International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 3, pp.42-49, March 2026.
www.ijersem.com eISSN- 3107-9075

Cartoonify Image Using Machine Learning

[1]J Vaishnavi, [2]Nagari Daisy, [3]Balasubramanyam Hemanth Kumar, [4]R Arif, [5]Vecham Deepika

Department of CSE, Siddharth Institute of Engineering & Technology, Puttur, India.

[1]vaishnavi.spmvv@gmail.com, [2]nagaridaisy2005@gmail.com, [3]balasubramanyam884@gmail.com, [4]arifvijay6@gmail.com, [5]vechamdeepika@gmail.com

Abstract: A rise in interest in the ability to quickly generate digital content through the use of automated systems has created a strong need for efficient methods to create cartoony images. A hybrid machine learning and conventional cartoonification algorithm was provided to convert real-world images into appealing cartoony types of images, while maintaining all of the required image features such as edges, contours and colour blends. Unlike the popular use of Deep Learning-based approaches, such as Convolutional Neural Networks (CNNs), Neural Style Transfer (NST) and Generative Adversarial Networks (GANs), the approach does not require the computationally intensive and complex training datasets. The cartoonification pipeline was designed in modules consisting of an image preprocessing stage involving colour space changes, converting colour images to grey images, applying Median Blur Filters to reduce noise, performing Adaptive Threshold Edge Detection, Colour Quantisation Through K-Means Clustering and creating a cartoon-like effect using Bilateral Filtering to smooth the image edges. By the use of bitwise AND, combine the structural edges of the image with the Simplified Colour Regions to create the final cartoony image. Since the study and framework were independently validated and used to create consistently high-quality cartoony images with the same level of quality, the studies also demonstrated the ability of the system to perform efficiently and effectively across all image types and resolutions. The framework for the Cartoonification Pipeline was built and can run on the Desktop Computer platform; an Android mobile application was created using Kotlin that allows users to generate and use the cartoon-style images generated from their own digital images; the application demonstrates the efficiency, scalability, and performance of the cartoonification pipeline in real time and in resource-limited environments.

Keywords: Cartoon image generation, image processing, machine learning, K-Means clustering, bilateral filtering.

1 Introduction

The advancements in Digital Image Processing and Computer Vision have been rapidly accelerated by the increasing spike of Multimedia Applications, Socialmedia, Photography via Mobile Devices and Creative Generation of Digital Content. Of all of these applications, Artistic Image Stylisation has received a lot of attention because many users are interested in doing a creative transformation of a photograph into a more visually appealing image while maintaining the original Semantics of the picture. One very popular form of artistic image stylisation is the creation of Cartoon-Style Images. There are a Lot of uses for this type of stylisation, from entertainment, education, gaming, to Mobile Applications. The ability to easily transform a real-world photo into a cartoon representation has brought about an automated way of doing digital art that is extensive and Eco-friendly to the user, who does not have to possess professional skills to create a high-quality work of art [1].

The related computational complexity associated with generating cartoon images is unclear because of the technological and computer science aspects of the process involved. In generating cartoon images, the algorithm must abstract real-world photographs into cartoon representations by executing an analysis of the original photograph's features while maintaining important aspects of the original image, such as edge detail, colour areas and other critical features of the image. Too much edge smoothing will remove some of the physical edge details, whereas too much edge enhancement will create disruptive visual artefacts and unnatural textures associated with the edges. An ongoing research effort is occurring to design a robust algorithm for generating consistent cartoon-like images for a variety of photographs and their various resolutions [2][3].

In the past, cartoons were created using a combination of hand-drawn illustration and digital painting techniques. Both of these methods have been proven to work well, but they can take a lot of time to create a quality product and rely on the expertise of an artist. Because of their limitations in terms of speed and scalability, many new methods for creating cartoons have been developed. "Learning-based" methods (such as Convolutional Neural Networks (CNNs), Neural Style Transfer (NST), and Generative Adversarial Networks) are excellent examples of these types of techniques [5]. While the learning-based methods can produce excellent stylization results, there are some barriers to using them effectively, including the need for large labeled datasets to train on and lengthy training times; all three of these learning-based techniques require a significant amount of computation. In addition, because most of the learning-based approaches are designed to be used on high-powered computers, it is not easy to build devices that can take advantage of them for lightweight and real-time applications. Alternative solutions for cartoonifying images that utilise hybrid methods instead of only deep learning techniques can be very effective. A good solution should produce good quality results that are fast and easy to understand [6].

