Optimizing Artistic Synthesis: An Analysis of Pre trained Convolutional Neural Network Layer Selection for Neural Style Transfer

Bipasha Iasmin*
Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh

Rafit Hosen
Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh

Shammi Hossain Mou
Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh

Mahima Binta Mosharof
Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh

Abhijit Pathak
Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh

ABSTRACT
Neural Style Transfer (NST) has revolutionized digital art by synthesizing the content of one image with the artistic style of another image, relying fundamentally on feature extraction from pre-trained Convolutional Neural Networks (CNNs). A critical, yet often heuristically determined, factor in this process is the selection of specific CNN layers for content and style representation, which significantly affects the final aesthetic quality. This study presents a systematic experimental investigation of the effect of layer selection on the perceptual quality of stylized images. We compared the performance of the established VGG19 architecture with the more modern and efficient EfficientNet-B0, as well as a Custom-Designed CNN, across various combinations of content and style layers. The rating is based on three important criteria: color palette transfer, visibility of the artistic technique (e.g., brushstrokes), and level of detail generalization. By objectively evaluating these criteria, this study aims to provide real, non-heuristic guidance for optimizing creative synthesis. The findings offer valuable insights for researchers and practitioners, enabling the informed selection of network architectures and layer configurations to achieve superior and predictable artistic outcomes in Neural Style Transfer applications.

Keywords
Neural Style Transfer (NST), Convolutional Neural Networks (CNN), VGG19, EfficientNet-B0, Layer Selection, Artistic Synthesis, Perceptual Quality, Deep Learning

1. INTRODUCTION
Human visual perception is crucial for understanding the patterns, shapes, and relationships around us, and it helps the brain quickly identify objects and understand the context, even in visually complex environments. This perceptive skill also determines how artists develop and evaluate their work. Through their cognition, artists creatively simplify, enhance, and transform visual stimuli. While reducing elements in visual content, the brain automatically filters information deemed crucial or relevant to the artwork, thereby guiding the viewer's attention to specific features or qualities. This selective processing shapes the distinctive artistic style cultivated by the individual. Neural networks emulate this layered feature extraction through hierarchical structures. Owing to this resemblance, deep learning models can computationally replicate artistic transformations. Understanding human perception can help explain why neural style transfer produces visually meaningful results. This connection provides a significant motivation for investigating the synthesis of styles using modern neural architectures.

The structure of the human brain has long served as an inspiration for the development of artificial neural networks. Researchers have discovered that biological neurons learn patterns through repeated exposure to stimuli. This finding prompted the development of early computational models that attempted to imitate similar learning behaviors in humans. As the field has developed, scientists have discovered that hierarchical representations can capture increasingly complex aspects of the data. Convolutional theories were inspired by the visual cortex, where distinct cells respond to specific edges and patterns. This biological understanding directly influences the construction of the convolutional neural networks. Over time, a better understanding of the brain has led to more complex architectures with deeper and more efficient brain layers. These enhancements considerably improved the models' capacity to evaluate and synthesize visual data. Such developments directly increase the effectiveness of neural style transfer, which largely depends on the accurate analysis of both material and style patterns. Ultimately, recent breakthroughs in creative and generative AI systems have been enabled by brain-inspired computing.

The evolution of neural networks has progressed from simple linear models to complex deep architectures. Rosenblatt's perceptron in 1958 marked an early step toward enabling computers to learn simple decision boundaries. Although limited to a single layer, it laid the foundation for subsequent machine learning developments. In 1974, Steven W. Werbos introduced a backpropagation algorithm that allows the training of multilayer networks. This enables deeper architectures and sophisticated concept representation. With increased computing power, networks can handle larger and higher-dimensional datasets. In the early 2000s, Hinton and Salakhutdinov advanced hierarchical feature learning and automated feature extraction. These developments have

improved the depth and efficiency of neural networks. Today, neural networks are widely applied in vision, pattern recognition, and creative tasks, such as Neural Style Transfer. By extracting content and style features from images, they demonstrated the transformative potential of deep learning in research and industry applications.

The advent of Convolutional Neural Networks (CNNs) has revolutionized the process of analyzing and generating images. Earlier neural models failed to localize spatial patterns in images. CNNs solve this issue by employing convolutional layers to obtain local characteristics of an image. This architecture allows them to recognize certain features, such as edges, textures, and shapes of pixels. This has significantly increased the dependency on manually designed features to a large extent. The first paper written by LeCun et al. (1998) demonstrated the high potential of CNNs. Most existing deep learning procedures are based on research. With the growing volume of data and more sophisticated hardware, CNN architectures have become deeper and more robust. This progress has enabled highly precise image recognition and generative processes.

Convolutional Neural Networks (CNNs) have emerged as a critical technique in modern computer vision, dramatically advancing the analysis and interpretation of visual inputs. As a result, CNNs are among the most common applications of image-based machine learning. Their layered nature enables models to gradually acquire complex features, allowing them to succeed in jobs requiring high accuracy and specific design identification. CNNs have professional applications in fields such as medical imaging, remote sensing, and automatic diagnostics. Within this expanding research landscape, it is essential to differentiate between image analysis and integration methods. Image resolution focuses on interpreting observable guidelines and supplementary tasks such as detection, classification, and segmentation. These supports depend on responsible feature extraction and the use of robust computational models. Image integration, on the other hand, involves fertilization or transformation of images, which is complementary to areas such as texture creation, style transfer, and digital effects. Rather than analyzing material visuals, synthesis techniques produce new outputs that are influenced by learned patterns. This difference highlights the dual influence of CNNs on both the understanding and creation of images. As research progresses, CNNs are central to future developments in computational imaging.

Image processing in a manner similar to the human brain became possible after 1998, when Yann LeCun and his team proposed the first model of a convolutional neural network (CNN). Today, convolutional networks have become the primary tools for image analysis and synthesis. Image analysis refers to the process of extracting information from images. The result is not an image but data in numerical or symbolic form. Examples of image analysis applications include medical processing to aid diagnosis, pattern recognition in quality control systems, and facial recognition in surveillance systems. Image synthesis refers to the generation of new images based on input data, including the images. Examples include the generation of weather maps, textures for computer games, and 3D objects for scientific simulations.

