International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI: 10.55041/ISJEM06138
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

A Multi-Style Image and Video Cartoonization Framework using Deep Learning and Adaptive Style Filters

1st Muriki Parvathi
Computer Science and Engineering
Rajiv Gandhi University of Knowledge Technologies
Basar, India murikiparvathi05@gmail.com

2nd Dheekonda Rishitha
Computer Science and Engineering
Rajive Gandhi University of Knowledge Technologies
Basar, India rishithadheekonda3@gmail.com

3rd Paka Akshaya
Computer Science and Engineering
Rajive Gandhi University of Knowledge Technologies
Basar, India akshayapaka2003@gmail.com

4th Assistant Professor Ms.Lingavva
Computer Science and Engineering
Rajive Gandhi University of Knowledge Technologies
Basar, India bhanu.lingava@gmail.com

***Abstract*—**Cartoonization of images and videos has gained sig-nificant attention due to its applications in entertainment, social media, and digital content creation. Most existing approaches fo-cus on generating a single artistic style, limiting their adaptability. In this paper, we propose a multi-style cartoonization framework that combines deep learning-based cartoonization with adaptive style filters to produce diverse cartoon effects.

The system uses a white-box deep learning model for base cartoon generation and enhances the output with multiple styles such as anime, comic, and sketch. The framework supports both image and video inputs and is implemented as a real-time web-based application. Experimental results show that the proposed system generates visually appealing outputs while maintaining efficiency.

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

I. Introduction

A. Background and Motivation

Cartoonization is a process of transforming real-world im-ages and videos into stylized cartoon-like representations. It combines techniques from computer vision, image processing, and deep learning to simplify textures, enhance edges, and apply artistic effects. Traditional cartoonization methods pri-marily rely on edge detection, color quantization, and filtering techniques such as bilateral filtering to produce cartoon effects. While these approaches are computationally efficient, they often lack flexibility and fail to generate high-quality artistic results.

With the advancement of deep learning, especially convo-lutional neural networks (CNNs) and generative models, more sophisticated cartoonization methods have been developed. Models such as White-Box Cartoonization provide improved visual quality by separating image structure and texture com-ponents. However, most existing approaches are limited to generating a single cartoon style, restricting their usability in diverse real-world applications.

In addition, many existing systems focus only on static images and do not efficiently support video cartoonization due to challenges such as temporal consistency and computational complexity.

B. Challenges in Multi-Style Cartoonization

Developing a multi-style cartoonization system for images and videos involves several technical and practical challenges. One of the primary challenges is maintaining high visual quality while transforming real-world images into cartoon representations. The system must preserve important features such as edges, textures, and colors while simplifying details to achieve a cartoon-like appearance. However, excessive sim-plification can lead to loss of important information, whereas insufficient processing may fail to produce the desired artistic effect.

Another major challenge is supporting multiple cartoon styles within a single framework. Different styles such as anime, comic, and sketch require distinct transformations and filtering techniques. Ensuring that each style produces visually consistent and meaningful outputs without degrading image quality is difficult. Additionally, balancing style diversity and computational efficiency adds further complexity to the system design.

Video cartoonization introduces additional challenges com-pared to image processing. Handling large volumes of data, maintaining temporal consistency between frames, and ensur-ing smooth transitions are critical for generating high-quality outputs. Moreover, synchronizing audio with the processed

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

video and managing file sizes requires careful implementation to avoid performance bottlenecks.

Achieving real-time or near real-time processing is another significant challenge, especially when deep learning models are involved. These models often require substantial computational resources, making it difficult to deploy them efficiently on systems without high-end GPUs. Optimizing performance while maintaining output quality is therefore essential.

Furthermore, integrating deep learning-based cartoonization with traditional image processing techniques requires careful coordination. Improper integration may lead to artifacts or inconsistencies in the final output. In addition, variations in input images, such as lighting conditions, resolution, and background complexity, can affect the performance of the system, making robustness an important consideration.

