

FROM CLASSICAL FILTERS TO CONVOLUTIONAL NEURAL NETWORKS: THE MATHEMATICAL FOUNDATIONS AND EVOLUTION OF NEURAL STYLE TRANSFER IN IMAGES

MAHMOOD AMINTOOSI^{}

ABSTRACT. Neural Style Transfer (NST) stands as a remarkable example of the synthesis of mathematics, statistics, deep learning, and art. In this article, we present a historical and analytical exploration of the mathematical evolution of image-processing tools—from classical and derivative-based filters to frequency-domain transforms and multiscale representations—and trace how these concepts ultimately paved the way for convolutional neural networks, which form the foundation for understanding style transfer. We show how these mathematical developments culminated in the formulation of the NST algorithm. We then examine the mathematical structure of style transfer with a focus on the role of Gram matrices, learned feature spaces, and the optimization-based objective function. A central insight of NST is that its loss function becomes meaningful only through the representational power of deep neural networks: the notions of “style” and “content” are defined in learned feature spaces rather than in pixel space. Grounded in statistical concepts and solved through classical optimization methods, this formulation illustrates the deep and elegant interplay between applied mathematics, statistics, and deep learning in creating an artistic computational technique.

AMS Subject Classification (2020): Primary: code; Secondary: code.

Keywords: Neural style transfer, Convolutional neural networks (CNNs), Gram matrix, Frequency-domain transforms, Multiscale representations, Optimization, Feature spaces, Texture synthesis.

Article Type: Promotional Paper.

Communicated by Alireza Amini Harandi.

Received: 20-11-2025, Accepted: 20-06-2026, Published Online: xx-xx-202x.

Cite this article: M. Amintoosi, From classical filters to convolutional neural networks: the mathematical foundations and evolution of neural style transfer in images, *Math. Soc.*, **xx** (202x), no. x, x–x. <https://dx.doi.org/10.22108/msci.2026.147524.1771>.

1. Introduction

In recent years, the emergence of deep learning has fundamentally transformed the fields of computer vision and image processing. One of the most striking manifestations of this transformation is a technique known as *Neural Style Transfer*[9, 20] (NST)—an algorithm that leverages Convolutional Neural Networks (CNNs) to transfer the visual style of one image to another. At first glance, this process appears to be an artistic blend of color, texture, and form; however, at its core, it rests upon profound mathematical, statistical, and computational foundations. These include signal decomposition into frequency components, spatial and statistical filtering, and correlation analysis in feature spaces.

The fundamental question that Neural Style Transfer addresses is: *How can the style of an image be separated from its content, and then combined with the content of another image?* Answering this question was not feasible with classical image processing tools, because a formal definition of “style” and “content” required an understanding of multi-layered feature spaces that only became possible with the advent of Convolutional Neural Networks. In this framework, the mathematical structure of style transfer is defined through an objective function with two main components: a content loss function and a style loss function. The latter is defined via the Gram matrix, which measures statistical correlations between features across different layers of the network. This objective function is designed such that minimizing it through gradient descent methods produces an image that preserves the content of the original image while adopting the style of the reference image. This elegant fusion of mathematical, statistical, and optimization concepts with deep learning represents a rare example of the application of mathematical theory, programming, and experience in computer science.

The goal of this paper is to illuminate, through a historical and analytical lens, the mathematical foundations that gave rise to Neural Style Transfer. This article is of a *scientific-popular* nature, written for a reader familiar with basic mathematical and statistical concepts but not necessarily with machine learning, image processing, or neural networks. Therefore, the main emphasis is on demonstrating the connection between classical mathematical knowledge and modern technology, rather than presenting a novel research analysis or a comprehensive survey of all style transfer methods. To this end, we first review classical image representation and filtering tools, then examine concepts such as convolution, wavelet transforms, and Principal Component Analysis (PCA) as traditional data representation and feature extraction tools. Subsequently, we explain the structure of convolutional networks and their role in hierarchical image representation, and finally present the Neural Style Transfer algorithm and related topics.

2. Discussion

The central argument developed throughout this article is that Neural Style Transfer should be understood as the culmination of a long mathematical evolution in image representation theory. The modern NST framework did not emerge in isolation; rather, it arose from the convergence of several distinct traditions in applied mathematics, signal processing, statistical modeling, optimization theory, and machine learning which will be discussed in the following sections.

2.1. Mathematical Foundations of Image Representation and Classical Filters. Understanding certain fundamental concepts in classical signal and image mathematics facilitates comprehension of the behavior of Convolutional Neural Networks in image processing. Many novel ideas in CNNs are, in fact, continuations of principles developed decades ago in electrical engineering and applied mathematics. In this section, we briefly review these foundations so that we can understand the mathematical place of Neural Style Transfer in this evolutionary path.