If traditional methods of processing images are combined with machine learning algorithms for producing images with varying colours, then a system can be built that will produce good results and will save computing power and time. Hybrid systems are especially helpful for developers who need to produce images for desktop development environments where resources may be limited or for mobile devices. This paper proposes a hybrid approach to generating cartoon images using a combination of machine learning techniques while maintaining edge preservation, colour consistency and computational simplicity [7]. A structured image processing pipeline is used to produce cartoon images. This includes pre-processing images with grayscale conversion, noise reduction, especially adaptive edge detection methods, and colour quantisation through the K-Means clustering algorithm and edge-preserving smoothing methods. The novelty of this system is that these methods preserve important visual characteristics for generating cartoon-style images while discarding unwanted textures and excessive amounts of color. An advantage over using deep neural networks is that the results can be achieved without training on huge datasets or using highly complex models, which enables use on standard desktop computers [8].

This research focuses on automating the process of cartoonification, while at the same time preserving the visual quality of the images and improving the speed at which they will be cartoonified (operational efficiency). The goal is to develop an automated cartoonification system that preserves the most prominent features (edges and contours), produces smooth full color areas and provides consistent visual output across a large spectrum of images (from natural scenes to scientific illustrations) [9][10]. Furthermore, the modular conception of the framework allows for improved clarity of design, scalability and ease of maintaining and extending the functionality of each of the individual processing components. The prototype has been developed on a number of standard desktop systems and successfully deployed to a mobile Android platform to showcase the effectiveness and versatility of the developed technique in practical, real-world applications.

2 Literature Survey

Researchers from diverse fields, including computer vision, Image Processing and Machine Learning, have studied a broad range of approaches to Cartoonifying Digital Images and generating Stylized Images. Their findings demonstrate that classic Image Processing principles, as laid out can be used to transform grayscale images, remove noise, etc., all whilst preserving the fundamental structure of the image itself. These principles act as the backbone of several stylisation techniques where clear outlines and abstract representations need to be achieved without adding significant processing load.

The traditional works in computer vision show that edge-based representations and region-based segmentation effectively reduce the complexity of an image's visual representation. Providing deterministic and understandable methods for creating visual representations, or models, enables the use of these techniques in applications with limited time and computational resources, and for real-time operation. Additionally, adaptive thresholding techniques enhance the robustness and reliability of edge extraction in low-light, high-contrast, or otherwise dramatically altered lighting conditions, which is essential for producing consistently high-quality cartoon-like images.

Image abstraction employs techniques based on clustering methods have done an in-depth review of clustering algorithms and concluded that K-Means is a reliable, scalable solution for Color Quantisation. They suggest that K-Means is widely used within stylisation operations as a technique to reduce color variance whilst still being visually effective. This was further supported from Bishops findings when he investigated how unsupervised learning could be effectively engaged within visual data in Pattern Recognition, not needing an annotated data set to group them visually. In contrast, the Deep Learning approaches have been used by CNNs, NST, and GANs using similar techniques to generate impressive stylization methods [2]. However, they need to rely on large data sets and consume significant computing power and extensive training to develop the best results, limiting their usability for low-power applications. Therefore, most papers consider these models as High-Performance but not economically viable alternatives.

The use of libraries like OpenCV and Scikit-learn has resulted in the quick creation of hybrid systems for processing images using machine-learning algorithms. Both libraries provide an optimized implementation of the various methods used for filtering, clustering, and transforming images; therefore, making Hybrid Systems available for immediate implementation. Furthermore, the development frameworks found within Android Development Frameworks have provided application developers the ability to create mobile applications that allow the use of such hybrid systems, thereby making them accessible to end-users and allowing for real-world applications of these systems [11]-[13].

