loss, and outline loss to control how much structure or style dominates in the final output. This ability to fine-tune the balance allows for a wide range of artistic variations, catering to different creative requirements.

6. Integration with Emerging Technologies

With further optimization, NST has the potential to be integrated into virtual reality (VR), augmented reality (AR), and immersive media. For example, it could be used to apply artistic filters to live environments in AR or to create unique visual effects in VR games and applications, enhancing user experiences.

7. Encouraging AI-Based Creativity

Beyond its technical aspects, NST encourages a new perspective on creativity. It demonstrates how AI can act as a partner in human creativity rather than a replacement. By giving artists and designers new tools for expression, it expands the boundaries of digital art and inspires innovation.

8.2 Limitations

1. High Computational Cost

One of the biggest limitations of NST is the heavy demand for computational power. Running NST requires high-performance GPUs or TPUs, especially for high-resolution images. Without such resources, processing becomes very slow, making it less practical for everyday or large-scale use.

2. Slow Processing and Lack of Real-Time Use

Even with good hardware, NST can take several minutes to generate a single stylized image. This makes it unsuitable for applications that require real-time performance, such as live video editing, AR filters, or mobile-based creative tools.

3. Limited Style Diversity Handling

While NST works well with certain styles, it does not perform equally well with all types of art. Very abstract, noisy, or highly complex styles can lead to results that are unclear, chaotic, or far from the desired artistic effect. This reduces its versatility across diverse artistic preferences.

4. Dependence on Pre-Trained Networks

The system relies heavily on pre-trained convolutional neural networks like VGG-19, which were originally designed for object recognition, not style transfer. As a result, the approach inherits limitations from these models and cannot always capture artistic styles as naturally as a human artist might.

5. Artifacts in Large Images

When working with very large or high-resolution images, NST often generates artifacts such as broken textures, pixelation, or mismatched patterns. Additional refinement is needed to correct these issues, which further increases computational requirements.

6. Limited User Control

Although NST allows some customization, it does not offer precise control over how the style is applied to specific regions of an image. For example, users cannot easily choose to apply the style only to the background while keeping the subject unchanged without advanced masking techniques.

7. Storage and Energy Consumption

Running NST models repeatedly consumes significant memory, storage, and energy. This is a practical limitation for mobile devices or organizations with limited computational resources, raising concerns about sustainability and efficiency.

8. Scalability Challenges for Video

Applying NST to video introduces additional challenges, as each frame must be stylized individually. This often results in flickering or inconsistency between frames, making video-based NST computationally expensive and visually unstable.

9. Accessibility Barriers

Due to its high hardware requirements, NST is not easily accessible to everyone. Small creative studios or individuals without access to advanced GPUs may find it difficult to use, limiting its widespread adoption.

10. Potential Overuse or Misuse

Finally, while NST enhances creativity, it also raises questions about originality. The ability to automatically apply styles from famous artworks may blur the line between authentic human-made art and AI-generated content, leading to concerns about ownership, copyright, and artistic integrity.

CHAPTER 9

FUTURE SCOPE

Neural Style Transfer (NST) has shown remarkable promise in the field of computer vision and digital creativity, enabling the generation of artistic images that blend the content of one image with the style of another. While current methods already deliver impressive results, there are many opportunities for extending its capabilities and making it more practical for widespread applications. One of the key directions for future improvement is achieving real-time performance. At present, NST is computationally expensive and slow, but with the development of lightweight architectures, optimized algorithms, and hardware acceleration through GPUs, TPUs, and edge devices, NST could be deployed in real-time applications such as live video filters, augmented reality (AR), and virtual reality (VR).

Another important area of growth lies in enhancing **control and flexibility** for users. Future systems could allow selective style transfer to specific regions of an image (for example, applying style only to the background while preserving the subject), adjusting style intensity more precisely, or even blending multiple styles in a single output. Such advancements would make NST more versatile for creative industries, education, and professional design workflows. In addition, **multi-style transfer** and adaptive style blending are expected to become more advanced, giving users the ability to generate highly personalized artistic effects.

Expanding the **application scope** of NST beyond digital art is another promising direction. With improvements in robustness and efficiency, NST could play an important role in video production, advertising, gaming, cultural heritage preservation, and interactive media. In particular, its integration into mobile applications and cloud-based platforms could democratize its use, making it accessible to a wider audience. Furthermore, NST may find utility in scientific visualization, medical imaging, and educational tools where stylistic enhancements can help highlight important features or present data in more engaging formats.