Mathematical Representation of Images. A digital image can be defined as a two-dimensional function of spatial coordinates:

$$(2.1) \quad f(x, y) : \Omega \subset \mathbb{Z}^2 \rightarrow \mathbb{R}^c$$

where Ω is the image domain and c is the number of color channels. From a mathematical perspective, every image can be considered a linear combination of basis functions:

$$(2.2) \quad f(x, y) = \sum_n \alpha_n \phi_n(x, y)$$

where ϕ_n are basis functions and α_n are combination coefficients. The choice of these bases constitutes the essential difference among various image representations, from Fourier and wavelet expansions to data-driven representations in deep learning.

Classical Filters and the Concept of Frequency. Filtering is the first step in image processing, aiming to separate structural components of an image based on their degree of spatial variation (frequency). In the simplest classification, filters are divided into two major groups:

Low-Pass Filters. These filters remove rapid variations in intensity and are used for image blurring and noise reduction. The most famous among them is the Gaussian filter:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where σ controls the degree of smoothing. The convolution of this kernel with the image removes fine details.

High-Pass Filters. Unlike low-pass filters, these focus on details and rapid intensity variations, particularly for edge detection. A well-known example is the Laplacian operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2},$$

which in practice is implemented with discrete kernels such as:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Prior to Laplacian, derivative-based operators such as Sobel and Prewitt were designed to approximate partial gradients of the image. These filters compute derivatives in horizontal and vertical directions, and their combination yields the gradient magnitude:

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}.$$

The quantity $|\nabla f|$ indicates the intensity of spatial variation and assumes high values in edge regions where sudden brightness changes occur. Thus, the gradient map provides an index of the “structural boundaries” of the image.

Composite filters such as the Laplacian of Gaussian (LoG) are constructed by smoothing the image with a Gaussian filter and then applying the Laplacian operator. This filter is effective in detecting regions with drastic changes (edges and salient points), and by applying it at different scales, a hierarchical structure of image features can be obtained: fine details at small scales and coarser structures at larger scales.

In Convolutional Neural Networks, a similar structure emerges: in early layers, filters similar to derivative-based and LoG filters are learned automatically, while deeper layers form more complex and abstract representations of data. Thus, CNNs can be considered the evolutionary continuation of the same hierarchical idea in image analysis.

Frequency Perspective: From Classical to Data-Driven Bases. Following spatial and derivative filters, a complementary perspective developed in mathematics and signal processing that enables image analysis in the frequency domain. In this view, rather than being merely a collection of pixels, an image is considered a combination of oscillatory components at various scales and directions. The Fourier Transform, Discrete Cosine Transform (DCT), and Wavelet Transform are among the most important tools of this analysis. These transforms provide sets of *orthogonal bases* such that any signal or image can be reconstructed through their linear combination.

In Fourier and DCT transforms, each basis represents a component with a specific frequency, and the image energy is primarily concentrated in the first few components—a property that underlies JPEG



compression. In contrast, the Wavelet Transform enables multiscale analysis; meaning that global structures appear at large scales while fine details emerge at smaller scales. This multiscale idea later inspired versions of Neural Style Transfer where style is applied in specific frequency bands[27].

With the advent of deep learning, these same mathematical principles were reinterpreted in a data-driven framework. In Convolutional Neural Networks, filters are no longer predetermined; rather, the network learns mappings during training that automatically extract the most useful information from data. Each layer of the network can be considered a set of nonlinear mappings that transfer data to a new feature space; a space in which representations are organized hierarchically, from edges and textures in early layers to conceptual structures in deeper layers. Thus, the historical trajectory of image analysis can be traced from analytical and orthogonal bases (such as Fourier and wavelets) to learned, data-driven mappings (in neural networks). This conceptual transition establishes a connection between classical mathematics, statistics, and deep learning, providing the theoretical foundation necessary for understanding the mathematics of modern algorithms such as Neural Style Transfer.

Data-Driven Bases: From PCA to Neural Networks. In the twentieth century, a fundamental idea emerged: instead of defining bases analytically, one could learn them directly from data. Principal Component Analysis (PCA) is one of the earliest methods for learning such bases. In this method, data are first centered by subtracting their mean, and then directions in space are selected that capture the maximum variability or variance of the data. These directions, called principal components, are the axes that best reveal the hidden statistical structure of the data.