C. Related Work and Existing Approaches

Cartoonization has been extensively studied in the field of computer vision and image processing, with approaches evolving from traditional filtering techniques to advanced deep learning models. Early methods primarily relied on image processing operations such as edge detection, color quantization, and smoothing filters. Techniques like bilateral filtering and adaptive thresholding were widely used to simplify textures while preserving important edges. Although these methods are computationally efficient, they often produce limited artistic quality and lack flexibility in generating diverse styles.

With the advancement of deep learning, more sophisticated approaches have emerged for image cartoonization. Convolutional Neural Networks (CNNs) are widely used to learn complex mappings from real images to cartoon-like representations. Methods such as neural style transfer transform images into artistic styles by optimizing feature representations extracted from pre-trained networks. While these approaches generate visually appealing results, they are computationally expensive and may not be suitable for real-time applications. Generative Adversarial Networks (GANs) have further enhanced cartoonization by learning the distribution of cartoon images and generating more realistic outputs. Models such as CartoonGAN and White-Box Cartoonization demonstrate significant improvements in preserving structural details and producing high-quality stylized images. In particular, White-Box Cartoonization introduces an interpretable framework that separates image components, such as edges and textures, enabling better control over the cartoonization process. However, most of these models are designed for a single style and lack support for multiple artistic variations within a unified system.

In addition to image-based methods, video cartoonization has also gained attention. Traditional approaches process videos frame by frame, often resulting in temporal inconsistencies such as flickering. Recent methods attempt to address this issue by incorporating temporal coherence constraints, but they increase computational complexity and are challenging to deploy in real-time systems.

Despite these advancements, existing approaches typically focus on either high-quality single-style cartoonization or computational efficiency, but rarely both. Moreover, there is limited work on integrating multi-style capabilities with both image and video processing in a unified framework. This highlights the need for a system that combines deep learning-based cartoonization with flexible style transformations, which forms the focus of the proposed work.

D. Proposed Work and Contributions

In this work, we propose a multi-style image and video cartoonization system that converts real-world images and videos into visually appealing cartoon representations. The system is designed as a web-based application using a deep learning model combined with image processing techniques.

nitially, the input image or video frame is processed using a pre-trained cartoonization model (White-box Cartoonizer), which generates a base cartoon output. On top of this, addi-tional style transformation modules are applied to enhance the output into different artistic styles such as anime, comic, and sketch.

To improve user flexibility and control, an intensity adjust-ment mechanism is introduced. This allows users to control the strength of the cartoon effect, making the system more interactive and customizable.

For video processing, the system applies frame-by-frame cartoonization. To reduce computational time and improve performance, a frame-skipping optimization technique is used, which processes only selected frames while maintaining visual continuity.

The key contributions of this work are as follows:

1) Developed a multi-style cartoonization system support-ing anime, comic, and sketch styles.
2) Introduced user-controlled intensity adjustment for customizable output.
3) Extended cartoonization from images to videos using frame-by-frame processing.
4) Improved performance using frame-skipping optimiza-tion.
5) Combined deep learning with image processing tech-niques for better results.
6) Built a web-based application for easy user interaction.

E. Organization of the Paper

The rest of this paper is organized as follows. Section II presents the related work and existing approaches in the field of image and video cartoonization. Section III describes the proposed methodology, including the model architecture, multi-style cartoonization, and intensity control mechanism. Section IV discusses the implementation details and exper-imental results. Section V presents the performance analysis and comparisons. Finally, Section VI concludes the paper with future work directions

II. DATASET AND ANNOTATION

A. Dataset Collection

In this project, no specific labeled dataset was used for training, as the system utilizes a pre-trained deep learning

model known as the White-box Cartoonizer. The model was originally trained on large-scale datasets consisting of real-world images and their corresponding cartoon-style representations.

For testing and evaluation, a diverse set of input images and videos were collected from publicly available sources and personal media. These inputs include human faces, natural scenes, objects, and social media images to evaluate the generalization capability of the system.

For video cartoonization, input videos are processed by extracting frames using video processing techniques. Each frame is then treated as an individual image and passed through the cartoonization pipeline.

B. Annotation Process

Since the task focuses on image and video transformation rather than classification, explicit annotations or labeled data are not required. The system does not depend on ground-truth labels, as it performs style transformation directly using a pre-trained model.

