Unlocking Artistry: Exploring the Depths of Neural Style Transfer with Pretrained Models

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of the organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Painting has remained a beloved and enduring art form for millennia, captivating the imagination of people through iconic masterpieces like van Gogh's "The Starry Night." Traditionally, recreating an image in a specific artistic style demanded the skilled hand of an artist and substantial time investment. However, since the mid-1990s, the alluring artistry that underlies such works has piqued not only the interest of artists but also that of computer science researchers [1].

In the digital age, this transformation finds its most promising realization in the technique known as neural style transfer. This innovative approach seamlessly marries the artistic essence of one image with the visual content of another, unleashing the potential for the creation of visually captivating and artistic representations that resonate with viewers on an emotional level.

The reach of neural style transfer extends beyond the confines of artistic creativity, permeating practical and commercial domains. In the realm of marketing and advertising, businesses have harnessed its power to modify product visuals, expertly crafting images that captivate and resonate with consumers. This application injects life into branding materials, infusing advertising campaigns with eye-catching, stylized visuals that leave indelible impressions in the minds of viewers. Furthermore, in the spheres of entertainment and multimedia, neural style transfer is revolutionizing the conversion of mundane photographs into captivating pieces of art, breathing new life into visual content and elevating the art of storytelling.

The ever-expanding landscape of applications underscores the urgent demand for style transfer methods that are both artistically sophisticated and efficiently practical, catering to a diverse array of creative and commercial requirements. As the field continues to mature, the horizon for innovation and impact broadens, offering transformative possibilities not only within the realm of art but also across various sectors. The future holds the promise of innovative and influential applications that will further bridge the worlds of art and technology.

II. Literature Review

The world of Neural Style Transfer (NST) is a captivating domain within computer vision that marries the content of one image with the style of another, producing visually compelling artistry. Our exploration into this field is guided by a series of research papers, each unveiling a unique facet of this creative process.

One of these papers introduces a novel perspective on NST by framing it as a domain adaptation problem. It emphasizes the importance of aligning feature distributions between style and generated images, particularly by matching Gram matrices, which can be seen as minimizing the Maximum Mean Discrepancy (MMD). This shift in perspective aligns NST with distribution alignment and sets the stage for future explorations [2].

Building on this foundational insight, another paper extends the application of NST from static images to dynamic videos. Drawing inspiration from a feed-forward CNN approach, this work introduces a novel real-time video style transfer

Identify applicable funding agency here. If none, delete this.