The result of this process is a compact representation of data in which each sample can be described as a combination of the mean and a few principal components. This approach has various applications in different fields, including face recognition and video background detection. For instance, in face recognition, these components are known as *Eigenfaces*, and every face image is a linear combination of the mean face and a number of these learned bases. The importance of this perspective lies in the fact that the concept of “learning bases from data” was later extended in deep neural networks; such that early layers of convolutional networks automatically learn bases similar to PCA or wavelet components, while higher layers extract more complex and semantic representations. This conceptual connection shows the transition from classical statistical methods like PCA to deep learning methods—a transition that paved the way for modern style transfer methods.

From Machine Learning to Deep Learning. In classical machine learning approaches, features were usually hand-designed, for example through methods such as PCA, edge detection filters, or gradient-based descriptors like Histogram of Oriented Gradients (HOG). However, in deep learning, the network itself automatically and hierarchically extracts features from data, eliminating the need for manual filter design.

The more important difference is that methods such as wavelets[1] or PCA [2] usually operate on the entire image as a single global unit, whereas Convolutional Neural Networks apply filters to local regions of the image and then, through successive layers, build a structure from simple to complex features. This local property gives networks tremendous power in pattern recognition and spatial detail combination—a capability that did not exist in classical methods. In CNNs, each layer in fact learns a new basis for data representation: early layers identify edges and colors, middle layers identify textures and local structures, and deeper layers identify high-level concepts. From this perspective, convolutional networks can be considered the natural extension of the data-driven basis idea, but with nonlinear, hierarchical, and optimization-based representational power. This fundamental transformation laid the groundwork for defining objective functions such as the style transfer function—a function that only has meaning in the space of internal deep network representations[6].

2.2. Convolutional Neural Networks and Feature Extraction. As we saw in the previous section, in classical methods such as PCA or wavelet transforms, bases are defined linearly and globally, treating the entire image as a single data unit. While useful for global analysis, this approach cannot adequately represent local patterns and artistic style textures. In contrast, Convolutional Neural Networks operate on the principle of “local receptive fields”: each filter sees only a small portion of the image and extracts features from that local region. In subsequent layers, these local features combine to build larger and more abstract structures. Thus, the network hierarchically represents everything from edges and colors to objects and high-level concepts. This capability for learning nonlinear, local, and multilayer representations constitutes the theoretical foundation of modern Neural Style Transfer methods.

General Structure of Convolutional Networks. Convolutional Neural Networks have become the primary tool in computer vision and image processing over the past decade. These networks consist of several main layer types: convolutional layers for extracting local features, activation functions for nonlinear transformations, and pooling layers for dimensionality reduction and increased stability.

Convolutional Layer. In the simplest case, each convolutional layer extracts local features by applying a filter K to the input image I :

$$(2.3) \quad (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) K(m, n).$$

In practice, cross-correlation is often used, but the role of both is similar: filters move across the image and reveal local patterns such as edges and textures. The important advantage of this structure is weight sharing, which ensures that the number of parameters remains independent of input dimensions.

Activation Function. After each convolution layer, a nonlinear function such as ReLU (Rectified Linear Unit) is usually applied:

$$(2.4) \quad \text{ReLU}(x) = \max(0, x),$$

which removes negative values and enables learning of nonlinear relationships. Without such nonlinear functions, the entire network would merely be a linear mapping.

Pooling Layer. For dimensionality reduction and increased robustness to spatial variations, pooling layers are used. In max pooling, the maximum value in a region is selected, while in average pooling, the mean of values is taken. This operation gradually focuses the network on more important structures. Hierarchical Feature Learning. One of the fundamental characteristics of Convolutional Neural Networks is hierarchical feature learning. In these networks, each layer effectively learns a new basis for data representation:

- Early layers respond to simple patterns such as edges and colors;
- Middle layers transform these patterns into composite shapes and textures;
- Deeper layers represent high-level semantic structures such as objects and scenes.

From a mathematical perspective, the behavior of each layer can be described by:

$$(2.5) \quad f_l = \sigma_l(W_l * f_{l-1} + b_l),$$

where f_l is the output (features) at layer l , W_l are learnable weights, b_l is the bias, and σ_l is the activation function. This process creates a kind of multilevel decomposition of data, reminiscent of multiscale transforms (such as wavelets and LoG), but with the crucial difference that in CNNs, the bases are learned from the data itself.

Pre-trained networks such as VGG[23], ResNet[11], AlexNet[18], and DarkNet[22] are commonly used in style transfer algorithms. These networks, despite differences in depth, connection type, and layer structure, all rely on the principle of gradual learning of internal representations. The size of these models is typically very large, involving millions of parameters that are tuned during the learning process.