Instead of annotation, the evaluation is performed qualita-tively based on visual characteristics such as:

- Edge preservation
- Color smoothness
- Texture simplification
- Cartoon-like appearance

C. Dataset Characteristics

The dataset used for testing exhibits the following charac-teristics:

1) **Diverse Input Data:**
The input includes various categories such as human faces, landscapes, objects, and real-world scenes, ensur-ing robustness across different domains.
2) **Unlabeled Data:**
The system operates on unlabeled data, as the task involves transformation rather than prediction or clas-sification.
3) **Image and Video Inputs:**
Both images and videos are used as inputs. Videos are processed as sequences of frames for cartoonization.
4) **Resolution Variability:**
Input data consists of different resolutions, which are resized during preprocessing to maintain consistency.
5) **Real-World Variations:**
The dataset includes variations in lighting conditions, backgrounds, and noise levels, reflecting real-world sce-narios.

D. Dataset Visualization and Processing

For this study, sample input images and videos were used to visualize the effectiveness of the proposed system. The cartoonization results are generated in multiple styles such as anime, comic, and sketch.

For video processing, each video is divided into frames, and cartoonization is applied frame-by-frame. The processed frames are then combined to reconstruct the final cartoonized video while maintaining the original frame rate.

Fig. 1. Sample Cartoonization Results in Different Styles

The performance is evaluated based on visual quality and processing time, demonstrating the effectiveness of the pro-posed multi-style cartoonization system.

III. Methodologies and Models Training

A. Overall Framework

The primary objective of this project is to develop an automated system for image and video cartoonization using deep learning and image processing techniques. The proposed system converts real-world images and videos into cartoon-style representations with multiple artistic styles.

The methodology consists of the following stages:

1) Input image or video upload
2) Image preprocessing and normalization
3) Base cartoonization using a pre-trained model
4) Multi-style transformation (anime, comic, sketch)
5) Intensity adjustment for user control
6) Video frame extraction and processing
7) Reconstruction of final output

B. Data Preprocessing

Preprocessing is an important step to ensure compatibility with the model and improve output quality. Since the system handles real-world images and videos, preprocessing is applied before cartoonization.

- Conversion of input images to RGB format
- Handling of RGBA images by removing transparency
- Resizing images to suitable dimensions
- Noise reduction using smoothing filters
- Frame extraction for video inputs

For video inputs, each frame is processed individually after extraction. This ensures consistent cartoonization across the entire video.

C. Deep Learning Model

The core of the proposed system is a pre-trained deep learning model known as the White-box Cartoonizer. This model is designed to transform real-world images into cartoon-like representations.

- The model uses convolutional neural networks (CNNs) to extract features
- It simplifies textures while preserving important edges
- It smooths colors to create flat cartoon regions

The model generates a base cartoon output, which is further enhanced using style transformations.

D. Multi-Style Cartoonization

To enhance visual diversity, the system introduces multiple artistic styles:

- **Anime Style:** Smooth color regions using bilateral filter-ing
- **Comic Style:** Edge detection combined with color filter-ing
- **Sketch Style:** Pencil sketch effect using grayscale trans-formation

These styles are implemented using image processing techniques applied on top of the base cartoonized output.
The overall working of the system is illustrated in Fig. 2.

E. Intensity Control Mechanism

An intensity control feature is introduced to allow users to adjust the strength of the cartoon effect.

- Pixel values are scaled using intensity factor
- Higher intensity produces stronger cartoon effect
- Lower intensity preserves more original details This improves user interaction and customization.

F. Video Processing

For video cartoonization, the system processes videos frame-by-frame:

- Frames are extracted using OpenCV
- Each frame is passed through the cartoonization pipeline
- Processed frames are recombined into a video

To improve efficiency, a frame-skipping technique is applied.