This study focuses on a specific image synthesis technique known as Neural Style Transfer (NST). Introduced by Gatys, Ecker, and Bethge in 2015, NST is an optimization technique that generates a new image by simultaneously minimizing two distinct loss functions: one for the content of a photograph and another for the artistic style of a separate artwork [7]. The resulting image is a unique blend thatretains the content structure of the photograph while adopting the texture, color palette, and brushstrokes of the style image.

NST has found practical applications across various fields, including

- *Photo Reconstruction:* Applying historical or artistic styles to modern photographs.
- *Multimedia and Entertainment:* Creating stylized video content and special effects.
- *User Interface (UI) Design:* Generating unique, stylized textures and backgrounds.
- *Medical Image Processing:* Standardizing or enhancing the visual appearance of medical scans for better analysis [8].

The foundational NST algorithm relies on a pretrained CNN, typically VGG-19, to extract the necessary feature representation. A critical, yet often heuristically determined, aspect of this process is the selection of specific convolutional layers for content and style representation. This study presents an experimental investigation into the use of two distinct pretrained CNNs, the established VGG19 and the more modern and efficient EfficientNet-B0, for NST.

The core contribution of this study is the systematic evaluation of how the selection of convolutional layers for content and style representation affects the perceptual quality of the generated images. Specifically, this study will evaluate three key criteria: Color Palette Transfer, Visibility of Artistic Technique (e.g., brushstrokes), and Level of Detail Generalization. By comparing the performance of VGG19, EfficientNet-B0, and a Custom-Designed CNN across various layer combinations, this study seeks to provide empirical guidelines for optimizing artistic synthesis.

2. LITERATURE REVIEW

Alexandru et al. (2022) presented a unified framework for object recognition, geometric modifications, and neural style transfer. This increases the artistic expressiveness and flexibility of visual stylization. Their pipeline starts with object detection using YOLOv5, which recognizes and localizes objects in the content image by assigning them bounding boxes. Once objects are detected, the system selects suitable style images and applies geometric warping to each detected object individually, a process inspired by previous deformation-based style transfer methods. After warping, each deformed object is composited back into the original image using blending techniques, and finally, style transfer is applied to ensure consistency of the artistic style across the image. This method offers several advantages over traditional style transfer approaches. First, treating each object separately preserves the structural integrity of complex images while allowing per-object geometric deformation, thereby providing greater artistic flexibility. Second, because it integrates object detection, warping, and style transfer in a single pipeline, the framework can handle multi-object scenes and produce stylized outputs within a reasonable timeframe, which is a benefit over the computationally expensive iterative methods. However, the authors acknowledge these limitations. The final quality heavily depends on the alignment between the content and the chosen style objects; poor matches or large class differences can lead to unsatisfactory warping or distortion. Additionally, because matching and warping are performed separately for each object, objects from different classes may not warp logically, reducing general applicability. Sparse or weak matches further compromise deformation and overall visual coherence. The authors suggested that better heuristics for

style-object matching and improved warp-style alignment could enhance consistency and broaden applicability in future studies [1].

Li et al. (2022) presented a deep learning–based approach that employs convolutional neural networks (CNNs) and object segmentation to apply high-quality styles to images while preserving the original content. The image was first divided into foreground and background regions, and a pre-trained Detectron2 model generated unique masks for each identified object. The SANet function was then applied to each segmented item individually, preserving the detail and integrity of the original structure throughout the process, including the style transfer. After stylization, each object is placed over the original image using a mask. Consequently, the finished image is consistent with stylistic alterations while preserving the original content. According to research, this strategy improves color brightness and item identification in multi-object images. When there is only one item, the texture and fine details are enhanced by the proposed method. Segmentation allows for a more precise application of style while effectively preserving the outlines and minute details. The algorithm also demonstrated real-time efficiency, making it suitable for various applications. However, limitations persist in this study. The Detectron2 model does not always correctly classify all items, leading to partial stylization in some cases. Furthermore, localized style application might weaken the contrast and cause small irregularities. The authors suggest that using content-aware importance masks can dynamically adjust the style strength based on object relevance, thereby improving the consistency and visual quality. Overall, Li, Vyas, and Penta (2022) present a robust framework that effectively balances artistic style and content structure, which offers valuable insights for advancing arbitrary style transfer techniques [2].

Gatys et al. (2015) established the basis for current neural style transfer by showing that Convolutional Neural Networks (CNNs) can efficiently distinguish and alter an image's content and style representations. Their work developed the concept that an image's structural information (content) and artistic patterns or textures (style) can be treated as distinct characteristics. Using Gram matrices to collect style data, they demonstrated that it is possible to create high-quality creative images by mixing the content of one image with that of another image. This innovation has facilitated a wide spectrum of artistic transformations and has sparked further research. Research following Gatys et al. has continually emphasized the method's capacity to produce visually pleasing and diverse artistic results. CNN-based style transfer achieves a strong and effective separation of content and style, allowing the duplication of the aesthetic characteristics of great artworks while retaining the core structure of the original content images. The success of this method has greatly aided breakthroughs in neural image synthesis and creative AI applications. However, the optimization-based technique proposed by Gatys et al. has several limitations. Iterative optimization requires significant processing resources, which makes real-time or interactive transmission difficult. The high computational complexity often limits the use of this technology in low-latency application. Additionally, the resulting images can sometimes show subtle distortions that reduce photorealism, especially when the style requires smooth textures or very fine details. Nevertheless, despite these limitations, CNN-based style transfer is still used as a highly effective method for creating artistic images and has made significant contributions to research in computer vision, graphics, and AI-based creativity [3].

Bhanu Duggal et al.2024 - Style Transfer (NST) is a method of creating images by combining the content of one photo and the style of another photo. Traditionally, NST techniques have used fixed amounts of content and style in the calculation of both content and style losses during image generation, which can cause images generated with these methods to place too much emphasis on the content or style. Recent advancements in NST include the introduction of dynamite weighting for the calculation of content and style loss, as well as multi-scale loss calculations for content/style pairing. Through the use of dynamic weighting, the amount of content/style affects the optimization process, enabling more visually balanced results. Multiscale loss calculations allow the loss calculation to occur on both coarse (large, low-resolution) and fine (small, high-resolution) structure levels simultaneously, thereby maintaining both types of visual detail. Additionally, many different optimization techniques (i.e., progressive scaling and the ability to efficiently use pre-computed images) have been proposed to assist in managing the increased computational burden created by multi-scale loss calculations. The proposed techniques produce images of higher quality, greater detail retention, and reduced chances of overemphasizing the content or style of an image. However, many limitations still exist concerning computational requirements, generalizability to all types of images and styles, and sensitivity to the hyperparameters. Additionally, this study did not provide sufficient data to compare the new NST techniques with previously existing NST techniques. In conclusion, using dynamic weighting with multi-scale loss calculations can be a viable means of creating high-quality stylized images while balancing the fidelity of the content with the richness of the style component [4].