3 System Architecture

The architecture of the Cartoonify Image Using Machine Learning system can be described as having a sequential and modular pipeline; that is, each component serves an explicit purpose by transforming a photograph or other real-world image into a cartoon. Additionally, the structure of this system has been designed in such a way as to be clear in its design and function, be efficient from a computational perspective, while preserving (or making evident) visually significant aspects of the image, such as edges, contours and colors

Image Input Module

The Image Input Module is the first step in creating the user's visualisation. It does this by retrieving the Image Source used to give input for processing. The Image Source can be selected from Local Storage or captured directly through the Camera Interface, as it pertains to the Android-based implementations of the Image Processing System. The Image Input Module also verifies that the retrieved image meets the parameters of a valid Digital Image Format and Resolution for use in the following Image Processing stages. By specifying and maintaining a consistent workflow in the Image Input Process, it is possible to ensure that the Image System works effectively across multiple Image Sources and Devices.

Preprocessing Module

The preprocessing pool works to prepare an input image for efficient extraction and modification of features. The first step in the preprocessing pool is to convert the image from BGR colorspace to RGB colorspace to meet the standard image processing formats. The second step is to create a grayscale version of the image; this reduces the number of calculations that need to take place when performing intensity-based operations. Lastly, applying median blur filtering when removing noise, Light inconsistencies affect uniformity in the color variations of the picture with respect to minor Texture. This is mainly used to stabilize the image and add reliability to both edge detection and clustering algorithms in the later stages.

Edge Detection Module

Instead of using fixed threshold values as an edge detection technique to identify shapes and forms that define objects in a cartoon image, use an adaptive thresholding approach that is based on local characteristics of the image and is adjusted for changing light levels. Therefore, extract very accurate edges that accurately represent the shape of the cartoon images. This original outline of the cartoon image contains all the defining features of the scenes and characters of the cartoon image.

Color Quantization Module

In the colour quantisation module, K-Means clustering reduces the colour complexity of the RGB image. Pixels in the RGB image are grouped according to their 'color similarity' into a pre-determined number of clusters. For each pixel, the nearest cluster centre is found, and the pixel is assigned to that cluster. This reduces the number of distinct colors significantly and maintains the overall visual structure of the image. The result is a collection of flat, homogeneous color regions that represent the 'cartoon' style and are part of the overall 'stylised' visual abstraction.

Edge-Preserving Smoothing Module

Using Bilateral Filter edge- preserving smoothing module, to improve the look and feel of the Quantized Image through this filtering technique which takes into account both distance in space and the difference in intensities of colors and helps to smooth out colors while keeping a distinct edge within a color region from one to the next; this also allows for unwanted textural information on a color region to be reduced or removed, while keeping important boundaries from becoming blurry. As a result, the simplified color areas are seen to be visually smoothed out while having continuity with the edge clustering that was performed on the quantized image.

Integration and Output Module

Combining the Retained Edge Map with the Smoothed, Color-Quantized or Simplified Image using Bit-wise Operations results in an output image consisting of a combination of strong and expertly defined Edges placed over simplified color Regions, creating the final Cartoon Image. This output image can either be displayed or stored for later use within the application, and completes the Transformation to provide a Continuity of Appearance between original input images and Cartoon representations based on those input images.

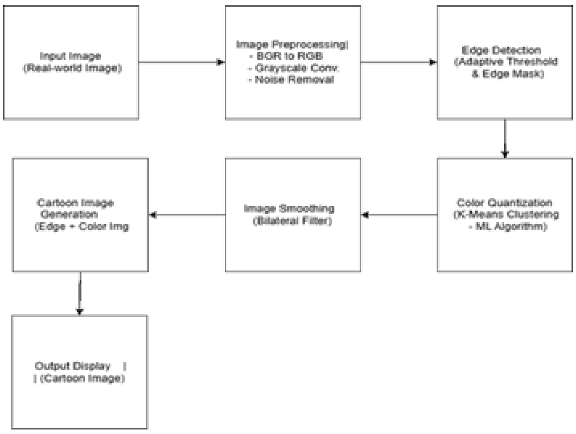

Fig. 1. Architecture Diagram of the Proposed System

4 PROPOSED SYSTEM

The approach taken in this research paper is a structured and modular approach for converting natural images to cartoon images by means of a combination of using computer vision technology, with a method of Machine Learning-based clustering technology to create these images. The design philosophy of the workflow is to create simple drawings of only the important features of the original image, allowing the effective and efficient production of professional quality cartoon images. The workflow consists of a defined set of deterministically defined steps that successfully use the principles of unsupervised learning to operate on smaller data sets; therefore the entire workflow can successfully operate on low-end computing platforms efficiently. The output from each stage of the proposed workflow builds on the output of the previous stage, building towards an ultimate output that is high quality, has a high level of reliability and is scalable.

