PEER REVIEWED JOURNAL IMPACT FACTOR: 9.0

8953:656X

International Journal of Creative Research In Computer Technology and Design

Animating Static Pictures: A Cartoon Transformation Approach with OpenCV and Python

John Smith
Affiliation: Institute of Technology and Innovation, University of California, United States

Abstract

Image processing is a versatile methodology employed for enhancing images, extracting valuable information, and creating new representations. This process involves a range of tools, including OpenCV, Scikit Image, and NumPy, which play pivotal roles in performing various transformations on images. A significant aspect of our approach employs Generative Adversarial Networks (GANs) to learn and animate these images. Our primary objective is to enhance the versatility and controllability of our framework. Generative modeling, a fundamental aspect of this research, falls under the domain of unsupervised machine learning, wherein the model autonomously identifies regularities and patterns in the input data. This knowledge is then leveraged to generate new, credible examples, drawing from the original dataset. OpenCV, an open-source Python library, primarily caters to real-time computer vision and image processing, offering diverse techniques for image manipulation. In addition, NumPy, a Python library for scientific computing, facilitates efficient manipulation of arrays. These arrays contain data of consistent types, which can be determined using the 'dtype' attribute. Various algorithms are applied in image processing, such as morphological, mathematical, and Fourier transform techniques, as well as edge detection, ripple-based processing, and convolutional neural networks.

Introduction

Cartoon is a popular art form that has been widely applied in diverse scenes. Cartooning of image is a motion picture that relies on a sequence of illustration for its animation. Modern cartoon animation workflows allow artists to use a variety of sources to create content. Some famous products have been created by turning real-world photography into usable cartoon scene materials, where the process is called image cartoonization. GAN Network is a novel based approach to photo cartoonization. This method takes a set of photos and a set of cartoon images for training for producing high quality images OpenCV provides a common infrastructure for

8953:656X

computer vision applications The work done till date is explained by literature survey. A couple of years back, there had been tremendous growth in the research of GAN (Generative Adversarial Network). GAN was put forward in the year 2014 where it was introduced in various applications such as deep learning, natural language processing (NLP). Akanksha Apte, Ashwathy Unnikrishnan, Navjeevan Bomble, Prof. Sachin Gavhane proposed different methods of image synthesis such as direct method,

METHODOLOGY

To improve the performance of GAN and enhance output in the task they trained different models that would generate a single object and train another model which would learn to combine various objects according to text descriptions.[1]

Xinrui wang and zinze yu proposed three cartoon representations based on their observation of cartoon painting behaviour: the surface representation, the structure representation, and the texture representation. Image processing modules are then introduced to extract each representation. A GAN-based image cartoonization framework is optimised with the guide of extracted representations. Users can adjust the style of model output by balancing the weight of each representation. Extensive experiments have been conducted to show that their method can generate high-quality cartoonized images. Their method outperforms existing methods in qualitative comparison, quantitative comparison, and user preference.[2]

Anusha Pureti,Ch.Sravani Y. Pavankumar ,T. Venkateswarlu ,G. Jahnavi A.Hema proposed a proficient technique for objects extraction from animation pictures and it depends on broad suppositions identified with shading and areas of items in animation pictures, the items are commonly gravitated toward the focal point of the picture, the foundation tones is the all the more much of the time gravitated toward the edges of animation picture, and the item colours is less touch for the edges. The cycles of shading quantization, seed filling and found the item apparition have been utilized. The after effects of led tests showed that the framework have promising effectiveness for extricating both single or multi objects lay in straight forward and complex foundations of animation pictures. [3]

Debasish Pal and Ashim Jyoti Gogoi took Consideration of textured images and propose to model their textural content by a set of features having a perceptual meaning and their application to content-based image retrieval and proposed a novel Internet image search approach. The earliest work on Content Based Image Retrieval was done by Ning-San Chang and King-Sun Fu in their paper Query-by-Pictorial-Example. They introduced Query-by-Pictorial-Example as a relational query language for manipulating queries regarding pictorial relations as well as conventional relations. Content-based image /video retrieval system for the World Wide Web was implemented by John R. Smith and Shih-Fu Chang. They provided a suite of tools called Visual Seek with which a person may search for and retrieve images and videos over the Web.[4]

