



ISSN (E): 2277- 7695  
ISSN (P): 2349-8242  
NAAS Rating: 5.03  
TPI 2019; 8(1): 701-706  
© 2019 TPI  
www.thepharmajournal.com  
Received: 22-10-2018  
Accepted: 26-11-2018

**Aastha Budhiraja**  
Assistant Professor,  
Computer Science &  
Engineering, Lingaya's  
Vidyapeeth, Faridabad,  
Haryana, India

## Image cartoonification using machine learning: Transforming visual content with artistic intelligence

**Aastha Budhiraja**

DOI: <https://doi.org/10.22271/tpi.2019.v8.i11.25405>

### Abstract

Within the field of artificial intelligence (AI), machine learning simulates human learning by utilizing data and algorithms to progressively increase system accuracy. With machine learning, a user can feed massive amounts of data to a computer programme, which will then analyse it and draw conclusions and recommendations based only on the data it is fed. In this research, the topic of conversation is cartoonizing an image. Your submitted photo will be transformed into a distinctive cartoon by Cartoonify using a neural network. A number of software programs, including Paint.net, Photoshop, Adobe Illustrator, Windows MAC, and others, can be used to turn an image into a cartoon. the assortment of Python libraries. For better results, use Python libraries rather than online tools like Photoshop. OpenCV is one of these libraries. A cross-platform computer vision library is called OpenCV. Among them are programs for taking and manipulating pictures and videos. Picture editing, object identification, facial recognition, and a host of other amazing uses are among its primary uses. Some common libraries, like Numpy and Matplotlib, will be utilized. You could turn your photo into a cartoon, use it as your profile picture, or create a hilarious avatar. Machine learning may be used to create such projects. Consequently, We are going to write a Python script that uses OpenCV to create a cartoon out of an image. Python programming is necessary for machine learning applications. The main finding of the study is how to use machine learning to cartoonify an image.

**Keywords:** Cartooning, machine learning, and open resumes

### Introduction

Cartoons are frequently used in a range of settings. Cartoons, as we all know, are artistically demanding and require the most sophisticated and graceful skills <sup>[1]</sup>. Since the artist must faithfully recreate the cartoon's sketch in order to achieve a pleasing result, it takes a lot of time to illustrate cartoons in large quantities for animated films. One way that cartoons are used outside of the film industry is that people attempt to turn their own images into cartoons to see how they appear in the cartoon effect. But because this procedure takes so much time and calls for so many resources, it could be challenging to gather them all and carry it out properly. The time required by artists in films is decreased by this cartoonizing process. With this method, they may get the same results as previously. However, cartooning a picture is effective and rapid compared to drawing one. The code for this approach is written in the Python programming language <sup>[2]</sup>. In order to obtain the most accurate result, many Python modules are employed. This method of cartoonizing mostly uses OpenCV, one of the Python libraries. The majority of computer vision applications in the domains of machine learning and artificial intelligence use OpenCV, a free and open-source Python library. In the tech industry, OpenCV is highly well-known at the moment. OpenCV can be used for a wide range of tasks on images and videos, such as object detection, object tracking, and face detection. Many different operating systems, such as Windows, Linux, Android, Mac OS, iOS, and all others, are compatible with Open CV <sup>[3]</sup>. It offers C, C++, Python, Java, and C interfaces. Second point. Python is necessary: Python is the programming language that we use to create the application. CV2: We use cv2 for image processing with Numpy: The main application for NumPy is array manipulation. The images we use here are stored in an array format. We make use of NumPy for that. Matplotlib: This programme is used to visualise data. Here, we use Matplotlib to plot the photos. OS: The OS is used to read images from the path and save the picture to the path when interacting with paths in our application. No. 3, No. 11, International Journal of Research Publications and Reviews 1082 November 2022 1.3., pages

### Correspondence

**Aastha Budhiraja**  
Assistant Professor,  
Computer Science &  
Engineering, Lingaya's  
Vidyapeeth, Faridabad,  
Haryana, India

1081–1089. Bringing in the required elements for cartoon image development 2<sup>[4]</sup>. Constructing a File Box to allow for file selection. 3. 4. Converting an image to grayscale 5. How are images stored? Grayscale image smoothing 6. Restoring an image's borders 7. Image 8: Mask Preparation. Assembling an animated effect 9. putting every plot point in order of harmony 10. The functionality of the download or save buttons.