Image Acquisition and Input Handling

Cartoonification begins when an image is obtained (from either a storage device or camera interface). The system supports various standard image formats that allow users to import images with a range of brightness and resolution. Once selected, the image is brought into the system memory as a multi-dimensional matrix (of pixels). This ensures that the image is correctly initialized in its original form so that it can be processed in the same uniform manner throughout the cartoonification process. In some cases, it may be necessary to resize or adjust the image dimensions to improve processing times and visual quality, particularly for locations with limited resources.

Color Space Conversion

Typically, images are converted from the OpenCV internal BGR image format to the commonly used RGB format for processing and visualization. In order to create a consistent foundation for developing other image processing algorithms through the use of a common color definition (RGB), it is important that any input images be converted from BGR to RGB before beginning the other stages of processing them. This conversion not only will prevent any distortion of the original colors, but will also provide a more accurate method for defining and describing any future transformations that may be made to that image.

Grayscale Conversion

The aim of grayscale conversion is that by eliminating the color element from the RGB image, the intensity data will be extracted and a reliable method for edge extraction may be provided. When a color image is converted into a single-channel image (grey), the overall amount of data represented decreases because there is only one channel to hold all of the intensity information. In addition, this simplification maintains structural features, including texture, object boundaries, and intensity gradients. Furthermore, the performance of edge detection is improved due to the fact that the impact of color variation no longer has any significant impact on structural analysis when a grey image (and not a color image) is used.

Noise Reduction Using Median Blur

Noise in a natural image is frequently created through the process of image acquisition or compression or transmission. Because of this noise, it is possible for false or fragmented edges to be formed. Therefore, a median-blurred filter is utilized to remove the noise from the grayscale image. The median filter is a type of non-linear noise filter that replaces every pixel value of the image with the median value of the pixel's neighborhood (the values of the pixels surrounding it in a square). The median filter works particularly well to eliminate impulse noise while keeping sharp edges intact, as they are important when rendering an image in a cartoon style. This filtered image then acts as a stable and noise-insensitive base for detecting the edges accurately.

Edge Detection via Adaptive Thresholding

Detecting edges is an important step for cartoonifying an image because cartoon images are defined by strong, stylised edge lines. In this paper, it is proposed to use adaptive thresholding in order to find edges from the noise-free grayscale image. Adaptive thresholding differs from global threshold methodologies, which use one threshold value for the entire image, and it calculates local threshold values for pixels from their surrounding pixel values. As a result, even in poor lighting and uneven illumination conditions or varying contrasts, the edge extraction will provide consistent results. The result of this stage will be a binary edge image, which displays the major edges and boundaries of objects that provide the framework for the cartoon image.

Color Quantization Using K-Means Clustering

Color quantisation is a K-means clustering process that gives cartoons the flat, colored, simplified look. The pixel color values are treated as unlabelled feature vectors in this unsupervised machine learning process. They are then grouped into clusters of similar colors found in the color space according to the requested number of clusters.By reassigning the centroid color of each pixel based on which type of region it belongs to, the number of unique colors can be reduced in an image, and this reduces overall visual clutter, the meaning of the image remains intact. The K-Means method for clustering allows us to create equal-sized areas with similar colors in a single image without having labelled training data, meaning that the method allows it to be applied to any combination of photographs, drawings, etc.

Edge-Preserving Smoothing with Bilateral Filtering

Bilateral filtering is the technique used to smooth out the color of the image created from the quantized color palette, while preserving edges. In bilaterally filtered images, spatial distance from the reference point and the intensity of each pixel at the reference point are taken into account in determining which pixels will be used to create the averaged color value. The filtering technique used in standard smoothing filters will create blurred edges, whereas bilateral filtering preserves sharp edges while smoothing color variations in similarly colored areas. In addition, applying bilateral filtering to the clustered color regions will improve the overall continuity and smoothness of colors, remove small errors resulting from clustering, and continue to provide the expected cartoon-like style of coloring.