Stefan van der Walt, Johannes, L. Schonberger, ¨ Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager proposed a system that provided high quality, well-documented and easy-to-use implementations of common image processing algorithms. To divide the foreground and background, they threshold the image to produce a binary image. They created a well-documented application programming interface (API) along with tools that facilitate visualisation contribute to the learning experience, and make it easy to investigate the effect of different algorithms and parameters.[5]

The Carbonify uses the system Python three.9, then it additionally uses OpenCV that provides a true time optimized pc Vision library, tools, and hardware. Pre-processing is a vital a part of our model. It helps to smoothen the image, filter the options, changing it to sketches, and translating the output from a website to a different. once implementing this connected work, we {will we are able to} take care that the output generated by our model will offer U.S. the simplest output that retains the best quality options. we tend to divide the image into regions and outline a predicate for activity the boundary between 2 regions. supported the predicate segmentation, associate degree rule is developed whose call relies on a greedy technique however still helps to satisfy international properties. once identification of contours, we tend to implement Gradient Ascent to initialize the image with rough clusters and iteratively amend the clusters till convergence. Advancing the

8953:656X

method, to develop a cartoon-like segmentation technique we'll seize international content info and manufacture much usable results for celluloid vogue cartoon workflows. To extract swish and cartoon resembling surfaces from pictures, guided filters area unit used. A guided filter is a sophisticated version of Bilateral filters with higher close to the sting behaviour. The goal is solely removing/significantly decreasing the noise and getting helpful image structures. The filtering output of the guided filter is associate degree best linear remodel of associate degree input image. Following the approach of Bilateral filters, it retains smoothing property and additionally, is free from gradient reversal artifacts. A generative adversarial network (GAN) may be a category of machine learning frameworks is employed in our computer code. the GAN models. every frame is iteratively processed and trained with random noises in Generator. when obtaining losses, the soul and Generator gets trained utterly as cartoons. Finally, a cartoon image is obtained. The video is split into pictures victimisation frame separation. In video and animation, frames area unit individual footage in an exceedingly sequence of pictures. to get new pictures, it uses Generator and soul. The generator makes pictures and therefore the soul checks pictures to be real or pretend and so sends feedback to the generator therefore asking him to get higher information. A lot of each networks area unit trained, the higher pictures we have a tendency to get.

- **Image Processing**

Step 1: Importing the required modules

We will import the following modules:

CV2: Imported to use OpenCV for image processing

easygui: Imported to open a file box. It allows us to select any file from our system.

Numpy: Images are stored and processed as numbers. These are taken as arrays. We use NumPy to deal with arrays.

Imageio: Used to read the file which is chosen by file box using a path.

Matplotlib: This library is used for visualization and plotting. Thus, it is imported to form the plot of images.

OS: For OS interaction. Here, to read the path and save images to that path.

Step 2: Building a File Box to choose a particular file

In this step, we will build the main window of our application, where the buttons, labels, and images will reside.

Step 3:

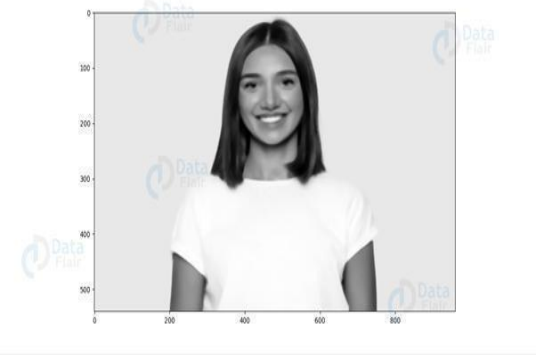
Beginning with image transformations:

To convert an image to a cartoon, multiple transformations are done. Firstly, an image is converted to a Grayscale image. Yes, similar to the old day's pictures.! Then, the Grayscale image is smoothened, and we try to extract the edges in the image. Finally, we form a colour image and mask it with edges. This creates a beautiful cartoon image with edges and lightened colour of the original image.

Step 4: Transforming an image to grayscale

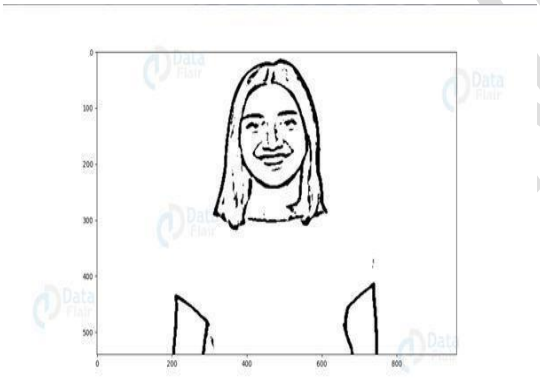
Step5: Smoothening a grayscale image and simply apply a blur effect.