Rhythm Bhardwaj et al.2024- Style Transfer (NST) has evolved considerably since the introduction of the first NST model developed by Gatys et al. The first NST model only used Convolutional Neural Networks (CNN) to extract the content and style statistics of an image, but was computationally intensive and resulted in distorted outputs. Newer NST methods, such as adaptive instance normalization (AdaIN), have improved the computational efficiency of NST compared with the first NST method. AdaIN is approximately 99.92% faster at processing images than the first NST model and still produces high-quality stylization of the original content. Comparative studies of feature extractors have found that VGG-16 and VGG-19 extract similar style-loss values after training for over 1000 epochs; however, VGG-19 has a higher content-preserving power, which means that it extracts more information about the original content. The study showed a 73.1% difference in the content loss between the two models. Thus, it is advisable to use VGG-19 for stylizing detailed and structured tasks. Recently, researchers have begun to explore the use of multiscale techniques and wavelet transforms to improve the realism of the images produced using NST methods. The use of these techniques has allowed for better handling of textures and reduced the number of defects or artifacts created during the photorealistic style transfer. Another major advancement in the evolution of NST methods is the development of semantic segmentation as an additional component of the NST model. Semantic segmentation allows the selective application of styles from the source image based on object regions, leading to more semantically consistent results. The drawbacks previously associated with NST methods, such as temporal instability and spatial artifacts, which often require additional processing for correction, still exist. Overall, the NST methods of today are much more efficient at extracting content and applying a photographically

realistic style to images, thus indicating the evolution of NST methods from the original optimization-based models [5].

Ruta et al. (2022) introduce HyperNST, which is a neural style transfer model that combines hyper-networks with the StyleGAN2 structure to stylise artistic images of high quality, especially portraits. The algorithm uses the ALADIN metric space directions and semantic facial region maps to maintain the content structure and provide flexibility in artistic inputs, including semantic editing and interpolation between the semantic style codes. They show state-of-the-art content preservation at fast feed-forward inference, although at the expense of huge computational requirements, which are approximately 22 GB VRAM on a single batch and approximately 24 h of convergence for training. Although it has advantages, HyperNST is weakened by the lack of effective methods to encode actual portrait images into the StyleGAN2 weight space and the reliance on a relatively low 256×256 output resolution. The optimization process is still resource-consuming, and its generalizability outside portrait styles has not been studied. In addition, the effect of various semantic segmentation masks on the quality of stylization has not been studied in detail, and the next step is to refine this issue [6].

Li et al. (2024) re-implement image-based, fast, and arbitrary Neural Style Transfer (NST) and present activation smoothing to a ResNet architecture in an attempt to improve the stylization results. They indicate that the art of seamless transitions is a sure method of enhancing the quality of visualization and that smoothing activations can render ResNet superior to VGG-based approaches in specific situations. However, the authors are also careful to point out existing weaknesses, which include the high computational cost of iterative image-based NST, the difficulty in maintaining the balance between content and style being low, and a variety of failures, especially when it comes to the SWAG model. The research gaps identified by the study are the reduced stylization performance of advanced or lightweight networks, limited information on the performance of various smoothing approaches in various architectures, and unknown reasons that cause stylization failures [7].

Geralne, Raad, Lezama, and Morel (2022) proposed a method of neural style transfer that operates under the resolution and overcomes the computational and memory constraints of previous methods. They operate their strategy based on the spatial localization of the computation in the VGG network so that forward and backward passes can be executed on local image areas, and global optimization is still carried out. This framework enables multiscale style transfer at very high resolutions and can capture stylistic features at a broad scale, including the color structure, subtle brushstrokes, and texture of the canvas. The authors state that the method provides state-of-the-art visual quality when transferring a high-resolution painting style, which is supported by extensive qualitative demonstration and quantitative evaluation. Notably, this method retains image fidelity without the trade-off inherent in the speed-up and resolution-boosting methods. These strengths do not imply that the technique is no longer computationally intensive; the resolution that can be reached continues to be constrained by the large memory requirements of a GPU, indicating that issues of scale are not fully solved. This study fills a significant research gap: traditional neural style transfer algorithms are associated with large computational costs, memory limits, and reductions in quality at scale. Despite the fact that the proposed approach is a significant breakthrough in terms of high-resolution style transfer, further work is needed to expand its applicability to a wider variety of artistic styles [8].

Seyed, Cansever, and Hart (2025) presented a partial convolution method for forward-masked style transfer. Instead of applying masking afterward, the style is embedded immediately into the selected region of the image. This helps the model capture style cues more faithfully but also reveals the recurrent problem of artifacts at mask boundaries. To soften these changes, the authors integrated three blending techniques: mask feathering, mask expansion, and content-feathering. Tests on 500 SA-1B images showed that combining all three methods yielded the most natural results, with content feathering having the strongest effect on reducing the border separation. Quantitative metrics, such as the EMD and perceptual style loss, consistently favor partial convolution methods over traditional style transfer pipelines. Although some artifacts persist and the effectiveness of the method still depends on segmentation quality, this study fills a clear gap in region-specific style transfer and demonstrates a more consistent way of integrating stylized and non-stylized content into a single image [9].

Kashyap et al. (2025) researched the evolution of the neural style transfer field, highlighting that VGG19 feature extraction and reflection layer pick balance content stability and misleading sufficiency. They note the key limitations of the living technique, including long processing times, fixed accessibility of style images, constant style-weight ratios, and numerous compromises in content structure. Their experimental system, Dynamic Neural Style Transfer for Artistic Image Generation using VGG19, introduces a multiple optimization-based approach that accepts flexible justification of style weights while significantly reducing processing time. The model confection content morality bounds multiple styles and delivers consistently high-quality outputs. Both qualitative and quantitative results confirm the method's effectiveness and improved adaptability for alternative artistic applications [10].