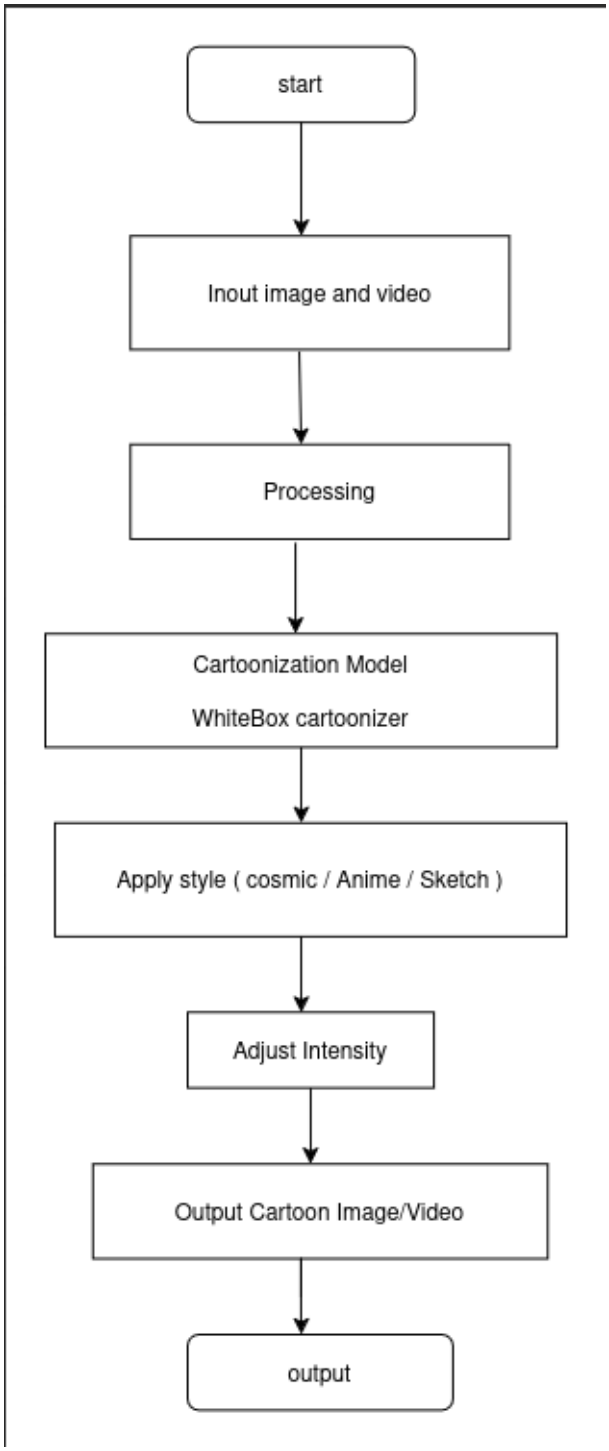

Fig. 2. Flow Diagram of Proposed System

G. Optimization Technique

To reduce computational cost, a frame-skipping approach is used:

- Only selected frames are processed
- Remaining frames are skipped to reduce time
- Maintains acceptable visual continuity

H. Mathematical Representation

Let the input image be represented as I and the cartoonized output as C.
The transformation can be defined as:

$$C = f(I, \theta)$$

where f represents the cartoonization model and θ denotes the learned parameters.
For intensity control:

$$C' = \alpha \cdot C$$

where α represents the intensity factor.

I. Algorithm

Step 1: Read input image/video
Step 2: Preprocess input
Step 3: Apply cartoonization model
Step 4: Apply selected style (anime/comic/sketch)
Step 5: Adjust intensity
Step 6: Save output

J.System Architecture

The system consists of three main components:

- Input Module
- Processing Module
- Output Module

K. Computational Complexity

The processing time depends on image resolution and number of frames. Video processing is computationally expensive as each frame is processed individually. Frame-skipping reduces the time complexity significantly.

L. Drawbacks of Existing System

The existing image and video cartoonization systems have several limitations that affect their performance, flexibility, and usability.

- **Limited Style Diversity:** Most existing systems focus on a single cartoon style, which restricts the variety of outputs. Users cannot switch between different artistic styles such as anime, comic, or sketch.
- **Low-Quality Output using Basic Image Processing:** Traditional approaches rely mainly on edge detection and filtering techniques. These methods often fail to preserve important features like edges and textures, resulting in less realistic cartoon effects.
- **Lack of User Control:** Existing systems do not provide options to control the intensity or strength of the cartoon effect. This leads to fixed outputs without customization.
- **High Computational Cost for Video Processing:** Video cartoonization requires processing each frame individ-ually, which increases computational time significantly, especially when running on CPU without GPU support.
- **No Optimization Techniques:** Most systems process ev-ery frame of a video, leading to unnecessary computation. Techniques like frame skipping are not commonly used to improve efficiency.
- **Limited Real-Time and Web Integration:** Many exist-ing approaches are not designed for real-time applications or web-based platforms, making them less accessible for general users.