Intuitive Understanding of Feature Learning. Visualizing filters and features of different layers provides important intuitive understanding of how Convolutional Neural Networks operate. In early layers, filters often resemble classical image processing filters (such as Sobel or Gabor filters) and respond to edges, intensity changes, and simple textures. However, in middle and deeper layers, these filters gradually represent more complex patterns and high-level concepts (such as face parts, body components, or geometric structures of objects).

One of the most influential studies in this area is the work of Zeiler and Fergus[28] in their paper “Visualizing and Understanding Convolutional Networks.” They introduced a mechanism called

DeconvNet to map internal activations of each layer back to image space, demonstrating what each layer of the network “sees.” Using this method, it became clear that the hierarchical structure of the network—from edges to complex concepts—is in fact a gradual organization of features. This intuitive understanding also plays an important role in explaining the separation of content and style in Neural Style Transfer networks.

Image Representation in CNN Feature Spaces. From a mathematical perspective, a Convolutional Neural Network can be viewed as a sequence of mappings:

$$(2.6) \quad \Phi = \Phi_L \circ \Phi_{L-1} \circ \cdots \circ \Phi_1,$$

where Φ_l is the mapping operation from the output of layer $l - 1$ to layer l . The features extracted from image I at layer l are written as:

$$(2.7) \quad F^l = \Phi_l(I).$$

These features hierarchically transform from local and textural representations in early layers to semantic and conceptual representations in deeper layers. To measure content similarity between two images I_1 and I_2 , one can measure the Euclidean distance between their representations at a specific layer (l_c):

$$(2.8) \quad d_{l_c}(I_1, I_2) = \|\Phi_{l_c}(I_1) - \Phi_{l_c}(I_2)\|_2.$$

This metric forms the basis for defining “content similarity” in Neural Style Transfer.

According to the analysis of Gatys and colleagues[9], different layers of the VGG network provide representations with different natures from the image. In the early stages of the network (such as conv1_1 and conv2_1), filter responses are highly sensitive to local changes and fine textures; while in deeper layers (such as conv4_2 and conv5_2), pixel details fade away and only the overall structure and object arrangement are preserved[9, 14]. Therefore, features from higher layers represent “content” and lower layers represent “style.”

In the original Neural Style Transfer implementation, Gatys and colleagues used layer conv4_2 for content representation and layers conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1 for style representation. Combining multiple style layers enabled the network to match texture statistics at various scales (from fine details to overall structure), resulting in a smoother and more natural style transfer effect.

2.3. Mathematical and Statistical Definition of Style in Images. In the previous sections, we saw how classical image processing tools—from low-pass and high-pass filters to wavelet and Fourier transforms—shaped our understanding of visual structure and patterns. Now we examine how, within the framework of deep neural networks, the concept of “style” is defined and mathematically modeled.

Neural Style Transfer bridges statistical mathematics and deep learning, allowing the “statistical features” of an image to be recognized as its style.

Understanding the concept of “style” in the context of Convolutional Neural Networks requires first understanding how images are internally represented in these networks. As mentioned, studies by Zeiler and Fergus[28] demonstrated that filters at different layers in CNNs represent patterns ranging from the simplest structures (edges and local textures) to more abstract concepts at higher levels. Within this framework, Gatys and colleagues discovered that the statistical relationships among these activations constitute the essence of what we perceive as the “style” of an image.

From Feature Map to Style Representation. Consider an input image I passed through a CNN, and at one of the middle layers, the output or feature map is obtained as $F \in \mathbb{R}^{C \times H \times W}$. Here C is the number of channels (or filters), and H and W are the height and width of the feature map, respectively. For computational convenience, this three-dimensional tensor can be reshaped into a two-dimensional matrix $F \in \mathbb{R}^{C \times M}$ with $M = H \times W$; that is, each row represents the responses of one filter at all spatial positions.

Gram Matrix: The Mathematical Language of Style. Gatys and colleagues were the first to demonstrate that statistical correlations between filter responses can adequately represent the style structure of an image. These correlations are defined in the form of Gram matrices. For the feature matrix $F \in \mathbb{R}^{C \times M}$, the Gram matrix $G \in \mathbb{R}^{C \times C}$ is defined as:

$$(2.9) \quad G = FF^T, \quad G_{ij} = \sum_{k=1}^M F_{ik}F_{jk}.$$

Each entry G_{ij} expresses the degree of correlation between filters i and j . The diagonal elements G_{ii} reflect the power of each filter’s response, and off-diagonal elements reflect the co-occurrence or covariation of filters. From a statistical perspective, the Gram matrix can be considered an uncentered approximation of the covariance matrix of features.