3. THEORETICAL FOUNDATIONS OF NEURAL NETWORKS

3.1 Artificial Neural Networks (ANNs) and the Neuron Model

The McCulloch-Pitts (MP) Neuron Model We [2] is one of the first formal modeling efforts for a neuron. This refers to a simple computational node that takes several binary inputs ($x_1,x_2,\ldots,x_n$)with individual weights ($w_1,w_2,\ldots,w_n$). The neuron sums the inputs of its inputs with some weights and then compares this sum to a (fixed) threshold θ. If the sum is greater than threshold, the neuron "fires" (outputs 1), and if not it's inactive (output 0)

$$W.X = \sum_{i=1}^{n} wi\ xi.$$

$$\text{Output} = \begin{cases} 1\ if\ w.x \geq \theta \\ 0\ if\ w.x < \theta \end{cases}$$

This model established the concept of a neuron as a simple threshold logic unit capable of performing basic logical operations.

3.2 General Network Architecture

Contemporary ANNs are composed of layers and constitute a directed graph through which data flow from the input to the output. Consider a standard deep network with three types of layers:

- *Input Layer:* It layer receives the input for processing (features of raw data pixel values of the image).
- Hidden Layers: Perform most of the 'thinking' for transforming the input data to something that the output layer can use. There are many such hidden layers in deep learning, which is why it's called "deep."

- Output Layer: This is the end of the network, a final output (e.g., classification label, generated image , etc.).

The concept of deep learning is closely related to the hierarchy of feature representation. Low level (edges, corners) while combining these simple features to learn more and more complex high-level representations (eyes, wheels, children). Such hierarchical content and style separation is crucial for tasks like NST, which exhibit certain notions of content and style at different levels of abstraction.

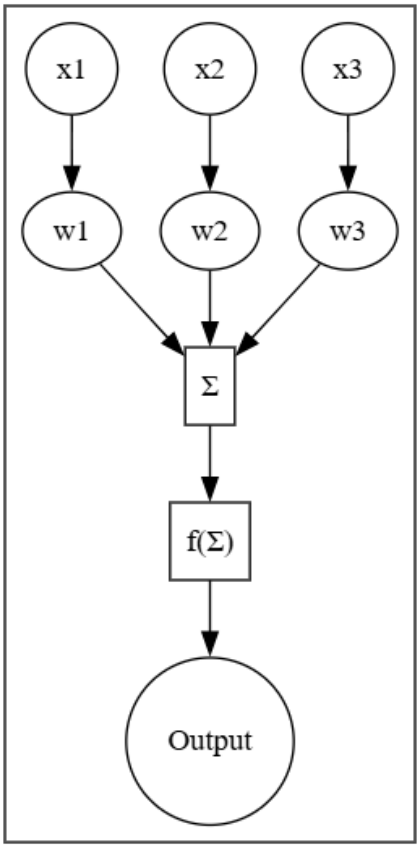

Fig 1: The McCulloch-Pitts neuron model

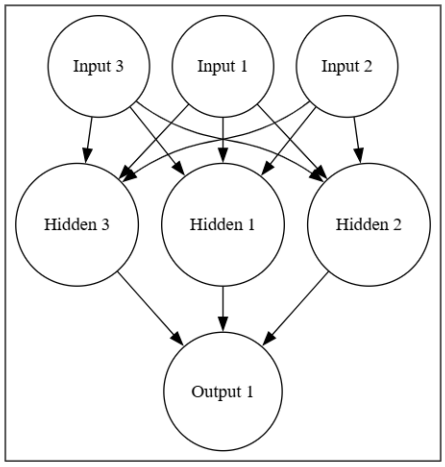

Fig 2: General Artificial Neural Network Architecture

3.3 Digital Representation of Graphic Data

For a neural network to process an image, the image must be converted into a numerical format.

3.4 Image Representation as Matrices

A digital image is fundamentally represented as a two-dimensional matrix of numerical values, where each element in the matrix corresponds to a pixel (picture elements).

Monochrome (Grayscale) Image Representation: On the other hand, a grayscale image can be described by only one 2D matrix. Such a pixel value usually varies between 0 (black) and 255 (white), indicating the amount of light present at this spot.

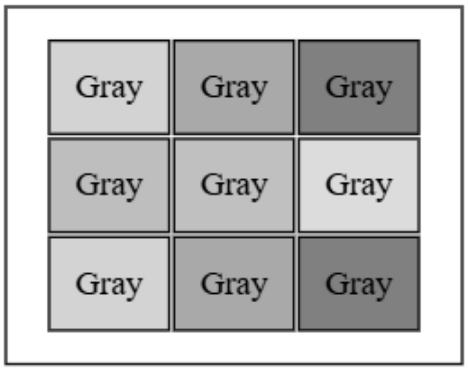

Fig 3: Monochrome (Grayscale) Image Representation

Color (RGB) Image Representation: RGB color images are composed of three 2D arrays corresponding to each of the primary channels: red (R), green (G), and blue (B). The value intensity-wise in three channels at a particular point of the pixel determines its color. This is cast into a 3D tensor.

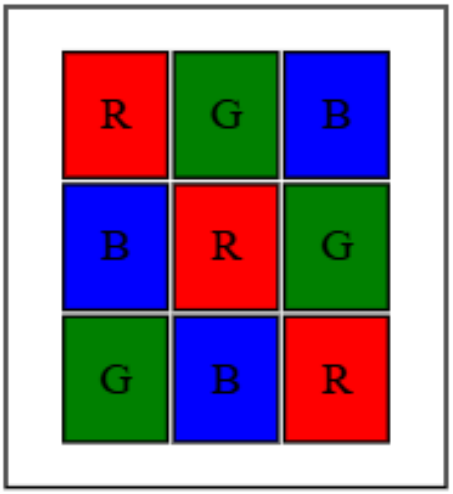

Fig 4: Color (RGB) Image Representation (Conceptual Placeholder)

3.5 Resolution and Detail

The resolution of an image refers to the total number of pixels in it. Higher resolution images have more "details" but you have to pay for that with a lot of computing time. The resolution-slowdown trade-off is especially important in deep learning problems, as doubling the pixel size increases the input size and computational difficulty by a factor of four.