### Review of Related Works

The following research articles are carefully read in order to understand the experimental results for output forecasting as well as the recommended approach<sup>[5]</sup>. 2 points. Bollepalli Pranitha, MD. Salar Mohammad, Pulakanti Teja Sree, and Shivani Goud Pandula. (2021). "Cartoonizing the Image," July–August 2021 issue of *Advanced Trends in Computer Science and Engineering: An International Journal*, Volume 10, No. 4. In this work, a picture is cartoonized using a processing method. Image manipulation tasks like object detection in images, object size and quantity estimation, and picture blurring are all highly valued in the field of image processing research. The Image Processing has a variety of features. Each of the properties predicts that the image will be created with increased detail and sharpness. Each image is evaluated using a different grid<sup>[6]</sup>. When taken as a whole, each component of the image is seen as a 2-D matrix. For each cell, keep several pixel values determined by each component of the image. 2. "Cartoonizer: Transform Pictures and Videos into Cartoon-Style Pictures and Videos." S. Neha L., Rajatha, and Anusha Shrikant Makkigadde. Volume 4, Issue 7, July, *International Journal of Science, Engineering, and Management Research*. K and Kanchan. Sapna. Bhat Janardhana<sup>[7]</sup>. This study uses Cartoon GAN, which stands for Generative Adversarial Network, to the loss function, which is available in adversarial loss and content loss flavors, allowed us to create adaptable and precisely defined images<sup>[8]</sup>. Additionally, we have converted video into animation (cartoonized video) with the aid of CV2, or computer vision. 2 points. "Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks," *International Conference on Image Processing*, 2017, Zengchang Qin, Zhenbo Luo, and Hua Wang. Realistic image creation with deep neural networks has gained popularity recently in computer vision and machine learning. In this paper, Conditional Generative Adversarial Networks (CGAN) are used to investigate the sketch-to-image synthesis problem<sup>[9]</sup>. Based on two sketch datasets, experiments reveal that the auto-painter outperforms current image-to-image techniques. In addition to applying the proper color to hand-drawn sketches, the new model allows users to designate their preferred color scheme. The supervised learning model Auto-Painter can create a painted, colorful image from a black-and-white sketch by using sketch image pairings from the training data. Two things. Muttur, Kavya S., Harshitha R., and Prof. Jyothi Shetty, RV College of Engineering's Computer Science Department, Bengaluru<sup>[10]</sup>.

This article uses the 2020 "Cartoonization Using White-box Technique in Machine Learning" Using the white-box technique, the image is cartooned. This study uses an artist consultation and cartoon painting process to independently identify three white-box representations from images. Vol. 11, no. 11, November 11, 1083, 2022 *Journal of Research Publications and Reviews International*, No. 3, No. The representation's cartoon-like surface lies between 1081 and

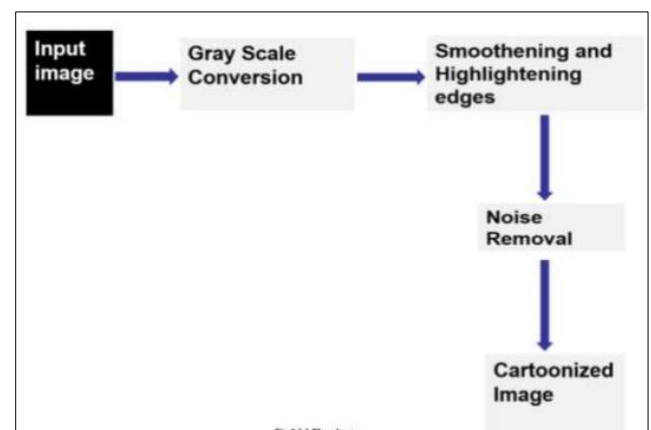
1089. The celluloid-style workflow is the structural representation that flattens global content and links to sparse colorblocks. Furthermore, the high frequency curves and features present in cartoon images are captured by the texture representation<sup>[8]</sup>.