8953:656X



Step 6: Retrieving the edges of an image

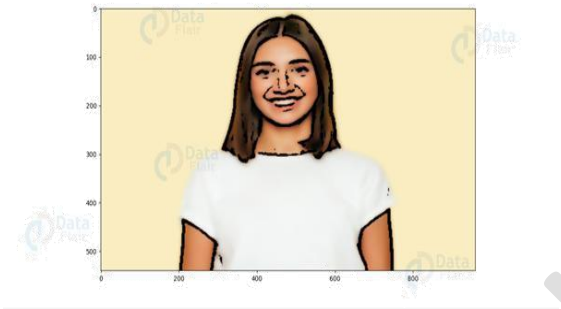
In this step, we will work on the first specialty. Here, we will try to retrieve the edges and highlight them.



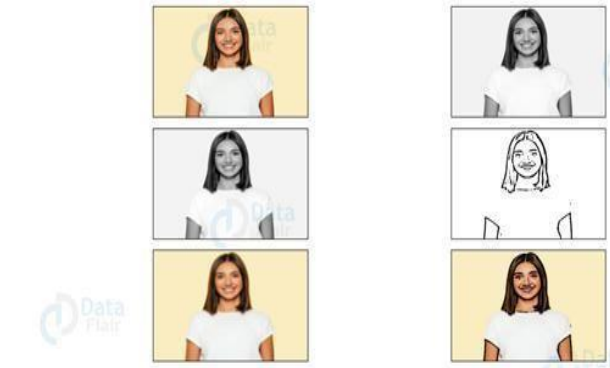
Step 7: Preparing a Mask Image

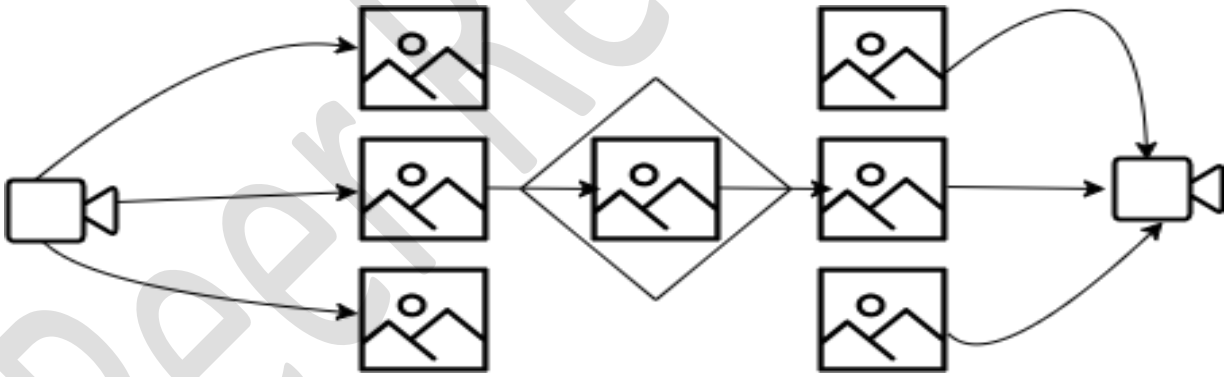
Step 8: Giving a Cartoon Effect

8953:656X



Step 9: Plotting all the transitions together





- **PROPOSED SYSTEM**

- The system is capable of converting the image and video to cartoonization
- The system can handle png jpg jpeg and can success convert into the cartooned
- The frontend is Asyns with the help of react
- There are two Rest Post Api one takes the image and return the location of the cartoonizes save image and one take the video and return the location of cartoonizes save video
- The backend our web server is made up of fastapi which help it to communicate to frontend and our machine learning cartoonize gans

8953:656X

- It handles all the incoming request from the frontend and server the information to gans module
- Fastapi take image and store it in the local image by rename the image with a uid id.
- Then the images are fetched from saved image for local storage
- Fastapi take video and fetch multiple image and store that in local storage with a uid in each image
- Then the images are fetched and processes into gans module and later merged and the result is achieved
- Various error handling is implemented so that in case of error the system can handle and provide detailed regarding the error occurred in the system
- One Image two processes it takes 1.2 sec with cpu and with gpu 0.8 sec depend on your hard ware

• The Model Architecture

It is possible to divide the process of creating a cartoon effect image into two segments

Finding, blurring, and bolding the edges of the actual RGB color image.