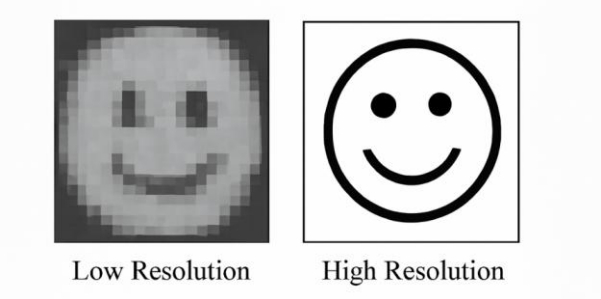

Fig 5: Impact of Resolution on Detail and Processing Time

4. IMAGE PROCESSING WITH CONVOLUTIONAL NEURAL NETWORKS (CNNs)

4.1 CNN Architecture and Components

CNNs are specifically designed to process data with grid-like topologies, such as images. Their architecture is built upon three primary types of layers that work sequentially to extract features.

Primary Layer Types

- *Convolutional Layer:* The core building block of a CNN. It performs a convolution operation by passing a small filter (kernel) over the input volume to create feature maps.
- *Pooling Layer:* A down-sampling operation that reduces the dimensionality of the feature maps, thereby reducing the number of parameters and computations in the network.
- *Fully Connected (FC) Layer:* Typically found at the end of the network, these layers connect every neuron in one layer to every neuron in the next, performing high-level reasoning based on the features extracted by the preceding convolutional and pooling layers.

Typical CNN Architecture Flow

A typical CNN architecture involves an alternating sequence of Convolutional and Pooling layers, followed by one or more Fully Connected layers.

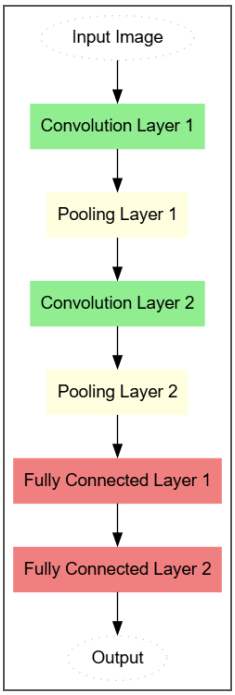

Fig 6: Typical CNN Architecture Flow

4.2 Early versus Deep in Feature Extraction

It is also the location in the network that indicates the complexities of the feature(s) it abstracts:

- *Early Layers (Shallow):* These layers are responsible for learning the lowest-level representational features, such as edges, lines, or color blobs. These features are very local and are almost model-independent. In NST, these layers encode the subtle texture of an image.
- *Deep Layers (Late):* This layer type models emphfurther layers on the same architecture, which learn high-level, abstract features, such as object parts (e.g., eyes, noses, and wheels) or complete objects. These features are the content of the image; they contain the overall structure and semantics.

4.3 The Convolution Operation

Convolution is an operation that the convolution layer is based on.

- **Receptive Field (RF) and Feature Detector**

The receptive field is a small region in the input volume that is connected to a neuron in the convolutional layer. The receptive field of a feature detector is small (a filter / kernel that is slid across the input volume). The weights of the kernel are tuned during training to identify a particular feature (e.g., a vertical edge).

- Mathematical Definition of Convolution

The convolution operation is defined as the integral of the product of the two functions after one function is reversed and shifted. In the discrete two-dimensional case for images, the output feature map S(i, j) is calculated as

$$S(i,j)=(I*K)(i,j)=m\sum n\sum I(i-m,j-n)K(m,n)$$

Where:

- I is the input image,
- K is the kernel (filter),
- i and j are the coordinates of the output feature map.

This formula defines how the convolution operation works by summing the element-wise product of the image I and kernel K as the kernel slides over the input image to compute each value in the output feature map $S(i,j)$.

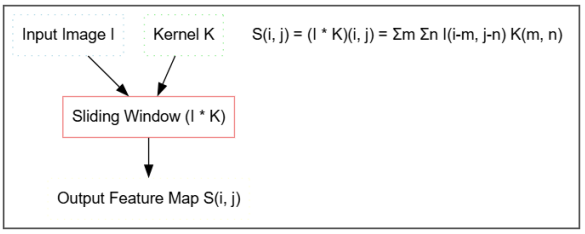

Fig 7: The Convolution Process

4.4 Generation of Feature Maps

The Feature Map (or activation map) is the result of this application, which consists of filters on the input image. This is a common practice in real-world classifiers because each feature map identifies the location of the specific type of feature of the given filter in the input. Multiple filters are typically used in a convolutional layer, which generates a feature map stack.

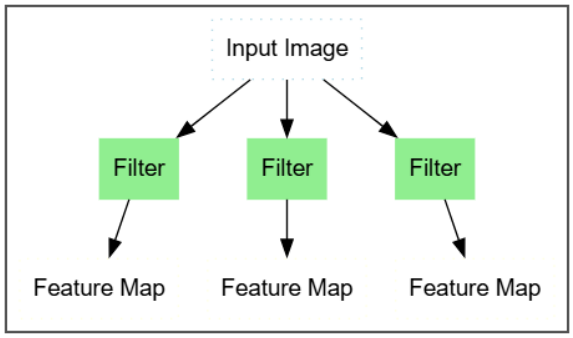

Fig 8: Generation of Feature Maps

4.5 The Pooling Layer and Activation

- **The Pooling Layer**

The Pooling Layer progressively reduces the spatial size of the representation to reduce the number of parameters and computations in the network, and hence control overfitting. It works separately on each feature map.

- **Max-pooling:** Most commonly used, selects the maximum value in the receptive field (e.g., 2x2 windows). Here, it maintains the most salient feature response from that region.
- **Average pooling:** Computes the average value of the receptive field.

The pooling operation is shift-invariant, that is, the network can still detect a feature even if the stimulus moves slightly.

- **Nonlinear Transfer Functions**

After the convolution step, we applied a nonlinear Activation Function pairwise element with the feature map. This is important, as without non-linearity, the deep network would reduce to a stack of linear transformations (that is, the same as a single linear layer).

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

This is the definition for the so-called ReLU function, which gives an input x back if it is positive or zero, and otherwise 0. The next element in the sequence, after the convolution operation, applies the ReLU function pointwise to the feature map, which means introducing nonlinearity in the model and allowing the neural network to learn high-level features.

5. THE MECHANISM OF NEURAL STYLE TRANSFER (NST)

5.1 General Scheme of Style Transfer

Neural Style Transfer, as introduced by Gatys et al. [7], is performed via iterative optimization. The objective is to produce a new image $\vec{x}$ that is close in content to the given p and style of a given $\vec{a}$.