From an intuitive statistical perspective, the Gram matrix shows the correlation between responses of different filters in a layer. If filter i has high values at locations where filter j is also active, the corresponding entry G_{ij} will be large; meaning these two filters are often excited simultaneously. This co-occurrence can indicate the presence of a composite pattern in the image. For example, suppose one filter is sensitive to striped patterns and another detects orange color. If in an image these two features appear simultaneously and in similar regions, the value G_{ij} between these two filters increases; indicating that the simultaneous occurrence of stripes and orange color specifies a particular texture or pattern (such as tiger skin).

This insight forms the intuitive explanation of Gatys and colleagues’ style transfer method [9]: the style of an image can be described by statistical relationships between filters that respond to colors,

textures, and different structures. In other words, style is hidden not in the exact spatial values of pixels, but in *patterns of co-activation of filters*. As Zeiler and Fergus showed, filters at different levels of the convolutional network represent everything from edges and colors in early layers to more composite concepts in deeper layers. Therefore, computing the Gram matrix from multiple layers provides a multiscale view of statistical correlations among these concepts.

In summary, the mathematical definition of style in convolutional networks is a combination of these statistical relationships across different layers, represented as a vector or set of Gram matrices. This representation bridges statistical mathematics and visual perception in neural networks and forms the basis of modern Neural Style Transfer methods.

Texture Generation with the Gram Matrix. One of the first and most beautiful applications of the Gram matrix idea in Convolutional Neural Networks is artificial texture generation[10]. In this method, the goal is to construct a new image that is statistically and perceptually consistent with a given texture sample, using only that sample image. For instance, if the original image shows pebbles, similar images should be generated; or patterns similar to tile designs and wallpapers can be produced based on a single sample image.

First, the sample texture image I_{texture} is passed through several layers of a CNN, and the Gram matrices $\{G_l^{\text{target}}\}$ corresponding to each layer are computed. These matrices record the correlation among active filters in each layer and thus compactly represent the “style” statistics of the image at different scales. Next, a random initial image (such as white noise) is selected as input, and through repeated passes through the network and pixel value updates via gradient descent, it is gradually transformed into an image whose Gram matrices match the target values. In other words, the overall loss function is defined as a combination of Gram differences across several layers:

$$(2.10) \quad \mathcal{L}(\vec{x}, \hat{x}) = \sum_l w_l \|G_l^{\text{current}} - G_l^{\text{target}}\|_F^2$$

where w_l is the weight of each layer and $\|\cdot\|_F$ is the Frobenius norm.

This algorithm demonstrates that using only correlations among features (without needing initial geometric structure), complex and natural visual textures can be reproduced. Despite the fundamental role of the Gram matrix in style representation, subsequent research showed that this method faces challenges in some cases (for example, neglecting the spatial arrangement of features and reproducing repetitive textures). Therefore, alternative methods such as Adaptive Instance Normalization (AdaIN) and approaches based on linear transforms have been proposed, each establishing a balance among quality, speed, and content preservation.

2.4. From Texture Generation to Neural Style Transfer. After understanding the statistical concepts hidden in the Gram matrix, we now turn to the main problem: how can the content of

one image be combined with the style of another? The algorithm introduced in 2016 by Gatys and colleagues was a fascinating application of statistics, mathematical optimization, and Convolutional Neural Networks. In this method, no model is trained; rather, the image pixels themselves are directly optimized to simultaneously satisfy two conflicting objectives: preserving content structure and reflecting style features.

The main goal of the Neural Style Transfer algorithm is to find an image x that, on one hand, preserves the content of image p (such as a city view or human face), and on the other hand, reflects the style of image a (such as a Van Gogh painting or mosaic pattern). Thus, the final image is a kind of compromise between two spaces: the space of structural features and the space of statistical correlations.

Mathematical Formulation. The overall loss function is defined as a weighted combination of two parts:

$$(2.11) \quad \mathcal{L}_{\text{total}}(p, a, x) = \alpha \mathcal{L}_{\text{content}}(p, x, l_c) + \beta \sum_{l \in \mathcal{L}} w_l \mathcal{L}_{\text{style}}(a, x, l)$$

where:

- l_c is the content layer and $\mathcal{L}$ is the set of style layers;
- Coefficients α and β determine the emphasis on content and style.

From an intuitive perspective, this function is a multi-objective optimization problem that establishes a balance between “image geometric structure” and “feature correlation statistics” through content loss and style loss components.

