

# CARTOONIZATION OF IMAGES AND VIDEOS USING GAN AND OPENCV

---

<sup>1</sup>MR.P.V.RAM GOPAL RAO, <sup>2</sup>K.SAMHITHA, <sup>3</sup>M.SAI SHREYA, <sup>4</sup>M.ESHWARSAGAR

<sup>1</sup>(Assistant Professor) ,CSE. Teegala Krishna Reddy Engineering College Hyderabad

<sup>2,3,4</sup>B,tech scholar ,CSE. Teegala Krishna Reddy Engineering College Hyderabad

## ABSTRACT

Cartoonization of images and videos could be used in various different applications, which can be used in publishing a comic book for a comic, anime, T.V. shows as well as for fun events on social media. This paper proposes cartoonization of images and videos through Generative Adversarial Networks (GANs). Thus an idea to convert real world images and videos into cartoonized one is proposed through this paper. With cartoonization, the paper also proposes to make a complete Image-hub for the user with features including up scaling, denoising and editing filters to the input images through the Python OpenCV library. The project also includes video to GIF conversion to use in various social media platforms to achieve cartoon filters. Thus the project is built to be userfriendly and leveraging various other features rather than only cartoonization

## 1.INTRODUCTION

### 1.1 PROBLEM STATEMENT

Title: Cartoonization of Images and Videos using GAN and OpenCV  
Problem Statement: Develop a system for transforming images and videos into cartoon-like representations using Generative Adversarial Networks (GANs) and the OpenCV library. The primary challenge is to balance between preserving the essential content of the input while incorporating stylistic elements characteristic of cartoons. The system should be adaptable to various input types, computationally efficient for real-time processing, and feature a user-friendly interface for controlling cartoonization parameters. Additionally, it should generalize well to diverse content and lighting conditions, with quality evaluation metrics to ensure satisfactory output.

1.2 DESCRIPTION Cartoon is an image or series of images that are formed using a

sequence of illustrations for animations. These cartoons may represent realistic or non-realistic features. However cartoons have gained a huge attention especially by the children, teenagers and artists. Due to which there exists many applications where cartoons are used. Some of these applications include cartoon television shows, comic magazines, cartoon based image filters and animated films. Some of the applications may also contain some real-world scenes. For example, an animated film may contain an image having a city drawn which corresponds to a city that is present in a real world. These cartoon images are created by a skillful artist manually drawing those scenes or by using computer software's to create a single image. To obtain a better quality, the artists need to draw lines and must shade each color region based on real-life scenes. This entire process requires a lot of labor skills and is really time consuming especially while working on animated comics or films. Also the existing computer software's like Corel Draw or Adobe Photoshop are not free to use and also may not be easy for the beginners to understand and achieve the required quality. So there is a requirement of technology that can help transform a real-world based photo or video into an animated image or video

respectively. This technology when integrated with other software's can help the user to convert their real-world 10 photos or videos into cartoon versions as and when required or can also act as an image filter which is also freely available and easy to use. In this paper, we propose a Generative Adversarial Networks (GANs) based approach along with features like image denoising and image up scaling to convert an image, GIF or video files into their cartoon versions. The image upscaling and denoising is achieved using OpenCV. To train the model, data used are a set of photos and a set of cartoon images. The trained model helps in generating the cartoon images or videos that are not a part of training data



image no 1: cartoon image

## **2.LITERATURE SURVEY**

"Transforming photos to comics using convolutional neural networks" This is inspired by Gatys's recent work, we propose

a novel approach that transforms photostocomics using deep convolutional neural networks (CNNs). While Gatys's method that uses a pretrained VGG network generally works well for transferring artistic styles such as painting from a style image to a content image, for more minimalist styles such as comics, the method often fails to produce satisfactory results. To address this, we further introduce a dedicated comic styleCNN, which is trained for classifying comic images and photos. This new network is effective in capturing various comic styles and thus helps to produce better comic stylization results. Even with a grayscale style image, Gatys's method can still produce colored output, which is not desirable for comics. We develop a modified optimization framework such that a grayscale image is guaranteed to be synthesized.