M. Advantages of Proposed System

The proposed system improves existing cartoonization ap-proaches by integrating deep learning techniques with multi-style transformation and user control features.

- **Multi-Style Cartoonization:** The system supports mul-tiple cartoon styles such as anime, comic, and sketch, allowing users to generate diverse artistic outputs from a single input image or video.
- **High-Quality Output using Deep Learning:** The use of a deep learning-based model (White-box Cartoonizer) enables better edge preservation, texture smoothing, and visually appealing results compared to traditional image processing methods.
- **User-Controlled Intensity:** The system provides an in-tensity adjustment feature, allowing users to control the strength of the cartoon effect based on their preference.
- **Support for Both Images and Videos:** Unlike many existing systems, the proposed solution works for both images and videos, making it more versatile and practical.
- **Optimized Video Processing:** Frame-skipping technique is used to reduce computational cost during video pro-cessing, improving speed without significantly affecting visual quality.
- **Web-Based Implementation:** The system is imple-mented using a Flask-based web application, making it easily accessible and user-friendly.
- **Scalability and Real-Time Potential:** The system can be extended with GPU acceleration and optimized models for faster processing, making it suitable for real-time applications.

IV. Evaluation Metrics

To assess the effectiveness of the proposed cartoonization system, both quantitative and qualitative evaluation metrics are used. These metrics help measure image quality, processing efficiency, and visual performance.

A. Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is used to measure the difference between the original image and the cartoonized image. It evaluates how much noise or distortion is introduced during processing. A higher PSNR value indicates better image quality and closer similarity to the original input.

B. Structural Similarity Index (SSIM)

Structural Similarity Index (SSIM) measures the visual similarity between the original and cartoonized images. It considers three important factors: luminance, contrast, and structural information. SSIM provides a better representation of human visual perception compared to traditional error-based metrics.

C. Processing Time

Processing time refers to the total time required to convert an input image or video into its cartoonized form. This metric is used to evaluate the computational efficiency of the system. For videos, processing time increases with higher resolution and longer duration.

D. Frames Per Second (FPS)

Frames Per Second (FPS) is used to measure the speed of video processing. It indicates how many frames are processed per second. A higher FPS value represents faster processing and better suitability for real-time applications.

E. User Evaluation

In addition to quantitative metrics, qualitative evaluation is also performed. The output images and videos are visu-ally inspected based on criteria such as edge preservation, smoothness, color consistency, and overall artistic appearance. This helps assess how visually appealing and realistic the cartoonized outputs are.

V. Future Scope

A. Following Future Scope can take this project to a better level

1) ***Integration of GAN-based Models:****:* Future work can include integrating advanced Generative Adversarial Network (GAN) models such as AnimeGAN to produce more realistic and high-quality cartoon effects. These models can signifi-cantly enhance artistic output compared to traditional methods.

2) ***Real-Time Cartoonization:****:* The system can be ex-tended to support real-time cartoonization using webcam input. With GPU acceleration and optimized models, users can experience instant cartoon effects for live video streams.

3) ***Enhanced Style Customization:****:* Currently, the system supports limited styles such as anime, comic, and sketch. Future improvements can include adding more styles and allowing users to customize parameters like color intensity, edge thickness, and texture details.

4) ***Improved Video Processing Efficiency:****:* Video pro-cessing can be further optimized using parallel processing, GPU-based acceleration, and advanced frame interpolation techniques to improve speed and maintain quality.

5) ***Mobile and Cloud Deployment:****:* The system can be deployed as a mobile application or cloud-based service, enabling users to access cartoonization features from anywhere with minimal computational resources.

6) ***High-Resolution Output Support:****:* Future work can focus on generating high-resolution outputs (HD/4K) while maintaining quality and performance, making the system suit-able for professional applications.