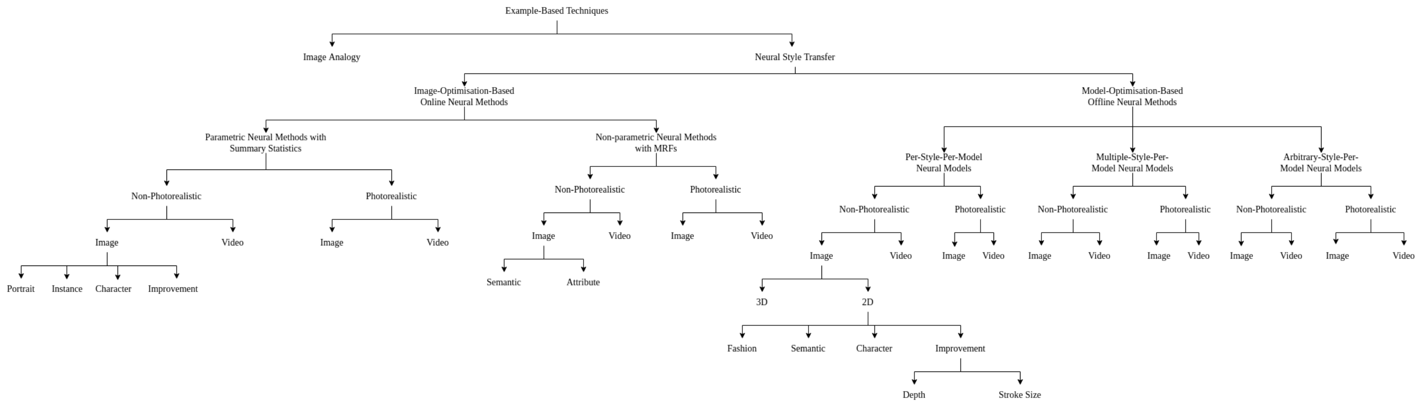

Fig. 1. A Taxonomy of NST Techniques

method, ensuring temporal consistency while preserving semantic content and stylistic integrity. Leveraging advances in perceptual loss, style loss, and instance normalization, this research broadens the horizons of NST, making it accessible for dynamic multimedia [3].

For those seeking greater control over the stylization process, yet another paper offers a comprehensive review of image stylization techniques. It fuses insights from patch-based methods, color style transfer, and procedural stylization, building upon the strengths of NST. This synthesis introduces intuitive control mechanisms for shaping the stylization outcome. The fusion of these techniques represents a testament to the evolving artistry that NST enables [4].

A critical review paper takes a step back to evaluate the broader landscape of NST. This comprehensive review scrutinizes the advantages and limitations of various architectural approaches, including the incorporation of Generative Adversarial Networks (GANs) into NST. It also underscores the challenges posed by real-time mobile applications and the research gaps in the NST domain. This paper serves as a reflective compass, guiding the direction of future research endeavors [5].

Lastly, an innovative paper embarks on a fascinating journey by employing meta networks for NST. By eschewing the need for training specific image transformation networks for each style, this work contributes to the efficiency and generality of NST. It draws inspiration from hypernetworks, style loss, and image transformation networks, while also overcoming the limitations of gradient descent optimization. The use of meta networks to generate textures is a commendable innovation, reducing computational burdens and enhancing flexibility. This approach showcases the growth potential of NST and its adaptability across various applications [6].

In summary, these research papers collectively unravel the multifaceted world of Neural Style Transfer, showcasing its evolution from a domain adaptation perspective, its application to real-time videos, its enhanced control mechanisms, its critical evaluation, and its efficient style transfer using meta networks. As we explore the depths of this artistic domain, we find a treasure trove of opportunities for further research, innovation, and creative expression.

III. Understanding Neural Style Transfer

A. *Introduction to Neural Style Transfer*

In the arena of image processing and artistic transformation, neural style transfer is a captivating technique that combines art and computational science. This innovative method uses deep convolutional neural networks (CNNs) to seamlessly merge the artistic style of one image with the content of another. What sets it apart from traditional image editing is its ability to intuitively learn and apply artistic styles, eliminating the need for explicit instructions.

This unique quality marks neural style transfer as a promising approach, where computational algorithms and artistic expression work together naturally. It blurs the lines between art and technology, fostering a creative synergy that redefines the boundaries of visual manipulation and artistic expression.

B. *Separation of Content and Style*

The basic idea of breaking down an image into its two constituent parts—content and style—is essential to the effectiveness of neural style transfer. Content is an acronym for the meaningful aspects of the image, which includes objects, shapes, and the story. Conversely, style encompasses the non-semantic, artistic elements that enhance visual attractiveness, like brushwork patterns, colour palettes, and texture.

This segment is fundamental to the neural style transfer mechanism, enabling the neural network to independently alter various components. This skill lays the foundation for a subtle interaction that results in an artwork that carefully maintains the core substance of the content while blending in the selected artistic style. In addition to promoting artistic expression, operational dualism highlights how flexible and adaptive neural style transfer is.