To avoid converging to poor local minima, we further initialize the output image using grayscale version of the content image. Various examples show that our method synthesizes better comic images than the state-of-the-art method. "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization", We propose a solution to transforming photos of real-world scenes into cartoon style images, which is valuable

and challenging in computer vision and computer graphics. Our solution belongs to learning based methods, which have recently become popular to stylize images in artistic forms such as painting. However, existing methods do not produce satisfactory results for cartoonization, due to the fact that cartoon styles have unique characteristics with high level simplification and abstraction, and cartoon images tend to have clear edges, smooth color shading and relatively simple textures, which exhibit significant challenges for texture-descriptor-based loss functions used in existing methods.

We propose CartoonGAN, a generative adversarial network (GAN) framework for cartoon stylization. Our method takes unpaired photos and cartoon images for training, which is easy to use. Two novel losses suitable for cartoonization are proposed: a semantic content loss, which is formulated as a sparse regularization in the high-level feature maps of the VGG network to cope with substantial style variation between photos and cartoons, and an edge-promoting adversarial loss for preserving clear edges. We further introduce an initialization phase, to improve the convergence of the network to the target manifold. Our method is also much more efficient to train than existing methods.

Experimental results show that our method is able to generate high-quality cartoon images from real-world photos (i.e., following specific artists' styles and with clear edges and smooth shading) and outperforms state-of-the-art methods. 12

“Generative adversarial nets,” We propose a new framework for estimating generative models via adversarial nets, in which we simultaneously train two models: a generative model  $G$  that captures the data distribution, and a discriminative model  $D$  that estimates the probability that a sample came from the training data rather than  $G$ . The training procedure for  $G$  is to maximize the probability of  $D$  making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions  $G$  and  $D$ , a unique solution exists, with  $G$  recovering the training data distribution and  $D$  equal to  $1/2$  everywhere. In the case where  $G$  and  $D$  are defined by multilayer perceptrons, the entire system can be trained with back propagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Transformation of Realistic Images and Videos into Cartoon Images”, Aim of the project is to put forward a solution for transforming snapshots or videos of real-world into animated photos (Cartoon Images) or Video. The earlier method of transformation requires complicated computer graphics and skills. The idea of the paper is based on designated snapshots and videos which are converted to an art form such as painting. Amongst all the techniques usable, the application of a Generative Adversarial Network (GAN) called Cartoon GAN will be used for the styling real-world images that use 2 loss functions namely, content loss and adversarial loss for getting a sharp and clear image. With the help of GAN, it is possible to convert video as well to its cartoonized version and the development of the project shows that our Proposed Idea provides high quality cartooned images and videos. “Enhanced Deep Residual Networks for Single Image Super-Resolution” Recent research on super-resolution has progressed with the development of deep convolutional neural networks (DCNN). In particular, residual learning techniques exhibit improved performance. we develop an enhanced deep super-resolution network (EDSR) with performance exceeding those of current

state-of-the-art SR methods. The significant performance improvement of our model is due to optimization by removing unnecessary modules in conventional residual networks. The performance is further improved by expanding the model size while we stabilize the training procedure. We also propose a new multi-scale deep super-resolution system (MDSR) and training method, which can reconstruct high-resolution images of different upscaling factors in a single 13 model.

The proposed methods show superior performance over the state-of-the-art methods on benchmark datasets and prove its excellence by winning the NTIRE2017 Super-Resolution Challenge. "Accelerating the Super-Resolution Convolutional Neural Network", As a successful deep model applied in image super-resolution (SR), the Super-Resolution Convolutional Neural Network (SRCNN) has demonstrated superior performance to the previous hand-crafted models either in speed and restoration quality. However, the high computational cost still hinders it from practical usage that demands real-time performance (24 fps). We aim at accelerating the current SRCNN, and propose a compact hourglass-shape CNN structure for faster and better SR. We

re-design the SRCNN structure mainly in three aspects. First, we introduce a deconvolution layer at the end of the network, then the mapping is learned directly from the original low-resolution image (without interpolation) to the high-resolution one. Second, we formulate the mapping layer by shrinking the input feature dimension before mapping and expanding back afterwards. Third, we adopt smaller filter sizes but more mapping layers. The proposed model achieves a speed up of more than 40 times with even superior restoration quality. Further, we present the parameter settings that can achieve real-time performance on a generic CPU while still maintaining good performance. A corresponding transfer strategy is also proposed for fast training and testing across different upscaling factors.

### 3. SYSTEM DESIGN

System Design: Cartoonization of Images and Videos using GAN and OpenCV

1. Input Module: - Accept input images and videos from users or external sources. - Validate input data formats and properties. - Preprocess input data to ensure consistency (e.g., resize images, normalize pixel values).