Content and Style Losses. Content loss is computed from the direct difference of features extracted from a specific layer of the network (for example, conv4_2 in VGG-19):

$$(2.12) \quad \mathcal{L}_{\text{content}}(p, x, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

where F_{ij}^l are the features of the current image and P_{ij}^l are the features of the content image at layer l .

Style loss is defined based on the same Gram matrix introduced in the previous section:

$$(2.13) \quad \mathcal{L}_{\text{style}}(a, x, l) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

where N_l is the number of filters and $M_l = H_l W_l$ is the number of spatial points in layer l . This metric in fact measures the difference between correlation structures of features in the current image and the style image. From a statistical perspective, content loss attempts to preserve mean features, while style loss aligns covariances.

Optimization Process. In this algorithm, the decision variable is the output image itself, not the network weights. The CNN is kept fixed, and only the pixel values of the input image x are changed to minimize the overall loss function. The process usually begins with an initial image (the content image or white noise), and then the gradient of the loss function with respect to pixels is computed:

$$(2.14) \quad \frac{\partial \mathcal{L}_{\text{total}}}{\partial x} = \alpha \frac{\partial \mathcal{L}_{\text{content}}}{\partial x} + \beta \sum_l w_l \frac{\partial \mathcal{L}_{\text{style}}^l}{\partial x}$$

Pixel updates are performed using standard optimization methods such as L-BFGS, Adam, or SGD. The final result is an image that simultaneously reflects the structure of the content image and combines the texture and coloration of the style image. This algorithm is not only the starting point for many modern style transfer methods, but also an instructive example of how statistical thinking and mathematical concepts can be combined with deep learning structures.

2.5. From Optimization to Statistical Mappings: The Evolution of Style Transfer. The algorithm of Gatys and colleagues, while conceptually elegant and perceptually very successful, is computationally expensive. Computing Gram matrices for each layer and executing hundreds of optimization steps for each image makes this method unsuitable for real-time applications. These challenges motivated subsequent research to present faster and more practical versions while preserving the statistical foundation of the original method[15, 16, 24].

One of the key directions in this path was replacing iterative optimization with direct mappings. In these approaches, instead of gradually changing pixels, a feedforward network is trained to produce the output image in a single pass. A notable example is the work of Johnson and colleagues[17], which using “perceptual loss” based on CNN features, reduced style generation time from several minutes to a few hundredths of a second while preserving artistic quality.

The next step in this evolution was the introduction of “Adaptive Instance Normalization” (AdaIN), presented by Huang and Belongie[12]. In this method, instead of computing the Gram matrix, the mean and standard deviation of content image features are replaced with corresponding values from the style image. This simple yet effective idea enables statistical alignment of content and style without complex optimization and provides real-time arbitrary style transfer. Its important feature is that with a single network, controls such as balance between content and style, interpolation among multiple styles, and color and position adjustment are provided.

Simultaneously, other methods such as Whitening and Coloring Transform (WCT) were developed. This method, inspired by linear transforms such as PCA, whitens the covariance of content features and then “colors” them with style covariance to achieve similar statistical compatibility. Subsequently, works such as Laplacian-Steered Style Transfer, by adding Laplacian loss, helped preserve content details and reduce artificial effects[21].

Comprehensive reviews[15, 16, 24] have shown that Neural Style Transfer has transformed from an artistic phenomenon to a practical tool in various domains, from design and artistic content production to video applications, fashion, and virtual reality. However, challenges such as maintaining temporal coherence in videos and quantitative evaluation of perceptual quality remain open and active research topics. At present, the primary method of comparing results is visual comparison among different methods.

Another wave of research in this domain has focused on utilizing Diffusion Models. These models, by generating images through gradual noise removal, have achieved unprecedented quality and diversity in style transfer. For example, the “Style Injection in Diffusion” method [8] enables style transfer without additional optimization, merely by manipulating self-attention layer features in large-scale diffusion models. Approaches based on Generative Adversarial Networks (GANs) also remain efficient and competitive, particularly for specialized domains such as illustration visualization [26]. In the latest developments, methods based on Reversible Networks[19] have taken a step beyond previous approaches and, using parameter fusion strategies, have simultaneously improved content preservation and style fidelity. These developments show that the evolution of style transfer from classical methods toward deep generative models continues to expand rapidly.