C. *Feature Extraction in Neural Networks*

Feature extraction in the neural network is a crucial step in the field of neural style transfer. In order to extract and extract important information from the input image, it entails

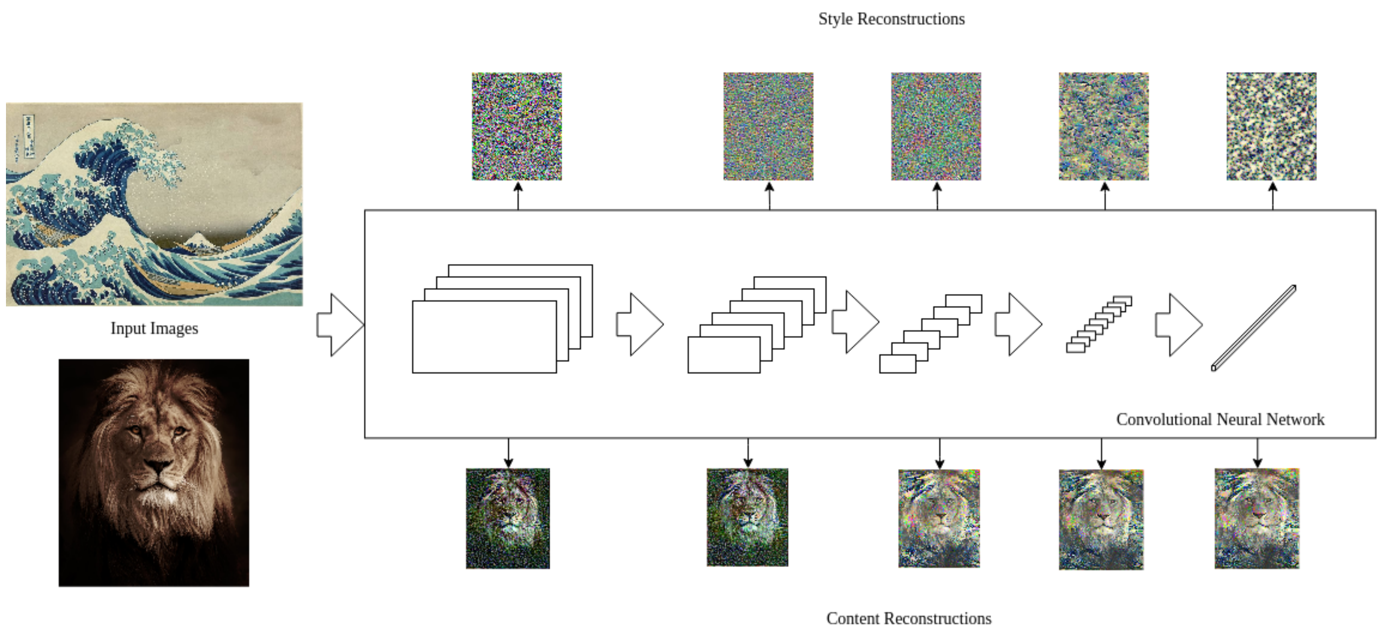

Fig. 2. Visualizing the Neural Style Transfer Pipeline - Input Images, Convolutional Layers, and Style-Content Reconstructions at Each Stage.

going through multiple layers. Making a subtle depiction that separates the text and style components is the aim.

The key to this process is figuring out which crucial layers to use for feature extraction. An intermediate layer is carefully selected for content, positioned to provide a balance between low-level and high-level elements. The essential qualities of the image, including the objects and their spatial connections, are well captured by this layer.

On the other hand, style extraction requires the use of several layers that are dispersed over the network architecture. These layers are grouped according to various phases of picture processing and concentrate on distinct stylistic elements. This multi-pronged strategy guarantees a thorough comprehension of the image's aesthetic components, such as texture, colour shifts, and complex patterns.

Notably, this feature extraction approach doesn't require redundancy to function. By providing unique information, each layer reduces needless duplication. This non-redundant method fosters a more nuanced representation of both content and style while minimising computing resources and improving the network's ability to distinguish fine nuances.

To sum up, neural network feature extraction for style transfer is a precisely calibrated orchestration. It picks its way through layers with accuracy, capturing only the necessary elements without redundancy. This tactical move improves the network's ability to identify and replicate style as well as content, which eventually helps to develop visually stunning and creatively expressive synthetic artworks.

D. Optimization and Loss Functions

In the optimization and loss functions phase of neural style transfer, the aim is to iteratively refine the generated image to seamlessly blend content and style. This process is guided by two critical components: the content loss and the style loss.