Bitwise Integration of Edges and Color Regions

Using bitwise operations, the final cartoonified image is created by combining the edges and the color-quantised and smoothed image. In addition, the detected edges are overlaid on top of the smoothed color image, allowing for bold outlines to contrast against simpler colored areas, which is one of the most important components of the cartoon effect, as it emphasises the structure of the image and minimises the incorporation of any texture details that are not necessary. As such, the finished artwork contains a degree of abstraction along with enough recognizability to remain familiar.

Output Generation and Visualization

In the last part of the process, the created cartoon picture is processed and shown on the user interface. It is also possible to save the end product for use later, sharing with others, or analysing it. Since the method proposed has a modular architecture, any processing step may be separately improved or substituted, offering future expansion opportunities for the capability to perform a live video with cartooniser, adjusting the parameters, and incorporating into a system based on advanced learning methods.

5 RESULTS AND DISCUSSION

The results of the Cartoonify Image Using Machine Learning System as described in this section are thoroughly analysed. This analysis allows us to determine how well the system converts a real image to a cartoon image while retaining the visual importance of an image. In order to do so, the evaluation of the final output is examined for quality, edge clarity, simplified color representation, amount of computing resources required to run this system, and system performance overall. Systematic observation of the output and intermediate output of each part of the pipeline allows for a thorough analysis of the results.

Analysis of Cartoonification Output Quality

The results of this study demonstrate the ability of the proposed system to generate images that resemble cartoons created from photographs. The resulting images all have a similar cartoon-like look and feel that is defined by a simplistic color palette, a strong emphasis on borders and edges, and a lack of detail or richness compared to the original images. However, even with these features, the visual cues (i.e., face, object and environment) from the original photos have been preserved to allow for easy recognition of these cues. This system accomplishes a proper balance between the expectations for a realistic representation of the original content and usefulness from a stylistically appealing standpoint. By properly controlling how combine the processing of images with the machine-learning model, potential degradation is prevented in visual quality due to the overuse or misordering of images within a sequence of images. The effect of a "cartoonify" image is similar to that which you would obtain when creating a similar image using traditional hand-drawn techniques. The ability to produce itemised and stylised artwork, mobile camera apps, and educational tools illustrates that the goal of an automated technique for creating cartoon images has been achieved.

Edge Preservation and Boundary Representation Analysis

The edges of objects provide a structural basis for a cartoon-style image. They allow for the creation of visually expressive objects by defining the boundaries of an object, where one stops, and another begins. In this work, significant edges will be detected using adaptive thresholding methods on the grayscale image. Before the application of these adaptive thresholding methods, a median blur filter will be used to reduce noise in the image; thus, suppressing any small variability in intensity levels due to the presence of noise, thereby ensuring that all detected edges represent actual objects rather than unwanted characteristics.

Different types of images reveal strong and continuous edges, such as the facial outline, the object contour or outline, and the large shapes within the image. Since the thresholding technique employed is adaptive, the edge detection can be accomplished in similar conditions (e.g., variations in lighting, contrast) without failure. In conjunction with the colour-quantified image, the effective cartoon effect of adding the edges to the colour-quantised image via bitwise operations results. Therefore, the success of the edge detection and integration strategies implemented in this system is fully supported.

Detailed Evaluation of Colour Quantisation

Colour simplification is essential to achieve a cartoon-style look to an image. For example, the proposed system uses K-Means clustering to perform colour quantisation. This means that similar shades of color are grouped, resulting in fewer colors being used in the final image. Additionally, using K-Means clustering converts random, chaotic distributions of colour into cohesive, homogeneous regions of colour that create the typical style of a cartoon image.

Findings demonstrate that K-Means clustering effectively preserves dominant hues within the images while removing extraneous hues. The addition of K-Means clustering provides a way to control the degree of abstraction using the number of clusters as a variable, thus enhancing the adaptability of the system to various types of images (e.g., portrait images, object images, and exterior scene images). In addition, the outcomes associated with K-Means clustering provide assurance and predictability with respect to the consistency and dependability of the overall system.

Impact of Edge-Preserving Smoothing

Before Bilateral Filters were applied, quantising an image resulted in the loss of an entire channel. In addition, unlike conventional smoothing filters, Bilateral Filters were able to maintain the color representation within the boundaries of each of the channels. This generally enhances the quality of the rendered cartoon images. Through this process, the processed images feature color regions that are free from rough or jagged edges as well as poor definition. When using Chromaticity as a clustering tool, noise was introduced into some of the images and color discrepancies were recorded. Both of these issues were minimised in the final output of the cartoon versions. Combining Bipolar Color Quantisation and a Bipolar Filter results in increased quality of the cartoon images. Therefore, these results validate the benefits of the proposed image processing Method.