Thus, the evolutionary path of Neural Style Transfer (NST) has moved from direct solution of an optimization problem toward statistical mappings and learnable models. In all these versions, the mathematical essence remains constant: statistical alignment among layer-wise network representations, which builds a bridge between classical signal processing and deep learning. In fact, Gatys’s objective function can be considered an example of combining two fundamental domains: *analytical optimization in mathematical space* and *statistical learning in feature space*. Subsequently, methods based on perceptual loss and statistical normalization showed that even with significant simplifications in mathematical structure, the same concept of balance between content and style can be maintained in a faster and more generalizable form.

3. Conclusions

Neural Style Transfer can be considered not merely a deep learning achievement, but a reinterpretation of fundamental mathematical concepts in a data-driven framework. In this framework, concepts such as convolution, gradient, and covariance matrix—which have roots in twentieth-century mathematics—are recreated in the structure of deep networks to model the statistical and geometric relationships hidden in image space.

In recent years, the fundamental idea of Neural Style Transfer has inspired new branches in computational art. In the domain of “Neural Painting,” this same theoretical framework is employed to

simulate the painting process with brushes and color mixing[13, 30]. Also, methods such as CartoonGAN and CycleGAN[7], by extending the concept of statistical feature alignment, enable targeted recreation of specific styles such as painting, comic, or anime, and have been used in recognition problems as a data augmentation solution[5]. This trend shows that the idea of Gatys and colleagues not only transformed the mathematical understanding of “style,” but also opened new horizons at the border between art and deep learning.

Today, style transfer has transcended the boundaries of art and is employed in domains such as medical imaging, industrial design, virtual reality, data compression, and aesthetic analysis. Despite these advances, numerous theoretical and computational challenges remain; including more precise definition of style space, design of perceptual metrics aligned with human judgment, and development of models capable of hierarchically and interpretably combining multiple styles. Furthermore, novel research is examining the connection between neural networks and multiscale analysis tools such as wavelet and Laplacian transforms, to merge mathematical interpretability with data-driven learning power in a unified framework.

From a broad perspective, the evolutionary path of Neural Style Transfer reflects the larger movement in image processing science: the transition from analytical and theoretical classical models toward learning-based and data-driven systems; a movement that, at the same time, preserves the ancient connection between *mathematics, statistics, computer science, and art*.

REFERENCES

- [1] M. Amintoosi, M. Fathy and N. Mozayani, Regional varying image super-resolution, *In: IEEE international joint conference on computational sciences and optimization*, **1**, (2009), 913–917.
- [2] M. Amintoosi and F. Farbiz, Eigenbackground revisited: can we model the background with eigenvectors?, *J. Math. Imaging Vision*, **64** (2022), no. 5, 463–477.
- [3] M. Amintoosi, Application of Taylor expansion in reducing the size of convolutional neural networks for classifying Impressionism and Miniature paintings, *Math. Soc.*, **5** (2020), no. 1, 1–16. [In Persian]. https://math-sci.ui.ac.ir/article_25351.html.
- [4] M. Amintoosi, Fully connected to fully convolutional: road to yesterday, *Journal of soft computing and information technology*, **11** (2022), no. 1, 60–72. [In Persian]. https://jscit.nit.ac.ir/article_149453.html.
- [5] M. Amintoosi, Style transfer for data augmentation in convolutional neural networks applied to fire detection, *Computational intelligence in electrical engineering*, **13** (2022), no. 4, 97–114. [In Persian]. https://isee.ui.ac.ir/article_26042.html.