1) Content Loss: The generated image's features are contrasted with the content image in the content loss process. Its purpose is to ensure that the content of the created image remains unchanged while the algorithm modifies its style. In this manner, the style components are added to the content image while preserving its authenticity.

As loss networks, two instances of the identical pre-trained CNNs for image categorization are employed. A test image (created) and a reference image (content) will be supplied to these networks, respectively. The loss function receives the outputs from these two classifiers as inputs.

In plain English, the calculation is a distance (Euclidean) between two contents output from the loss CNNs; one content from the generated image and the other from the base image. [7]

Two calculations must be made before you can determine the content loss function. The content characteristics of the pre-trained loss networks-based content image as well as the generated image. The L2 Norm, also known as the Mean Squared Error, is then computed.

We follow the name in order to complete this calculation. First, use element-wise subtraction to calculate the error.

Deduct the generated image features from the image features of the material.

To obtain the squared errors, square these mistakes element-by-element. To find the average, add up all the values, divide by the total number of features, and you'll have the mean squared error.

$$L_{content} = \frac{\sum \left(\phi_j\left(y_{test}\right) - \phi_j\left(y_{ref}\right)\right)^2}{C_j H_j W_j}$$

The content loss makes sure that the final image in the higher layers includes activations that are comparable to those of the base image. While style and other losses address lower layers, content loss deals with the activations of upper layers. Content loss is crucial in order to produce an ideal and effective image.

2) Style Loss: Another important loss is the style loss which is represented by the gram matrix.Style loss is the measure of how different the lower-level features of the generated image are from the base image. For example, features like color and texture.Style loss is obtained from all the layers whereas content loss is obtained from higher layers. It goes into the deepest of layers to make sure that there is a visible difference between the style image and the generated image. After all, we don't want the original image to lose its value and real meaning.

The style loss is meant to penalize the output image when the style is deviating from the supplied style image. Now, for content loss, you can simply add up and divide for the Mean Squared Error value. For style loss, there is another step.First, a loss network is used like with the content loss. Both the test (generated) image features and style image features are fed to the loss networks. This produces their activations.

Next, these outputs are averaged over every value in the feature map to calculate the Gram matrix. This matrix is a measurement of the style at each layer. The matrix measures covariance and therefore it captures information about regions of the image which tend to activate together. The benefit of using a gram matrix is that it enables different features to co-exist in different parts of the images.

$$G_j(x) = \frac{\sum_{h=0}^{H_{j-1}} \sum_{w=0}^{W_{j-1}} \phi_j(y_{test})\phi_j(y_{test})}{C_j H_j W_j}$$

After the gram matrix, the style loss is a squared distance (i.e. error) between the gram matrix of the generated image and the gram matrix of the style image. In this case, the distance is called the Frobenius norm because it is measuring the distance between two matrices. It is an extension of the Euclidean norm for matrices.

$$||X||_F = \sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |a_{i,j}|^2}$$

These additional steps make the style loss more complex and its calculation and implementation more effort. There are also approximation methods for Gram Matrices to speed up their calculation. [8]

3) Gram Matrix: Gram matrix measures how much correlation there is between all the style feature maps. It calculates the features of every layer of a style.

The Gram matrix is used to determine the calculation for all the particular styles for both the generated image and the style image.

4) Total Loss: Total loss is simply the sum of the content loss and style loss. By finding out the total loss, you can understand how the optimizer is finding an image that has the style of one image and the content of another.

$$Total_{Loss} = Style_{Loss} + Content_{Loss}$$

By controlling the total loss with the user-defined content weight and style weight, we can either have a more artistic image or an image with more features but less style or art.