Three main inputs are used in the process:

- *Content Image* (Photograph or image/whose composition needs to be preserved.)
- *Style Image* - The artwork with artistic characteristics (color, texture, brushstrokes) that you want to transfer.
- *Feature Extractor CNN:* A pre-trained convolutional neural network (CNN), such as VGG-19 (usually used) , acts as a fixed feature extractor. The weights of the network do not change during the course of performing style transfer.
- The reasons for the layered choice are a focal element in the NST mechanism. As the network is hierarchical, different layers represent different types of information.
- *Upper Layers for Content:* The higher layers of the CNN represent the high-level semantic content of the image. These layers contain feature maps that encode the object positions and overall structure. Thus, the content loss is commonly computed from a single deep layer.
- *Deeper as Style, Lower for Details:* Shallower layers represent local texture-like details, including edges, corners, and color patches. Combinations of feature responses across several layers allow the network to build complex textures and match the colors present in the artistic style given an input image. As a result, the style loss is generally computed over several layers closest in depth to the input (e.g., two or three layers), from shallow to medium depth (i.e., conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1 in VGG-19).

5.2 The Total Loss Function

The optimization objective in NST is to minimize the **Total Loss Function** (L_{total}), which is a weighted sum of the Content Loss ($L_{content}$) and Style Loss (L_{style})

$$L_{total}(\vec{p},\vec{a},\vec{x}) = \alpha L_{content}(\vec{p},\vec{a},\vec{x}) + \beta L_{style}(\vec{a},\vec{x})$$

Where:

- L_{total} is the total loss function , and
- $L_{content}$ is the content loss, measuring the difference between the content of the target image $\vec{x}$ and the content image $\vec{p}$.
- L_{style} is the style loss that measures the difference between the style of the target image $\vec{x}$ and style image $\vec{a}$.
- Where α and β are the weights that control the relative importance of the content and style losses , respectively .
- The parameters α and β are the weighting parameters that control the balance between content preservation and style application.
- A higher value of α relative to β will result in an image that closely preserves the content structure but may only lightly apply the style.
- A higher value of β relative to α results in an image with a strong, aggressive style application, often at the expense of content fidelity, leading to greater abstraction.
- The generated image $\vec{x}$ is initialized with random noise or the content image itself and is iteratively updated via back propagation and gradient descent to minimize L_{total} .

5.3 Content Loss ($L_{content}$)

The Content Loss measures the extent to which the generated image $\vec{x}$ deviates from the content image $\vec{p}$ in terms of high-level feature representation.

Let F_{ij}^{l} be the activation of the i-th filter at position j in layer l for the generated image $\vec{x}$, and P_{ij}^{l} be the corresponding activation for the content image $\vec{p}$. The content loss for a single layer l is defined as the Mean Squared Error (MSE) between the feature maps of the two images:

$$L_{content}(\vec{p},\vec{a},l) = \frac{1}{2}\sum_{i,j}(F_{ij}^{l} - P_{ij}^{l})^2$$

The gradient of this loss with respect to the generated image $\vec{x}$ is calculated and backpropagated through the network to update the pixel values of $\vec{x}$. By minimizing this loss, the generated image is forced to have the same high-level feature responses as the content image at the chosen layer, thereby preserving the structural content.

5.4 Style Loss (L_{style})

Style Loss measures the similarity between the style of the generated image $\vec{x}$ and the style image $\vec{a}$. Style is defined not by the absolute feature responses but by the correlations between the feature maps across different filters in a given layer.

5.5 The Gram Matrix

The Gram Matrix (G^l) is the key mathematical tool used to capture the style features. For a given layer l, the feature map $F^l$is a matrix of size $N_l \, x \, W_l$, where N_l is the number of filters (channels) in layer l, and M_l is the size of the flattened feature map ($H_l \, x \, W_l$). The Gram Matrix G^l is the inner product of the feature map F^l with itself:

$$G_{ij}^l = \sum_{k=1}^{Ml} F_{ik}^l - F_{jk}^l$$

Where $G_{ij}^l$the covariance between features is maps i and j in layer l. This matrix captures the texture and color information by measuring how often and where different features co-occur, regardless of their exact spatial location.

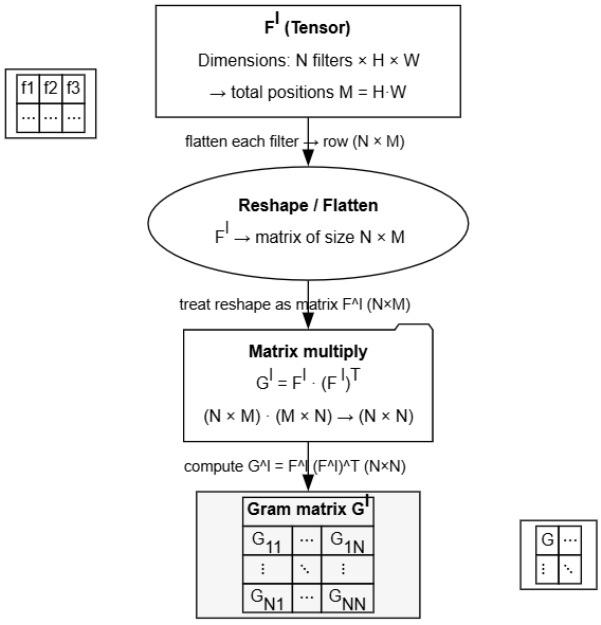

Fig 9: Transformation of Feature Maps to Gram Matrix

5.6 Style Loss Calculation

Let A^lbe the Gram Matrix of the style image $\vec{a}$ at layer l and $G^l\$$ be the Gram Matrix of the generated image $\vec{x}$ at layer l. The contribution of layer l to the style loss is the Mean Squared Error between these two matrices:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

The total style loss L_{style} is the sum of the style losses across a set of L_{layers}, weighted by a factor w_{ij}^l for each layer as follows:

$$L_{style}(\vec{a}, \vec{x},) = \frac{1}{2} \sum_{l=1}^{L} w_l E_l$$

By minimizing L_{style} , the generated image is optimized to have the same feature correlations (i.e., the same texture and style) as the style image across multiple scales.