2 points. Professor, Navjeevan Bomble, Ashwathy Unnikrishnan, and Akanksha Apte. "Transformation of Realistic Images and Videos into Cartoon Images and Videos using GAN," by Sachin Gavhane, was published in July 2021 in the *International Journal of Science, Technology, and Management Research*, Volume 4, Issue 7. A white-box programmable image cartoonization framework is suggested in this study as a way to turn real-world photos and videos into high-quality cartoonized images and movies. There are three distinct cartoon representations of an image: texture, structure, and surface. utilizing models for image processing like PIL, OpenCV, etc. in order to train the network, three representations are extracted. and the relative weight of each representation permits various styles of results<sup>[9]</sup>. Numerous qualitative, quantitative, and user studies have been carried out to verify the effectiveness of the technique.

### Methodology

#### Cartoonizing the Image

One can initially divide the process of making a cartoon effect image into two groups. -To recognize, emphasize, and distort an actual RGB color image's edge. to format, smooth, and quantize an RGB picture into grayscale<sup>[11]</sup>. The intended outcome is aided by the integration of the images. The user-provided image is first converted to grayscale before being transformed into a cartoon. This process starts with the image being converted to grayscale. We thus employ the BGR2GRAY flag. This generates a grayscale image. An image in grayscale is saved as a grayscale image in the file.



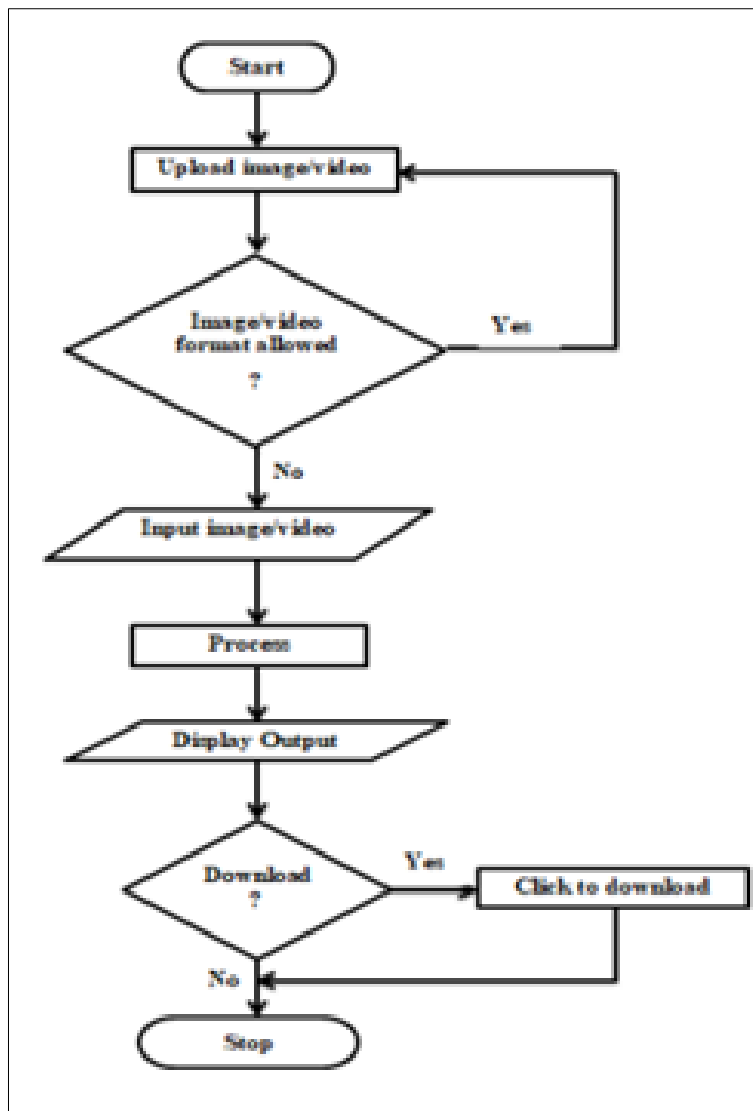
The blur effect only helps to smooth out an image. To achieve this, use the median Blur () function. A bilateral filter is employed to reduce the noise. Consider it a gentle version of image smoothing. Here, finding and emphasizing the edges will be our main objective<sup>[12]</sup>. The adaptive thresholding approach achieves this. We mask two images using a bitwise and. Finally, this CARTOONIFIES our picture.