IV. Exploring Pre-trained Models

This section presents an in-depth exploration of pre-trained neural network models—VGG19, VGG16, ResNet50, ResNet105, and MobileNet—acknowledged for their excellence in visual recognition tasks. Each model, distinguished by its unique architecture, plays a pivotal role in image classification and extends its utility to broader applications. Utilizing pre-training on the ImageNet dataset as a foundational knowledge base, these models exhibit enhanced capabilities for recognizing and classifying objects within our specific research domain. The streamlined structure of VGG19, the depth of representation in VGG16, the residual learning innovation in ResNet50, the increased depth in ResNet105, and the efficiency of MobileNet collectively contribute to their widespread adoption in computer vision. This section provides a comprehensive overview, elucidating the architectures, strengths, and applications of each model, setting the stage for subsequent analyses and experiments within our research framework.

A. VGG19

VGG19, short for Visual Geometry Group 19, is a deep convolutional neural network architecture that gained prominence for its effectiveness in image classification tasks. Developed by the Visual Geometry Group at the University of Oxford, VGG19 is characterized by its deep structure, consisting of 19 layers, with 16 convolutional layers and 3 fully connected layers. The network's architecture follows a straightforward design philosophy, employing small 3x3 convolutional filters throughout the layers, which allows the model to learn complex hierarchical features. The use of multiple layers with small filters enables VGG19 to capture intricate patterns and details in input images, making it highly effective in recognizing and classifying objects across various visual domains.

Each convolutional layer in VGG19 is followed by a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity to the network, enabling it to learn and represent complex relationships within the data. The architecture's simplicity and uniformity contribute to its ease of understanding