6. EXPERIMENTAL ANALYSIS AND RESULTS

6.1 Pre-trained CNN Models Utilized

The experimental study employed three different CNN structures to explore the importance of layer selection for NST quality.

- **VGG19**

VGG19 [9] is a deep convolutional network with a homogeneous architecture of 19 layers (16 convolutional and the last three fully connected). It was pre-trained on the ImageNet [23] dataset and has been commonly used as a standard feature extractor with the NST algorithm to achieve a rich hierarchy of visual features. Its design consists of small 3 X 3 convolutional filters and max-pooling, which results in a very deep architecture that can carefully disentangle the content from the style representations.

- **EfficientNet-B0**

However, EfficientNet-B0 [10] is a much more modern compound-scaled network that focuses on high accuracy with fewer parameters and a lower computational cost than VGG-style networks. It leverages a mobile inverted bottleneck convolution (MBConv) and a squeeze-and-excitation optimization that applies to network scaling methods in depth, width, and resolution. Its use in this study aims to verify whether the modern and much more efficient architecture can achieve the same or higher levels of artistic synthesis as the traditional VGG19, due to the way its various blocks encode style and content.

- **Custom-Designed CNN**

We also used a Custom-Designed CNN architecture that was designed explicitly for style transfer. The network is shallower than VGG19 but deeper than the original EfficientNet-B0, so that we can control the attention on feature map sizes and channel counts more finely. The goal of this custom network was to investigate the influence of network depth and specific layer selection on the resulting aesthetic quality, and thus act as a baseline compared to the two general-purpose pre-trained models.

6.2 Experimental Methodology

- **Content-Style Image Selection**

A standard set of images was used to maintain consistency and comparability. The Content Image selected was a high-resolution photo of a cityscape with complex geometry and detail richness. The Style Images consisted of two different artistic styles: Van Gogh (e.g., The Starry Night) with thick and swirling brushstrokes and vivid colors, and a Da Vinci sketch (e.g., Study of a Head) with fine lines and monochrome shades.

- **Procedure for Varying Layer Selection**

The heart of the experiment consisted of using combinations of layers for content and style calculation in a systematic way across the three models. For each model, a total of nine layer combinations were analyzed, which included shallow and deep for both the content and style layers.

Model	Content Layer Combinations	Style Layer Combinations
VGG19	conv3_1, conv4_2, conv5_1	[conv1_1], [conv1_1, conv2_1],
EfficientNet-B0	block3a, block4a, block5a	[block1a],[block1a, block2a], [block1a to block5a]
Custom CNN	layer3, layer4, layer5	[layer1], [layer1, layer2], [layer1 to layer5]

- **Optimization Process and Hyper parameter Settings**

Optimization was carried out using the L-BFGS algorithm, which is widely applied to NST because of its rapid convergence [12]. The combined loss function (Eq. 2) was reduced to a minimum over 1000 iterations in all cases. We

fixed the weights α and β to 1 and 10000 in proportion (α =1, β=10000) for the baseline experiment, respectively, then changed them to $1:10^3$, $1:10^5$, respectively, in order to analyze how sensitive the results are to content-style equilibrium.

The results were interpreted qualitatively using the three predetermined conditions and quantitatively compared by analyzing the final loss.

Impact of Layer Selection

The results of our experiments validate the basic NST idea: which layers to pick up determines the quality of artistic syntheses.

- *Color Palette Transfer:* Shallow style layers (e.g., conv1_1 in VGG19) resulted in a more immediate and localized color transfer, which often gave rise to a patchy or noisy look. The deeper style layers (e.g., the full five-layer sets) delivered better overall and more coherent color palette transfer, which also modeled the global color characteristics of the style image.
- *Style Technique Visibility:* Transfer of fine style procedures, such as Van Gogh's brush strokes, was most effective when a combination of shallow and mid-level style layers was employed. The shallow layers encoded local textures, and mid-level layers (e.g., conv3_1) organized this texture information into larger, more complex patterns.
- *Complexity of Detail Abstraction:* We often found that content layers that were too deep (e.g., conv5_1 in VGG19) resulted in the loss of smaller details in the image, thus achieving a highly abstract and generalized version of the cityscape. The middle level of content (conv4_2) always found the optimal compromise between maintaining perceptual structures from the content and injecting stylistic information.

Comparison of Models

A comparative analysis of the three models revealed distinct performance characteristics, which are summarized in Table 1.

Table 1: Comparative Performance of CNN Architectures for Neural Style Transfer

Features	VGG19	EfficientNet-B0	Custom CNN
Feature Hierarchy	Excellent separation of content/style	That is fine, but style features are	Moderate, less depth for fine separation
Color Transfer	High fidelity, especially with deep style layers	Excellent, due to strong channel-wise features	Patchy, less consistent
Detail Preservation	Best with mid-level content layers	High, even with deeper content layers	Low, prone to over-generalization
Computational Cost	High (Slowest)	Low (Fastest)	Moderate
Overall Artistic Quality	High (Benchmark)	High (Competitive)	Moderate

Specific Case Studies

- **Van Gogh Style Transfer:** VGG19 model setting conv4_2 for content and five layers for style provided the best visual outcome where the swirling, impasto texture transfers to cityscape without losing its structural outline. EfficientNet-B0 was faster and resulted in good color transfer, but the texture of the brushstrokes was less pronounced, indicating that its feature maps are not as sensitive to high-frequency textural details that contain the style.
- **Da Vinci Style Transfer:** With Da VInci's monochromatic, line-based style, the Custom CNN performed exceptionally well even with just the shallower style layer. It seems that for simple low-level features (lines, edges) that are the major contributors to simple styles (crisp highlighting), a simpler architecture may be adequate and even more computationally efficient.