7) ***AI-Based Style Transfer Learning:****:* The model can be extended to learn new styles automatically from datasets using deep learning techniques, enabling dynamic and adaptive cartoon generation.

The future scope of this project includes integrating ad-vanced deep learning models such as GAN-based architectures to improve visual quality. The system can be enhanced to support real-time webcam cartoonization and extended with more customizable styles. Additionally, improvements in video processing efficiency, mobile and cloud deployment, and high-resolution output generation will make the system more scalable and practical. Incorporating adaptive style learning techniques will further enhance the flexibility and intelligence of the system.

VI. Conclusion

This project presents a comprehensive approach for image and video cartoonization using deep learning and image pro-cessing techniques. The system is built using the White-box Cartoonizer model, which effectively transforms real-world images into cartoon-style representations while preserving essential features such as edges, textures, and structural de-tails. The experimental results demonstrate that deep learning-based approaches produce significantly better visual quality compared to traditional image processing methods.

The proposed system enhances flexibility by incorporating multiple cartoon styles, including anime, comic, and sketch, allowing users to generate diverse artistic outputs. Addition-ally, an intensity control mechanism is introduced to provide user-level customization of the cartoon effect. The system supports both image and video inputs, making it suitable for practical applications. For video processing, optimization techniques such as frame skipping are applied to reduce computational cost and improve efficiency, especially in CPU-based environments.

Overall, this work provides an efficient, scalable, and user-friendly cartoonization system that can be applied in areas such as entertainment, social media, and digital content creation. The project also lays a foundation for future enhancements, including real-time processing, GAN-based style generation, and high-resolution output support, making it a strong candidate for further research and development.

REFERENCES

[1] Y. Wang et al., "Learning to Cartoonize Using White-box Cartoon Representations," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[2] Y. Chen et al., "AnimeGAN: A Novel Lightweight GAN for Photo Animation," arXiv preprint arXiv:1908.04338, 2019.
[3] Y. Chen et al., "AnimeGANv2: A Better and Faster Version of AnimeGAN," arXiv preprint arXiv:2005.14297, 2020.
[4] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," CVPR, 2016.
[5] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
[6] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," ICCV, 1998.
[7] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.
[8] A. Ronacher, "Flask Web Development Framework," Available: https://flask.palletsprojects.com/
[9] FFmpeg Developers, "FFmpeg: A Complete Solution to Record, Convert and Stream Audio and Video," Available: https://ffmpeg.org/
[10] Z. Wang et al., "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, 2004.
[11] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs SSIM," International Conference on Pattern Recognition (ICPR), 2010.
[12] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.
[13] I. Goodfellow et al., "Generative Adversarial Nets," NeurIPS, 2014.
[14] A. Bovik, "Handbook of Image and Video Processing," Academic Press, 2005.
[15] Y. Cao et al., "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization," CVPR, 2018.
[16] Z. Yi et al., "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation," ICCV, 2017.
[17] J. Y. Zhu et al., "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," ICCV, 2017.
[18] J. Kim et al., "U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization," ICLR, 2020.
[19] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," ECCV, 2016.
[20] T. Karras et al., "Analyzing and Improving the Image Quality of StyleGAN," CVPR, 2020.
[21] T. Karras et al., "A Style-Based Generator Architecture for Generative Adversarial Networks," IEEE TPAMI, 2020.
[22] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," ICLR, 2021.
[23] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," ICCV, 2021.
[24] M. Ghiasi et al., "Exploring the Structure of a Real-Time, Arbitrary Neural Artistic Stylization Network," BMVC, 2017.
[25] Z. Li et al., "MobileGAN: Efficient GAN for Mobile Devices," IEEE Access, 2022.
[26] H. Wu et al., "Video Cartoonization with Temporal Consistency Using Deep Neural Networks," IEEE Transactions on Multimedia, 2022.
[27] T. Tewari et al., "Advances in Neural Rendering," Computer Graphics Forum, 2020.
[28] J. Ho et al., "Denoising Diffusion Probabilistic Models," NeurIPS, 2020.
[29] R. Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models," CVPR, 2022.