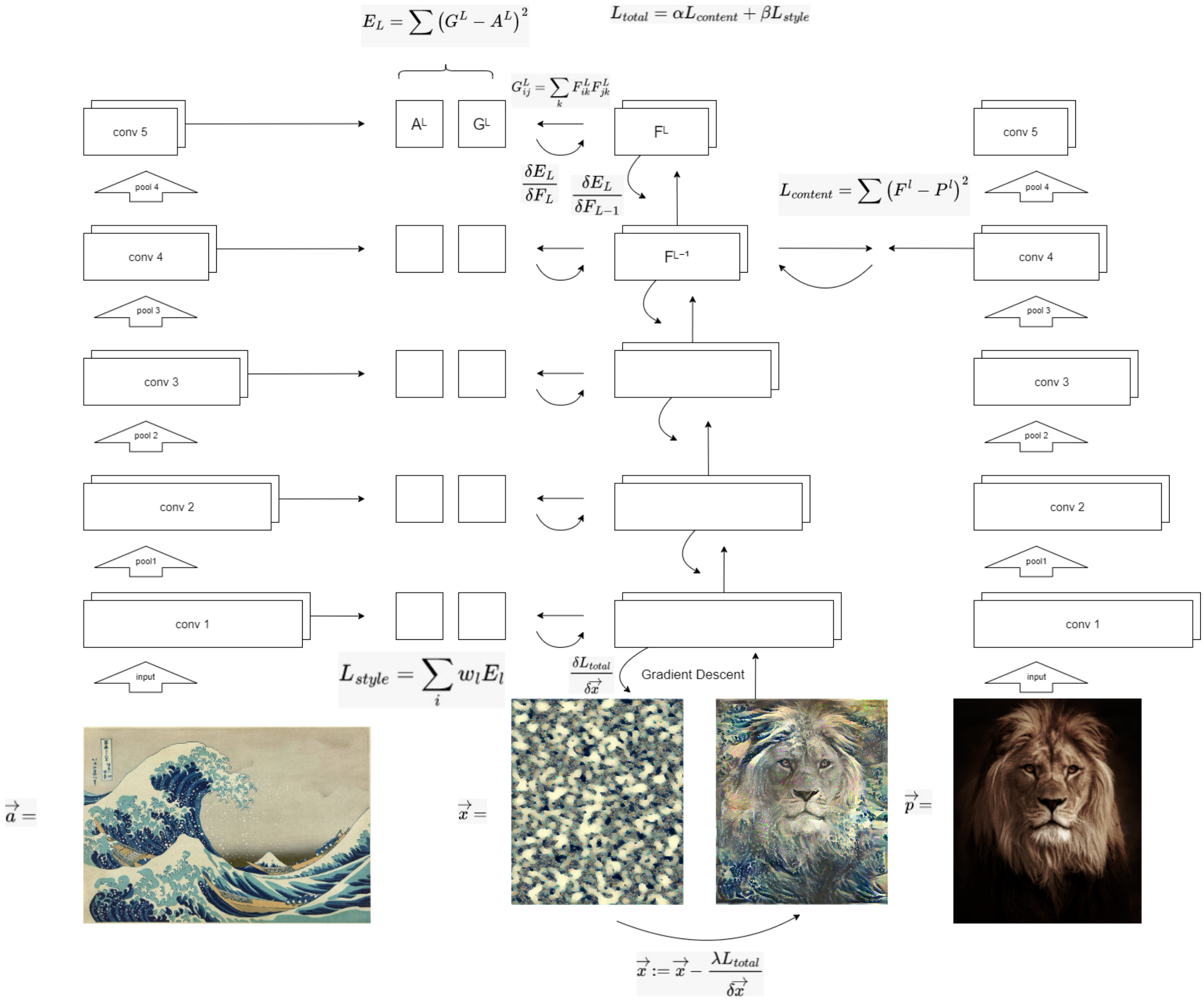

Fig. 3. Enter Caption

and implementation. VGG19's exceptional performance on image classification tasks, demonstrated in competitions such as the ImageNet Large Scale Visual Recognition Challenge, has made it a popular choice for researchers and practitioners working on computer vision applications. The ability of VGG19 to extract hierarchical features and its straightforward architecture have paved the way for its adoption and adaptation in various domains beyond image classification, such as object detection and image generation.

In this research, we also utilized a pre-trained model based on the VGG19 architecture, leveraging a dataset that played a pivotal role in its initial training. The dataset employed for pre-training VGG19 is the ImageNet Large Scale Visual Recognition Challenge dataset. ImageNet, a widely recognized and comprehensive dataset, encompasses over a million high-resolution images spanning a diverse range of object categories. The ImageNet dataset has been a cornerstone in the development and evaluation of deep learning models, serving as a benchmark for image classification tasks.

The ImageNet dataset is particularly valuable for training deep neural networks like VGG19 due to its extensive coverage of object classes, ranging from animals and everyday objects to complex scenes. The large-scale nature of ImageNet facilitates the learning of intricate features and patterns by the model, enabling it to generalize well to a broad spectrum of visual recognition tasks. Leveraging a pre-trained VGG19 model on ImageNet provides our research with the advantage of transfer learning, where the knowledge gained from the vast and diverse ImageNet dataset is applied to our specific domain of interest. This approach allows the model to capitalize on the wealth of visual information present in ImageNet, enhancing its ability to recognize and classify objects within our target dataset more effectively.

B. VGG16

GG16, an abbreviation for Visual Geometry Group 16, is a deep convolutional neural network architecture renowned for its success in image classification tasks. Developed by the Visual Geometry Group at the University of Oxford, VGG16 is composed of 16 layers, featuring 13 convolutional layers and 3 fully connected layers. The hallmark of VGG16 is its uniform architecture, employing small 3x3 convolutional filters throughout the network. This consistent use of filters allows the model to capture and learn intricate hierarchical features in input images, facilitating the recognition of complex patterns. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function, introducing non-linearity to the model and enabling it to better capture the underlying relationships within the data.

VGG16's simplicity and effectiveness stem from its deep architecture, as it can represent both low-level and high-level features in an image. The network's deep structure enables it to discern fine details and abstract features, making it particularly well-suited for image classification tasks. The straightforward design of VGG16 has contributed to its popularity and widespread adoption in the computer vision community. Its success in the ImageNet Large Scale Visual Recognition Challenge and subsequent competitions has established VGG16 as a benchmark architecture, serving as a foundation for subsequent deep learning models and inspiring research in various computer vision applications beyond image classification, such as object detection and segmentation.