#### Cartoonizer: Convert Images to Cartoon-Style Image

The Cartoonizer web application's flowchart is shown in the above figure. The user initiates the process by clicking the button to choose the image or video. The system will then display a window for local storage after that. The system

verifies the authorized picture and video formats. Subsequently, the chosen picture or video will be uploaded in order to be animated. After an image or video is successfully uploaded, the system processes the data, evaluates it, and generates the output. Once the cartoonized output has been shown on the screen, the user can download the image or video by clicking the download button <sup>[13]</sup>.

The downloaded file will be kept locally as a result. The user now has the option to log off the system. A video's file size must be 30 MB or less when uploading it. The video is trimmed to 15 seconds if it is longer than that and made like a cartoon in a video. There will be audio in the animated video. We'll just examine the first fifteen seconds of the video.



**Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks**

Encouragement The final colorful cartoon image is produced by concatenating encoder layer A and decoding layer A', where A' primarily consists of trained color painting information and A contains the sketch edge information, as shown in Figure. This makes it possible for us to interpret the drawing's low-level data. The encoder's layers are represented by yellow blocks in the first figure, while the decoder's layers are represented by blue blocks. In order to decode the layer that comes after, the relevant encoder layers are concatenated with the current layer during each layer of decoding. For instance, Deconvolution is used to combine B and B' to create A' <sup>[11]</sup>. The patch GAN discriminator is shown in the image on

the right. It takes two target images (red blocks) and two sketches (yellow blocks) or two sketches and generated images (blue blocks) as inputs. The discriminator only needs to compare the encoder units to the generator to determine whether the input sketch-image pair is "real" or "fake." The network is trained to improve the classification accuracy. The discriminator's output is displayed in Figure as a probability matrix, each member of which represents the chance that two related patches gathered using Patch GAN or a Markov random field will in fact be authentic. In our research, We output a probability matrix with the patch size of 70\*70 and a 30\*30 size. The input pairs' high-level characteristics are extracted by the convolutional layers between the input and the output <sup>[14]</sup>.



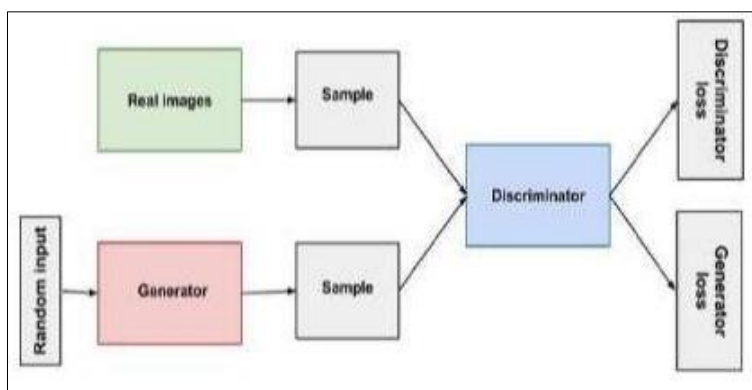
**Cartoonization Using White-box Technique**

With the aid of joint probability, it serves as an image synthesiser to produce fresh data. Generator and Discriminator are used to create fresh pictures [15]. The generator creates pictures, the discriminator determines if they are real or phoney, and after receiving feedback, the generator is asked to produce better data. Better pictures are produced when both networks are taught more. The Surface Representation, Structural Representation, and Texture Representation algorithms split the input image into three parts. A GAN model is presented, consisting of a two discriminators (Ds and Dt) and a generator (G). Whereas Dt's job is to extract textural information from model outputs and cartoons, Ds aims to characterize surface characteristics retrieved from them, Ds aims to characterise surface characteristics retrieved from model outputs and cartoons. We employ a pre-trained VGG Network to extract high-level characteristics and place a spatial limit on global content

between outputs and the given matched cartoons. TensorFlow is used to implement GAN. It is suggested that a discriminator Ds be used to determine if the surface properties of the result and the associated cartoon pictures are comparable. This discriminator Ds will also control the generator G to teach it how to extract the surface representation. For area segmentation, we use the Felzenszwalb algorithm. We employ a pre-trained VGG Network to impose a spatial restriction on the global content between outputs and the given matched cartoons.