2. GAN-based Cartoonization Model: - Design the architecture of the GAN model for image-to-cartoon translation. - Train the GAN using paired or unpaired datasets of real images and their corresponding cartoonversions. - Experiment with different GAN variants (e.g., Pix2Pix, CycleGAN) to find the most suitable architecture for cartoonization.

3. penCV Integration Module: - Utilize OpenCV for image and video processing tasks required for cartoonization. - Implement edge detection algorithms (e.g., Canny edge detector) to extract outlines. - Apply image filtering techniques (e.g., bilateral filter) to enhance cartoon-like features. - Integrate OpenCV functions for color manipulation and noise reduction.

4. Real-time Processing Module: - Optimize GAN inference and OpenCV operations for real-time or near real-time performance. - Utilize techniques such as batching, GPU acceleration, and parallel processing to improve processing speed. - Monitor system resource usage to ensure efficient performance.

5. User Interface Module: - Develop a user-friendly interface for interacting with the system. - Provide options for users to upload images or select videos for cartoonization. -

Include sliders, dropdowns, or input fields to adjust cartoonization parameters (e.g., intensity, color palette). - Display preview of input and output images/videos to allow users to visualize the effects of 19 cartoonization.

6. Output Generation Module: - Generate cartoonized images and videos based on user inputs and selected parameters. - Ensure the output retains the essential content of the input while incorporating cartoon-like features. - Apply post-processing techniques to enhance the visual quality of the cartoonized output.

7. Quality Assessment Module: - Implement metrics for evaluating the quality of cartoonized outputs. - Consider factors such as perceptual quality, fidelity to cartoon aesthetics, and preservation of content. - Provide feedback to users on the quality of the generated cartoonized images/videos.

## MODULES

### 1. IMAGE DENOISING

### 2. IMAGE UPSCALING

### 3. CARTOONIZATION OF IMAGE USING GAN

### 4. CARTOONIZATION OF VIDEOS USING GAN IMAGE DENOISING

One of the basic challenges within the field of image process and computer vision is removal of unnecessary things in an image file which is generally termed as image noise. Image noise could also be caused by totally different intrinsic (i.e., sensor) and accidental (i.e., environment) conditions. To remove those unnecessary things in an image, image denoising plays a vital role in a very wide selection of applications like image restoration, visual pursuit, image registration, image segmentation, and image classification. The underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of the Image. To obtain this image denoising feature, the OpenCV library contains a function `fast NlMeans Denoising Colored` which input converts image to CIELAB colorspace and then separately denoise L and AB components with given parameter regulating filter strength for luminance component.

### **IMAGE UPSCALING**

When scaling a vector graphic image, the graphic primitives that conjure the image can be scaled 20 victimization geometric transformations, with no loss of image quality. When scaling a form ation graphics image, a different image with an improved

or lower vary of pixels must be generated. at intervals, the case of decreasing the image part varies (scaling down) this usually finally end supduring a plain quality loss. The two techniques which we applied are EDSR (Enhance Deep Super-Residual Network) and FSRCNN (Fast Super Residual Convolutional Neural Network). EDSR takes up to 120 seconds to upscale the image whereas FSRCNN is a faster technique and gives an upscaled image output within 10 seconds. But comparison wise EDSR gives a higher up scaled image than

### **FSRCNN CARTOONIZATION OF IMAGE USING GAN**

The Image will be first Denoised and then it will follow the cartoonization Algorithm. In Cartoonizing Images, the generator network is utilized to map input pictures to the animation complex. Cartoon stylization is created once the model is prepared. The generator starts with a flat level convolution stage followed by two down-convolution squares to spatially pack and encode the pictures. Valuable local signals are separated in this stage for downstream change. Afterward, eight remaining squares with indistinguishable formats are utilized to build the substance and complex element. At last, the output cartoon-style pictures are

reproduced by two up-convolution blocks and it will be up scaled in order to increase the quality of image.