- [6] T. Y. Chiu and D. Gurari, PCA-based knowledge distillation towards lightweight and content-style balanced photorealistic style transfer models, *In: 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, (2022), 7834–7843, <https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00769>.
- [7] Y. Chen, Y. K. Lai and Y. J. Liu, CartoonGAN: generative adversarial networks for photo cartoonization, *In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, (2018), 9465–9474. <https://doi.org/10.1109/CVPR.2018.00986>.
- [8] J. Chung, S. Hyun and J. P. Heo, Style injection in diffusion: a training-free approach for adapting large-scale diffusion models for style transfer, *In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (2024), 8795–8805.
- [9] L. A. Gatys, A. S. Ecker and M. Bethge, Image style transfer using convolutional neural networks, *In: 2016 IEEE conference on computer vision and pattern recognition (CVPR)*, (2016), 2414–2423.
- [10] L. A. Gatys, A. S. Ecker and M. Bethge, Texture synthesis using convolutional neural networks, *In: Proceedings of the 29th international conference on neural information processing systems*, **1**, NIPS’15, Cambridge, MA, USA: MIT Press; (2015), 262–270.
- [11] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, (2016).
- [12] X. Huang and S. Belongie, Arbitrary style Transfer in real-time with adaptive instance normalization, *In: Proceedings of the IEEE international conference on computer vision (ICCV)*, (2017), 1510–1519. <https://doi.org/10.1109/ICCV.2017.167>.
- [13] A. Hertzmann, Can computers create art?, *Arts*, **7** (2018), no. 2, pp. 25. <https://www.mdpi.com/2076-0752/7/2/18>.
- [14] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu and M. Song, Neural style transfer: a review, *IEEE Transactions on Visualization & Computer Graphics*, **26** (2020), no. 11, 3365–3385. <https://doi.org/10.48550/arXiv.1705.04058>.
- [15] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu and M. Song, Neural style transfer: a review, *IEEE transactions on visualization and computer graphics*, **26** (2020), no. 11, 3365–3385. <https://doi.org/10.1109/TVCG.2019.2921336>.
- [16] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu and M. Song, *Neural style transfer: a review*, (updated 2022 version), (2022). arXiv preprint arXiv:1705.04058. <https://arxiv.org/abs/1705.04058>.
- [17] J. Johnson, A. Alahi and L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, *In: Proceedings of the european conference on computer vision (ECCV)*, (2016), 694–711. https://doi.org/10.1007/978-3-319-46475-6_43
- [18] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *In: F. Pereira, Burges CJC, L. Bottou, K. Q. Weinberger, editors, Advances in neural information processing systems 25*, Curran Associates, Inc., (2012) 1097–1105.

- [19] Y. T. Kao, H. J. Lin, K. J. Lin and Y. Tokuyama, Content-preserving image style transfer via reversible networks with meta ActNorm, *Electronics*, **15** (2026), no. 2, 395. <https://doi.org/10.3390/electronics15020395>.
- [20] X. Li, S. Liu, J. Kautz and M. H. Yang, Learning linear transformations for fast image and video style transfer, In: *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (2019), 3804–3812.
- [21] S. Li, X. Xu, L. Nie and T. S. Chua, Laplacian-steered neural style transfer, In: *Proceedings of the 26th international conference on world wide web (WWW)*, (2017), 1123–1132. <https://doi.org/10.1145/3038912.3052617>.
- [22] J. Redmon and A. Farhadi, YOLO9000: Better, Faster, Stronger, CVPR, 2017.
- [23] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, In: *International conference on learning representations*, (2015).
- [24] A. Singh, V. Jaiswal, G. Joshi, A. Sanjeeve, S. Gite and K. Kotecha, Neural style transfer: a critical review, *IEEE Access*, **9** (2021), 60297–60311. <https://doi.org/10.1109/ACCESS.2021.3112996>.
- [25] A. Vidhya, The world through the eyes of CNN, 2020. Accessed: (2025). <https://medium.com/analytics-vidhya/the-world-through-the-eyes-of-cnn-5a52c034dbeb>.
- [26] X. Wang and J. Jiang, Illustration image style transfer method design based on improved cyclic consistent adversarial network, *PLOS ONE*, **20** (2025), no. 1, e0313113.
- [27] J. Yoo, Y. Uh, S. Chun, B. Kang and J. W. Ha, Photorealistic style transfer via wavelet transforms, In: *IEEE/CVF international conference on computer vision (ICCV)*, (2019), 9035–9044.
- [28] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, In: *European conference on computer vision (ECCV)*, Springer, (2014), 818–833.
- [29] Y. Zheng, C. Yang and A. Merkulov, Breast cancer screening using convolutional neural network and follow-up digital mammography, In: A. Mahalanobis, A. Ashok, L. Tian and J. C. Petrucci, editors, *Conference: Computational Imaging III*, **10669**, International society for optics and photonics, SPIE, (2018). <https://doi.org/10.1117/12.2304564>.
- [30] Z. Zou, T. Shi, S. Qiu, Y. Yuan and Z. Shi, Stylized neural painting, In: *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, (2021), 15684–15693. <https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01543>.

Mahmood Amintoosi

Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, P.O.Box 1159, Mashhad, 91775, Iran. m.amintoosi@um.ac.ir



Mahmood Amintoosi is an Associate Professor of Computer Science at Ferdowsi University of Mashhad. He received his B.Sc. and M.Sc. degrees in Mathematics (Applied in Computer) and Computer Engineering (Software), respectively, from Ferdowsi University, and his Ph.D. in Computer Engineering (Artificial Intelligence) from Iran University of Science and Technology. His research interests include machine learning, computer vision, and combinatorial optimization. The slides related to this paper are available in the *Presentations* section of the author's homepage at <https://mamintoosi.github.io/>.