In our research, we employed a pre-trained VGG16 model, which was originally trained on the ImageNet Large Scale Visual Recognition Challenge dataset. The ImageNet dataset is a comprehensive and diverse collection consisting of over a million labeled images spanning thousands of object categories. This dataset has been instrumental in advancing the field of computer vision and deep learning, serving as a benchmark for image classification tasks.

The ImageNet dataset is well-suited for training models like VGG16 due to its extensive coverage of object classes, ranging from animals and everyday objects to complex scenes. The large-scale nature of ImageNet allows models to learn rich hierarchical features and intricate patterns, contributing to their ability to generalize well to various visual recognition tasks. By leveraging a pre-trained VGG16 on ImageNet, our research benefits from transfer learning, a technique that harnesses the knowledge gained from a broad dataset to improve performance on a more specific task or domain. The pre-training on ImageNet provides the VGG16 model with a foundational understanding of visual features, enhancing its capacity to recognize and classify objects within our dataset more effectively.

C. Resnet50

ResNet50, short for Residual Network with 50 layers, represents a pivotal advancement in deep learning architectures, specifically designed to address the challenges of training very deep neural networks. Developed by Microsoft Research, ResNet50 is a variant of the ResNet architecture, featuring 50 layers, and is celebrated for its innovative use of residual learning. The key insight behind ResNet50 is the introduction of residual blocks, where the output of a layer is combined with the input through a shortcut connection. This residual or skip connection allows the network to learn and emphasize the residual information, making it easier to train deep networks by mitigating the vanishing gradient problem. The residual connections enable the model to maintain a more direct and efficient flow of information during training, facilitating the training of extremely deep networks without suffering from degradation in performance.

The architecture of ResNet50 is characterized by its modular structure, with the repeated stacking of residual blocks. Each block consists of multiple convolutional layers followed by batch normalization and ReLU activation functions. The skip connections ensure the preservation of information across layers, aiding in the efficient propagation of gradients during backpropagation. ResNet50 has demonstrated exceptional performance in various computer vision tasks, winning the ImageNet Large Scale Visual Recognition Challenge in 2015. Its success has led to widespread adoption in both research and practical applications, serving as a foundation for subsequent deep learning models and influencing advancements in image classification, object detection, and other computer vision domains.

In this research endeavor, we harnessed the power of a pre-trained ResNet50 model, which had its origins in the ImageNet Large Scale Visual Recognition Challenge dataset. The ImageNet dataset is a seminal collection comprising millions of labeled images spanning an extensive array of object categories. Renowned for its diversity and scale, ImageNet has emerged as a crucial benchmark in the field of computer vision and deep learning, providing an ideal foundation for training sophisticated models like ResNet50.

The ImageNet dataset's richness lies in its inclusion of various object classes, ranging from common everyday items to complex and nuanced subjects. This expansive dataset empowers models to acquire intricate hierarchical features and learn nuanced patterns, enhancing their ability to generalize across a wide spectrum of visual recognition tasks. By leveraging a pre-trained ResNet50 on ImageNet, our research capitalizes on the benefits of transfer learning. This approach allows the model to transfer the knowledge acquired during its initial training on ImageNet to our specific domain, ultimately bolstering its proficiency in recognizing and classifying objects within our dataset. The pre-training on ImageNet acts as a knowledge reservoir, endowing the ResNet50 model with a robust foundation for more effectively addressing the intricacies of our targeted visual recognition task.

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [**?**].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 4", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Fig. 4. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [**?**].

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.

REFERENCES

[1] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3365–3385, 2020.

[2] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *CoRR*, abs/1701.01036, 2017.

[3] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 783–791, 2017.

[4] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3985–3993, 2017.

[5] Akhil Singh, Vaibhav Jaiswal, Gaurav Joshi, Adith Sanjeeve, Shilpa Gite, and Ketan Kotecha. Neural style transfer: A critical review. *IEEE Access*, 9:131583–131613, 2021.

[6] Falong Shen, Shuicheng Yan, and Gang Zeng. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8061–8069, 2018.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[8] Petros Drineas, Michael W Mahoney, and Nello Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(12), 2005.