## CARTOONIZATION OF VIDEOS USING GAN

Whenever we give an input video for cartoonization, the video gets converted into frames and each frame is Cartoonized individually. The process of cartoonization for frames is same as that of images. Since the video gets converted into frames, loss of sound is observed. In order to aid this problem, we have come up with a method so that there will be no audio loss. Here we apply the concept of multithreading. The input video will send its copy to the sound loss prevention and control unit so that the sound will be extracted from the video and an audio file will be created. At the same time the process of dividing video into frames will take place. Each frame will be Cartoonized. After the cartoonization of all the frames, the frames will be combined to form the Cartoonized video. At last both the outputs from Sound Loss Prevention Control Unit and Video Cartoonizing unit will go through Sound Merging Algorithm where the audio and video will be merged. The output of this whole process would result in

a Cartoonized video which will have same audio as that of the input video.

## 8. OUTPUT SCREENS

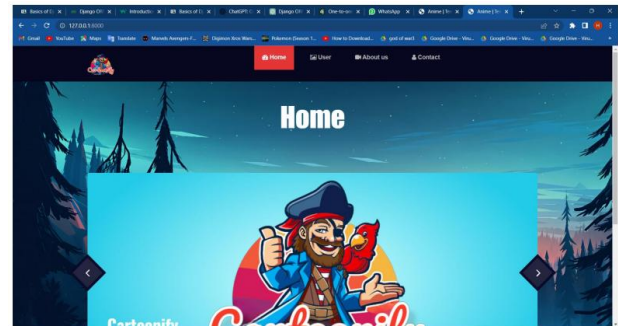


image no 2 home screen



image no 3 user login screen

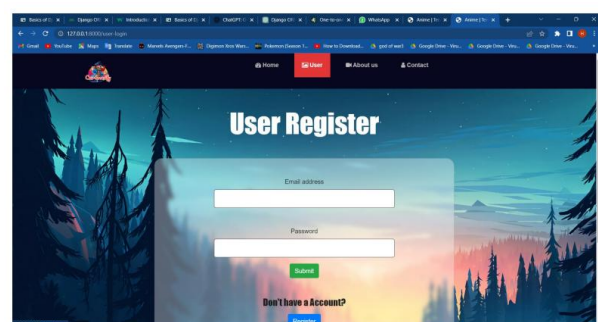


image no 4 user register screen





image no 5 about screen

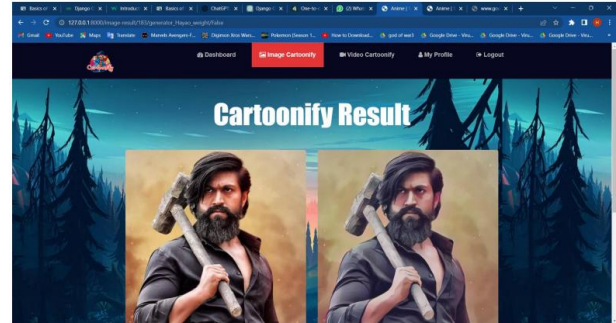


image no 9 result screen

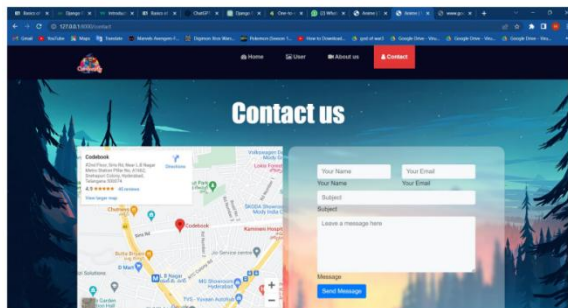


image no 6 contact screen

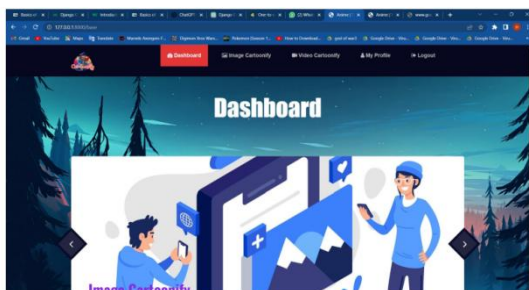


image no 7 dashboard screen

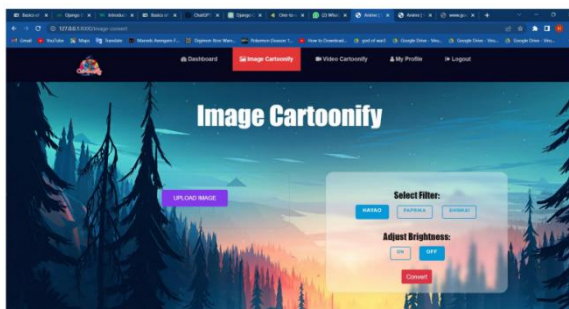


image no 8: image cartoonify

## 9. CONCLUSION

The present methods for converting real-world images or videos into cartoon versions which were proposed by other systems compromise the quality of the image. In the case of video, the audio present in the video file is lost in the resultant cartoon version of the video. We propose a system that helps in cartoonization of images and videos with the help of Generative Adversarial Models(GANs). To implement this, real-world image files are denoised and then passed through the GANmodel which generates the desired cartoonized image. The video is cartoonized by dividing the video into multiple image frames and simultaneously extracting audio from the image file. Each image frame is denoised and passed through the GAN to generate frames of cartoonizedimage. Those cartoonized image frames are aggregated

and converted into video which is aggregated with the audio file to obtain the cartoonized videos. After the implementation of the project, the desired cartoon version of images and videos was achieved. Also the loss of audio has been solved using this approach. Hence, the GAN based Cartoonization model helps in saving time to convert the real-time image/videos into their cartoon version with less noise and better quality.

## 10. FUTURE ENHANCEMENT

Future enhancements for cartoonization of images and videos using GAN and OpenCV can focus on improving various aspects of the system, including quality, efficiency, usability, and functionality. Here are some potential areas for enhancement:

1. Improved GAN Architectures: Explore and develop novel GAN architectures specifically tailored for cartoonization tasks. These architectures can better capture the characteristics of cartoon styles and produce higher-quality cartoonized outputs.
2. Fine-tuning and Transfer Learning: Investigate techniques for fine-tuning pre-trained GAN models on specific cartoon styles or datasets. Additionally, explore transfer learning approaches to leverage knowledge from related tasks and domains.
3. Semantic

Understanding: Incorporate semantic understanding into the cartoonization process to preserve the semantics and context of the input images and videos. This can help ensure that cartoonized outputs remain faithful to the original content.

4. Interactive Cartoonization: Implement interactive features that allow users to guide the cartoonization process in real-time. For example, users could provide feedback or make adjustments during the cartoonization process to achieve desired results.
5. Style Transfer and Fusion: Explore techniques for style transfer and fusion to enable users to apply various cartoon styles or combine multiple styles in the cartoonization process. This can enhance creativity and artistic expression.
6. Multi-Modal Cartoonization: Extend the system to support multi-modal cartoonization, where users can choose from different cartoon styles or even specify custom style preferences. This can cater to a wider range of artistic preferences and applications.

By focusing on these future enhancements, the cartoonization system can evolve to meet the growing demands for high-quality, efficient, and user-friendly cartoonization solutions across various domains and applications.

## REFERENCES

- [1] Y. Chen, Y.-K. Lai, Y.-J. Liu, "Transforming photos to comics using convolutional neural networks", International Conference on Image Processing, 2017.
- [2] Yang Chen, Yu-Kun Lai, Yong-Jin Liu, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization", 2020.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.
- [4] Akanksha Apte, Ashwathy Unnikrishnan, Navjeevan Bomble, Prof. Sachin Gavhane, "Transformation of Realistic Images and Videos into Cartoon Images", 2020
- [5] Bee Lim, Sanghyun Son, Heewon Kim Seungjun, Nah Kyoung Mu Lee, "Enhanced DeepResidual Networks for Single Image Super-Resolution", arXiv:1707.02921v1 [cs.CV], 10 Jul 2017
- [6] Chao Dong, Chen Change Loy, Xiaoou Tang, "Accelerating the Super-Resolution Convolutional Neural Network", arXiv:1608.00367v1 [cs.CV], 1 Aug 2016
- [7] [https://opencv.pythontutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_setup/py\\_intro/py\\_intro.html](https://opencv.pythontutorials.readthedocs.io/en/latest/py_tutorials/py_setup/py_intro/py_intro.html)
- [8] [https://docs.opencv.org/3.4/d1/d79/group\\_ghotodenoise.html#ga03aa4189fc3e31dafd638d90de335617](https://docs.opencv.org/3.4/d1/d79/group_ghotodenoise.html#ga03aa4189fc3e31dafd638d90de335617)
- [9] <https://www.geeksforgeeks.org/generative-adversarial-network-gan/>
- [10] [https://developers.google.com/machine-learning/gan/gan\\_structure](https://developers.google.com/machine-learning/gan/gan_structure).