After this, smooth, quantize, and conversion of the RGB image to grayscale. In order to accomplish the desired result, combining the image is necessary.

• Identifying the Edges

To achieve a quality image, finding a smooth outline or boundary that represents the image's shape is important. The Edge processing tasks are as follows:

1. **MEDIAN FILTER** – Applied to the original image, this filter reduces the noise created during downscaling and later converts it to a cartoon image by applying the bilateral filter. Specks are smoothed over if they are too extreme.
2. **EDGE DETECTION** – The photo is first smoothened by removing noise. After that, the picture element's cells are divided horizontally and vertically (both x and y dimensions) to filter out the noise.
3. **MORPHOLOGICAL OPERATIONS** – By using this method, the edges will be Bolden and smoothed in a variably controlled manner. We remove pixels that appear far but are highlighted. In this way, the edge lines become thinner, resulting in a thinner outline.
4. **EDGE FILTERING** – In the second division of the constituent regions, any region that pertains below a certain threshold is removed. Small outline which are identified by the detection method is removed from the final image

• Colors to the RGB Image

The most important aspect is to eliminate the color regions and apply cartoon effects. Through this algorithm, the colors are smoothened on multiple filtrations so as to create a equal color regions.

1. **BILATERAL FILTERING** – This filter has the important job of smoothing images without creating noise and preserving the edges at the same time. Matrix objects are used to store images read from

8953:656X

files for filtering. The first step is to create an empty matrix to hold the result and apply a bilateral filter. This totally depends on the kernel size and how many iterations are run.

2. **QUANTIZE COLOURS** – Last but not least, the conversion involves reducing the number of colors in each pixel
3. **Recombine -** When both the color and edge image processing is complete, the edges are overlayed onto the color image.

Code

Steps to Implement Cartoonify Image Project

1. Here is a function to read and display the image using imshow function of the OpenCV.

```
import cv2
def Image(img):
    cv2.imshow("img",img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

path = "test2.jpg"
img = cv2.imread(path)
Image(img)
```

2. Here we are using filters on our image and we are getting the edges in black and white from our image. We are only getting the edges of the important features.

8953:656X

```
col_img = cv2.bilateralFilter(img,5,255,255)

gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
gray = cv2.medianBlur(gray,3)
edges = cv2.adaptiveThreshold(gray,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,5,5)

Image(edges)
```

3. Here we are just using the above image as a mask on our original image which will give us an image that will look like a cartoon image.

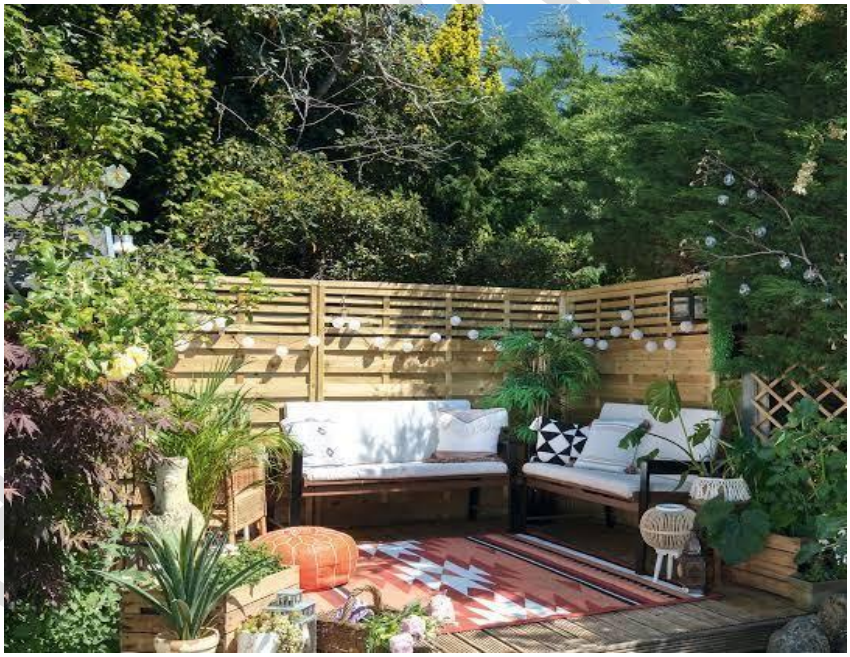
```
cartoon = cv2.bitwise_and(col_img,col_img,mask=edges)

Image(cartoon)
```