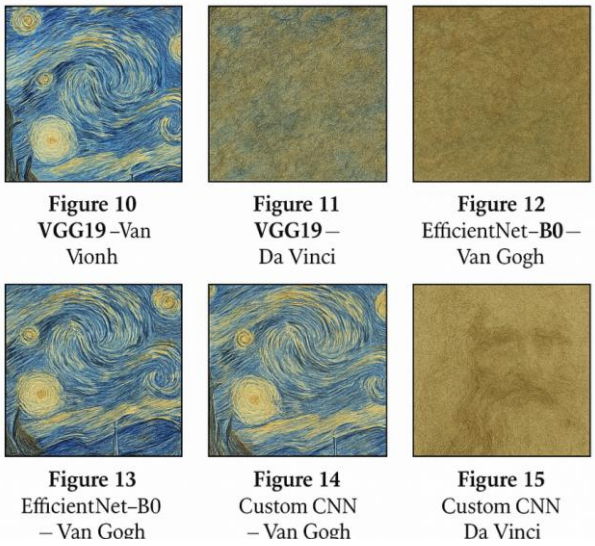

Figure 10 VGG19 –Van Vionh | Figure 11 VGG19 – Da Vinci | Figure 12 EfficientNet–B0 – Van Gogh

Figure 13 EfficientNet–B0 – Van Gogh | Figure 14 Custom CNN – Van Gogh | Figure 15 Custom CNN Da Vinci

Fig 10-15: Generated Images

7. CONCLUSION AND FUTURE WORK

This study proved that the choice of pre-trained CNN layer plays a critical role in the quality of artistic style synthesis in NST. By comparing VGG19, EfficientNet-B0 and a Custom CNN, we proved that the quality of the generated images is highly influenced by the complexity and nature of feature representations. Intermediate layers (i.e., VGG19 conv4_2) were repeatedly estimated to be optimal because of their preservation of spatial geometry and freedom from style variations. The results showed that there is no single layer that is the best representation; rather, we need to pool shallow and mid-level layers to encode local texture patterns, alongside the global structure of the style. Although VGG19 achieved the best perceptual quality in all experiments, EfficientNet-B0 showed competitive results with one-tenth of the computational effort. This makes EfficientNet-B0 competitive with state-of-the-art models when performance and resources are considered together. The more restrictive expressiveness of the Custom CNN also emphasized how architectural choices and hierarchical representation lead to style reproduction. The primary drawback of this study is that it uses a qualitative, perceptual-based evaluation criterion, which is appropriate for artistic material but is subject to subjective interpretation. We expect future work to include a quantitative evaluation, for example, using SSIM or some kind of structured user study to define more unbiased evaluation standards. Future work could focus on building real-time style transfer pipelines with feed-

forward networks, investigating different qualities and stabilities of loss functions to achieve more reliable stylized results beyond the Gram-Matrix-based method, and expanding our meta-learned strategy from VG in extending network architectures such as ResNets and Vision Transformers. They raise hope for a more profound understanding of how network hierarchies encode artistic styles and how this may be used to create faster style transfer systems that are more expressive and style-controllable.

8. REFERENCES

[1] Alexandru, I. C., Nicula, C., Prodan, C., Rotaru, R.-P., Voncila, M.-L., Tarba, N., & Boiang (2022). Image Style Transfer via Multi-Style Geometry Warping. Applied Sciences, 12(12), 6055. https://doi.org/10.3390/app12126055

[2] Artistic Arbitrary Style Transfer. (2022). https://doi.org/10.48550/arxiv.2212.11376

[3] Galerne B., Raad L., Lezama J., Morel J. (2022). Scaling Painting Style Transfer. arXiv.Org, abs/2212.13459. https://doi.org/10.48550/arXiv.2212.13459

[4] Gatys , L. A., Ecker , A. S., & Bethge , M. (2015). A Neural Algorithm of Artistic Style.arXiv preprint arXiv :1508.06576 .

[5] Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Neural Algorithm of Artistic Style. Journal of Vision, 16(12), 326.

[6] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data using neural networks. Science , 313(5786), 504–507.

[7] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution . European Conference on Computer Vision (ECCV).

[8] Kashyap , K. H., Garg , M., Fargose , S., & Nair, S. (2025). Dynamic Neural Style Transfer for Artistic Image Generation using VGG19. https://doi.org/10.48550/arxiv.2501.09420

[9] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning was applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

[10] McCulloch, W. S., & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 5(4), 115–133.

[11] Md Redoan Hosen, Md Borhan Hosen, Afzalur Rahaman , Yasin Arafat and Abhijit Pathak (2023); ENVIRONMENTAL POLLUTION ANALYSIS AND PREDICTION OF INFLUENTIAL FACTORS: A DATA-DRIVEN INVESTIGATION Int. J. of Adv. Res. 11 (Oct). [323-336] (ISSN 2320-5407).

[12] Miah , J., Cao, D. M., Sayed , M. A., & Haque , M. S. (2023). Generative AI Model for Artistic Style Transfer Using Convolutional Neural Networks. Journal of Computer Science and Technology Studies, 5(4), 78–85. https://doi.org/10.32996/jcsts.2023.5.4.9

[13] Mirza Maria Moon, Sadia Afrin, Abhijit Pathak . Classification of User Reviews on Online Travel Booking Applications in Bangladesh using Multinomial Naïve Bayes. International Journal of Computer Applications. 187, 37 (Sep 2025), 56-61. DOI=10.5120/ijca2025925646

[14] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 65(6), 386–408.

[15] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by backpropagating errors. Nature , 323(6088), 533–536.

[16] Ruta , D., Gilbert, A., Motiian , S., Faieta , B., Lin, Z. L., & Collomosse , J. (2022). HyperNST : Hyper-Networks for Neural Style Transfer. 201–217. https://doi.org/10.48550/arXiv.2208.04807

[17] Saiful Kabir, Sihabul Islam Safin, Marjahan Tanjin, Himu Akter, Rajib Ghose, Abhijit Pathak . Predicting Loan Repayment Reliability in Cooperative Societies using Naïve Bayes Classifier: A Data Mining Approach for Risk Mitigation and Decision Support. Int Journal C.omput. Appl . 186, 36 (Aug 2024), 16-23. DOI=10.5120/ijca2024923937

[18] Sethi, P., Bhardwaj, R., Sharma, N., Sharma, D. K., 38 , Srivastava, G. (2024). A deep learning-based neural style transfer optimization approach.Intelligent Data Analysis, 1–15. https://doi.org/10.3233/ida-230765

[19] Seyed , S. H., Cansever , A., & Hart, D. (2025). Improving Masked Style Transfer using Blended Partial Convolution. arXiv.Org, abs/2508.05769. https://doi.org/10.48550/arxiv.2508.05769

[20] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[21] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning (ICML).

[22] Werbos, P. J. (1974).Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University Cambridge.

[23] Xian-Fang Li, Han Cao, Zhaoyang Zhang, Jiacheng Hu, Yuhui Jin, Zihao Zhao -12 Nov 2024