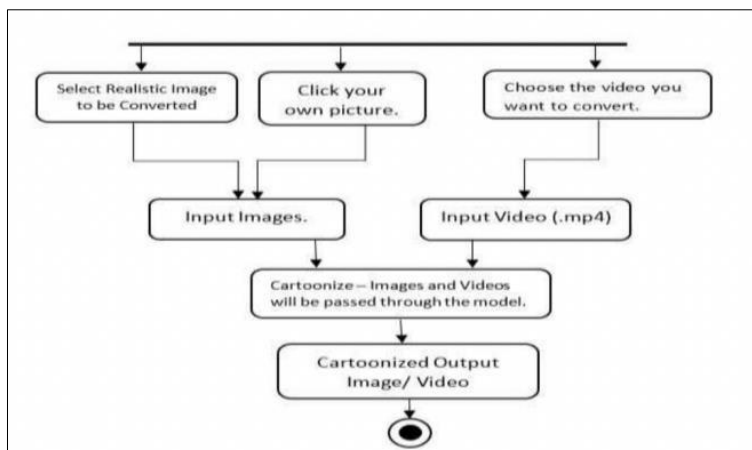
**Transformation of Realistic Images into Cartoon Images using GAN**

The "Generative Adversarial Networks" or GAN algorithm is utilised in the implementation. GAN combines generative modelling with discriminative modelling. The generative model produces fresh data examples that mimic the training set.



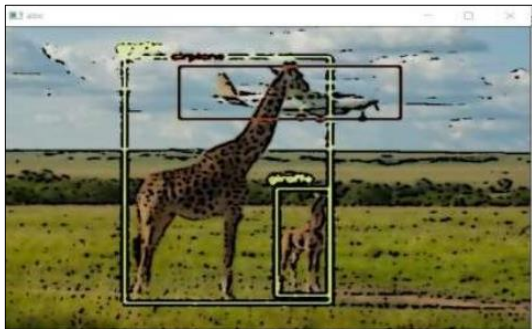
Getting an animated video out of a regular video is analogous to converting a video. In order to create an animated (cartoonized) version of the video, the video is first separated into frames and stored as an array, which is then run through a generator and discriminator. In a similar manner, an image

may be transformed into a cartoon image utilising a variety of ways. The suggested method's functionality is depicted in the activity diagram above. Where it provides a generalised, terse explanation of how cartoon-style (Animation) is created from camera-captured visuals and video.





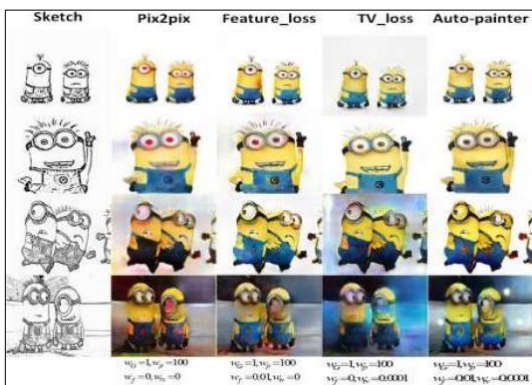
**Results and Conclusion  
Cartoonizing the Image**



**Cartoonizer: Convert Images to Cartoon-Style Image**



**Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks:**



An output picture painted in colour may be produced by the auto-painter in less than one second given an input image with a resolution of  $512 \times 512$ , providing immediate feedback for the development of interactive image editing tools. Users have access to a programme that lets them create virtual images in the "Minions style" using the auto-painter that was trained on the Minions. The initial sketch, the glass, the gesture, etc. may all be modified, as seen on the left-hand side of Figure 5, based on the basic sketch that is provided. It's fascinating how the auto-painter can produce "minions-style" visuals even on simple hand-drawn designs [15].