Computational Efficiency and System Performance

To determine the practicality of applying this system to real-world scenarios, the computational performance of the system was analyzed. The analysis shows that the proposed system can run effectively on standard desktop computers without the need for specialised hardware. In addition, because the proposed system does not depend on deep learning models, there are no requirements for training data sets, nor are there any requirements for GPU acceleration. The low computation required for the developed technique allows for quick processing of images while minimising memory consumption, and therefore the fact that it was written in KOTLIN shows, in addition to the above advantages, the lightness of the system. Images can be processed in a small amount of time, providing users with a smooth and responsive experience when using the application. Thus, the developed system is appropriate for use in real-time mobile applications.

Fig. 2. Sample original input image before processing

International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 3, pp.42-49, March 2026.
www.ijersem.com eISSN- 3107-9075

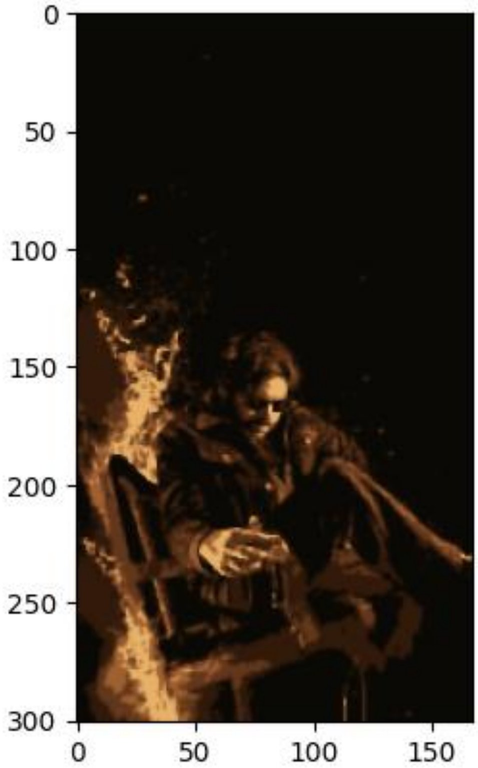

Fig. 3. Output image after applying the proposed cartoonification pipeline

6 CONCLUSIONS AND FUTURE SCOPE

This paper showcases the creation of an automated mechanism that will convert real-world images into cartoons using a combination of machine learning and image processing techniques. The methodology reduces visual detail while preserving important visual characteristics of images (i.e., Edge, Contour and Color consistency). The automated system produces visually coherent cartoon outputs by using grayscale conversion, median blur-based noise-reduction, adaptive threshold edge-detection, K-Means Color quantisation and Edge-Preserving Smoothing techniques. The analysis of the methodology demonstrates the balance that exists between artistic abstraction and structural fidelity; this is essential for the creation of high-quality cartoon images. While deep learning-based methods tend to require a lot of processing power, the approach allows users with less powerful computers to still be able to access and interact with the model due to lower computational requirements. Furthermore, the modular and sequential architecture makes it easy for users to upgrade, enhance, or expand the system as needed. Additionally, this application runs on a Windows platform in addition to being available in an Android-based mobile application. Through experiments, it was seen that the same quality resolution performance was obtained from various images, and when adding up those results, it proved that integrating Classical Image Processing with Unsupervised Learning could produce digital cartoons in media applications.

The newly designed method to cartoonize photos, utilizing machine learning technologies, serves as a great stepping stone to automated photo cartoonization technology, yet there are several avenues available for enhancement. More advanced types of machine learning models can be incorporated into the Cartoonify Image Using Machine Learning platform, including: Convolutional Neural Networks, Generative Adversarial Networks, and Neural Style Transfer. Using these types of models would result in a more comprehensive range of realistic and unique cartoon styles, including: anime, comics, and hand-painted cartoon styles. The development of the system into a real-time, automated cartoonisation solution for both live video streaming and camera feeds (e.g. Facebook Live) will provide users with more interactive features. Enhanced functionality in the mobile application can include style selection, intensity adjustment, batch processing of images, and the ability to perform calculations in a cloud-based environment, improving both usability and performance on low-end hardware. The current framework could be extended to allow for animating videos through frame-by-frame processing, thereby providing new opportunities for content developers and animators. As well as extending to offer a cross-platform deployment capability for iOS, Web and Desktop environments, and integrating into the existing professional image editing market, provides a significant opportunity for creating better access, greater scaling and improved practicality.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ETHICS STATEMENT

This study did not involve human or animal subjects and, therefore, did not require ethical approval.