Looking further ahead, NST could evolve through integration with **emerging AI paradigms** such as generative adversarial networks (GANs), diffusion models, and explainable AI techniques. These would help address current issues such as artifacts, style inconsistency, and lack of interpretability, leading to more stable and reliable outputs. Moreover, by incorporating advances in unsupervised and federated learning, NST models could be trained on broader datasets without compromising privacy, making them more generalizable and scalable.

In the long term, NST has the potential to become part of immersive **human–AI creative collaborations**, where artists and designers interact with AI systems in real time to co-create unique artworks and media. Coupled with developments in digital twin technology for simulation and testing, and aligned with ethical and copyright guidelines for responsible use of artistic styles, NST will continue to grow as a transformative tool at the intersection of technology, creativity, and culture.

CHAPTER 10

CONCLUSION

Neural Style Transfer (NST) represents a significant advancement in the intersection of deep learning, computer vision, and digital creativity. By leveraging convolutional neural networks (CNNs) such as VGG-19, NST enables the blending of the structural content of one image with the artistic style of another, producing outputs that capture both realism and creativity. The incorporation of an improved style loss function and spatial mapping enhances the clarity of textures and preserves fine details, addressing some of the limitations of the original approach proposed by Gatys et al. This refinement ensures that stylized images maintain the structural integrity of the content while delivering a visually appealing artistic effect.The design of NST emphasizes adaptability and creativity, making it a powerful tool for applications ranging from digital art and photography to film production, advertising, and virtual reality. Its ability to balance technical precision with aesthetic transformation showcases how artificial intelligence can move beyond traditional computational tasks into domains of human expression. Moreover, the flexibility of adjusting style intensity and combining multiple styles positions NST as a versatile framework capable of serving both professional designers and casual users.

Despite its strengths, NST also faces challenges such as high computational costs, slow processing times, and limited real-time performance. These issues currently restrict its scalability for broader applications such as live video processing and interactive media. However, ongoing research in lightweight models, generative networks, and optimized algorithms continues to reduce these barriers, indicating that NST will become more efficient and accessible in the near future.

In conclusion, Neural Style Transfer demonstrates the transformative potential of deep learning in creative fields. By continuously evolving through improvements in efficiency, interpretability, and user control, it is poised to become a core technology in digital content creation. As advancements in AI architectures, hardware acceleration, and real-time deployment continue, NST will play an increasingly vital role in shaping the future of art, design, and multimedia. It stands as a strong example of how artificial intelligence can collaborate with human creativity, opening new possibilities for innovation, personalization, and cultural expression.

REFERENCES

[1] K. Kanchanadevi and G. K. Sandhia, "A Neural Style Transfer Network for Image Augmentation of Corn Leaves using Convoluvational Neural Network," in *Proc. 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 2023. [Online]. Available: https://doi.org/10.1109/ICIICS59993.2023.10421717

[2] J. Liao, "A Study on Neural Style Transfer Methods for Images," in *2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, 2022, pp. 60–64. [Online]. Available: https://doi.org/10.1109/ICBAR58199.2022.00019

[3] C.-H. Yang, Y.-S. Liao, C.-R. Huang, and A. Dockhorn, "Explaining Neural Style Transfer [AI-eXplained]," *IEEE Computational Intelligence Magazine*, vol. 20, no. 3, pp. 75–76, Aug. 2025. [Online]. Available: https://doi.org/10.1109/MCI.2025.3569794

[4] M. Cotogni, M. Arazzi, and C. Cusano, "PhotoStyle60: A Photographic Style Dataset for Photo Authorship Attribution and Photographic Style Transfer," *IEEE Trans. Multimedia*, vol. 26, 2024. [Online]. Available: https://doi.org/10.1109/TMM.2024.3408683

[5] Z. Zhou and Y. Zhou, "Neural Style Transfer With Adaptive Auto-Correlation Alignment Loss," *IEEE Signal Processing Letters*, vol. 29, pp. 1027–1031, 2022. [Online]. Available: https://doi.org/10.1109/LSP.2022.3165758

[6] J. Li, Y. Xiang, H. Wu, S. Yao, D. Xu, et al., "Optimal Transport-based Patch Matching for Image Style Transfer," *IEEE Transactions on Multimedia*, 2022. [Online]. Available: https://doi.org/10.1109/TMM.2022.3201387