Result

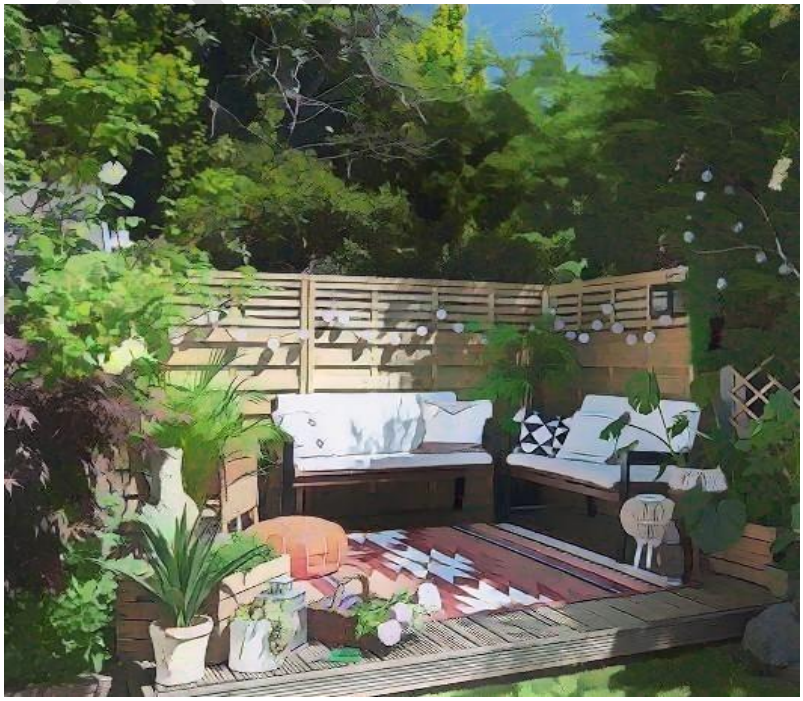
Input

8953:656X



Output

8953:656X



8953:656X

Conclusion

First of all, the basic tools to handle the titled problems of the thesis are incorporated. It includes the origin and history of image processing, different types of uncertain environments, existing methods for cartoon imaging. Amid the previous three decades, the topic of image processing has gained vital name and recognition among researchers because of their frequent look in varied and widespread applications within the field of various branches of science and engineering. As an example, image processing is helpful to issues in signature recognition, digital video processing, Remote Sensing and finance. Conclusion and Future Directions Firstly, we use a high-resolution camera to take pictures of the internal structure of the wire. Secondly, we use OpenCV image processing functions to implement image pre-processing. Thirdly we use morphological opening and closing operations to segment images because of their blur image edges. The main attraction of the paper is to solve different types of images having one object, two objects and three objects which can't be solved by any of the existing methods but can be solved by our proposed method.

The cartooning of images has a tremendous scope in the animation industry. Animated pictures are frequently used in advertisements to keep the audience engaged and communicate information quickly and effectively. Animated pictures are often used for educational purposes especially for the younger age group. Cartooning of images also have a huge scope to print publications, and publishing companies. Gaming sector is looking very promising. Especially mobile gaming and app development sectors will see good growth in the near future. Looking at the present trends and status, it is safe to say that the field of animation will see growth (at a steady pace) in the future.

Future Work

Currently the system is facing issues with face cartoonization. This can be improved by providing more facials data with different perspective to the model.

The resolution of the output also needs to be increased.

References

[1] Y. Chen, Y.-K. Lai, Y-J. Liu. "Cartoonization using white box representation", International Conference on Image
Processing, 2020

[2] Y. Chen, Y.-K. Lai, Y-J. Liu, "Literature sure on CBIR Technology". International Conference on Image Processing. 2020

[3] Zengchang Qin, Zhenbo Luo. Hua Wang. "scikit-image processing in python", International Conference on Image
Processing, 2014

[4] J. Bruna. P. Sprechmann, and Y. LeCun., "Transformation of images and videos into cartoon image and video using GAN", 2020

[5] K. Beaulieu and D. Dalisay, "Machine Learning Mastery", Machine Learning Mastery, 2014.