STATEMENT OF CONFLICT OF INTERESTS

The authors declare that they have no conflicts of interest related to this study.

LICENSING

REFERENCES

[1] Y. Dong, L. Li, and L. Zheng, "TSGAN: A two-stage interpretable learning method for image cartoonization," *Neurocomputing*, vol. 596, p. 127864, May 2024, doi: 10.1016/j.neucom.2024.127864.

[2] C. Tumer, E. Guvenoglu, and V. Tunali, "Robust Multi-Label cartoon character classification on the novel Kral Sakir dataset using deep learning techniques," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 85, no. 3, pp. 5135–5158, Jan. 2025, doi: 10.32604/cmc.2025.067840.

[3] H. Jeon, J. Shim, H. Kim, and E. Hwang, "A photo cartoonization method based on text-to-image diffusion model," *Neurocomputing*, vol. 620, p. 129221, Dec. 2024, doi: 10.1016/j.neucom.2024.129221.

[4] Z. Qi, D. Pan, T. Niu, Z. Ying, and P. Shi, "Bridge the gap between practical application scenarios and cartoon character detection: A benchmark dataset and deep learning model," *Displays*, vol. 84, p. 102793, Jul. 2024, doi: 10.1016/j.displa.2024.102793.

[5] A. T. Hoang *et al.*, "Machine learning supported measurement and control for line drawing and shading of a portrait drawing robot," *Measurement*, vol. 262, p. 120022, Dec. 2025, doi: 10.1016/j.measurement.2025.120022.

[6] H. Chen, X. Wang, and F. Shao, "Blind cartoon image quality assessment based on local structure and chromatic statistics," *Journal of Visual Communication and Image Representation*, vol. 101, p. 104152, Apr. 2024, doi: 10.1016/j.jvcir.2024.104152.

[7] Y. Liu, Z. Qin, T. Wan, and Z. Luo, "Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks," *Neurocomputing*, vol. 311, pp. 78–87, May 2018, doi: 10.1016/j.neucom.2018.05.045.

[8] Y. Tang, "ECGAN: Translate real world to cartoon style using enhanced Cartoon Generative Adversarial Network," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 76, no. 1, pp. 1195–1212, Jan. 2023, doi: 10.32604/cmc.2023.039182.

[9] T. Zhang, Z. Zhang, W. Jia, X. He, and J. Yang, "Generating Cartoon Images from Face Photos with Cycle-Consistent Adversarial Networks," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 69, no. 2, pp. 2733–2747, Jan. 2021, doi: 10.32604/cmc.2021.019305.

[10] Z. Liang, Y. Zhuang, Y. Yang, and J. Xiao, "Retrieval-based cartoon gesture recognition and applications via semi-supervised heterogeneous classifiers learning," *Pattern Recognition*, vol. 46, no. 1, pp. 412–423, Jul. 2012, doi: 10.1016/j.patcog.2012.06.025.

[11] M. Jalali, H. Behnam, and M. Shojaeifard, "Echocardiography image enhancement using texture-cartoon separation," *Computers in Biology and Medicine*, vol. 134, p. 104535, May 2021, doi: 10.1016/j.compbiomed.2021.104535.

[12] F. Zhang, H. Zhao, Y. Li, Y. Wu, and X. Sun, "CBA-GAN: Cartoonization style transformation based on the convolutional attention module," *Computers & Electrical Engineering*, vol. 106, p. 108575, Jan. 2023, doi: 10.1016/j.compeleceng.2022.108575.

[13] Z. Huang *et al.*, "A machine learning based method for tracking of simultaneously imaged neural activity and body posture of freely moving maggot," *Biochemical and Biophysical Research Communications*, vol. 727, p. 150290, Jun. 2024, doi: 10.1016/j.bbrc.2024.150290.