**Cartooniation Using White-box Technique:** This model has the shortest model size and is the quickest of the four approaches across all devices and resolutions. In particular, this model's ability to process a  $720 \times 1280$  picture on the GPU in just 17.23 milliseconds makes it suitable for real-time, high-resolution video processing workloads. This model was used to simulate a variety of real-world scenarios, including people, animals, plants, and views of the city and its surroundings.



In the findings shown below, the first picture is a still from a video clip that depicts the actual world. Later, using Cartoon GAN and OpenCV, we obtain a cartoonized version of the same image.



**Conclusion**

The primary focus of this essay is on creating high-quality cartoonized images from standard images. And other strategies and procedures, as covered in the publications above, can be used to accomplish this. In general, it takes more expertise, technique, and time to create a quality cartoon from any image, but thanks to modern technology, it has become simple. This procedure takes a remarkably short amount of time. As a result, the most recent techniques for obtaining cartoon graphics gained popularity. There are several methods for cartoonizing an image. The best strategy is employed to produce an effective result based on the user's needs. And as technology advances, several new approaches and procedures are developed for better application and the finest outcomes. These methods may be used to animate videos as well. As a result, many algorithms are put into practise. In the end, it can be said that a picture may be turned into a cartoonized image using the right procedures, which produces effective results in a short amount of time. Despite the fact that numerous programmes have been created for cartooning, Python or any other language implementation results in greater knowledge and better results.

**References**

1. Kaushik P, Yadav R. Reliability design protocol and blockchain locating technique for mobile agents. J Adv Sci Technol (JAST). 2017;14(1):136-141.

- <https://doi.org/10.29070/JAST>
2. Kaushik P, Yadav R. Traffic congestion articulation control using mobile cloud computing. *J Adv Scholarly Res Allied Educ (JASRAE)*. 2018;15(1):1439-1442. <https://doi.org/10.29070/JASRAE>
  3. Kaushik P, Yadav R. Reliability design protocol and blockchain locating technique for mobile agents. *J Adv Scholarly Res Allied Educ (JASRAE)*. 2018;15(6):590-595. <https://doi.org/10.29070/JASRAE>
  4. Kaushik P, Yadav R. Deployment of location management protocol and fault-tolerant technique for mobile agents. *J Adv Scholarly Res Allied Educ (JASRAE)*. 2018;15(6):590-595. <https://doi.org/10.29070/JASRAE>
  5. Kaushik P, Yadav R. Mobile image vision and image processing reliability design for fault-free tolerance in traffic jam. *J Adv Scholarly Res Allied Educ (JASRAE)*. 2018;15(6):606-611. <https://doi.org/10.29070/JASRAE>
  6. Qin Z, Luo Z, Wang H. Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. *Int J Image Process*; c2017.
  7. D'monte S, Varma A, Mhatre R, Vanmali R, Sharma Y, Joshi C, *et al*. Department Name of Organization: Information Technology Name of Organization: Kc College of Engineering Management Studies and Research City: Kopari Thane (East) Country: India.
  8. Pantofaru C, Hebert M. A comparison of image segmentation algorithms. *Tech. Rep. CMU-RI-TR-05-40, CMU*; c2005.
  9. Sharon E, Galun M, Sharon D, Basri R, Brandt A. Hierarchy and adaptivity in segmenting visual scenes. *Nature*. 2006;442(7104):810-813.
  10. Sun F, He J. The remote-sensing image segmentation using textons in the normalized cuts framework. *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation. Changchun, China*; c2009 Aug. p. 9-12.
  11. Yang AY, Wright J, Ma Y, Sastry S. Unsupervised segmentation of natural images via lossy data compression. *CVIU*; c2007.
  12. Sun F, He J. A normalized cuts based image segmentation method. *Second International Conference on Information and Computing Science*; c2009.
  13. Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell*. 2000;22(8):888-905.
  14. Malik J, Belongie S, Leung T. Contour and texture analysis for image segmentation. *Int J Comput Vis*. 2001;43(1):7-27.
  15. Julesz B. Textons, the elements of texture perception, and their interactions. *Nature*. 1981;290(5802):91-97.