

Cartoonization of images using machine Learning

Silviya D'monte¹, Aakash Varma², Ricky Mhatre³, Rahul Vanmali⁴, Yukta sharma⁵

¹Assistant Professor, Universal college of Engineering

^{2,3,4,5} Students, Universal college of Engineering

Abstract

Image process could be a methodology to perform some operations on a picture, so as to induce Associate in Nursing increased image orto extract some helpful data from it. it's a kind of signal process within which input is a picture and output might bean image or characteristics/features related to that image. Image process tools include: OpenCv, Scikit Image, Numpy. A Generative Adversarial Network (GAN) is essentially wont to learn the extracted representations and any animate the photographs. the most object of our methodology is to create our framework additional governable and adjusting. Generative modelling is Associate in Nursing unattended learning task in machine learning that involves mechanically discovering and learning the regularities or patterns in input file in such how that the model may be wont to generate or output new examples that credibly might are drawn from the first dataset. OpenCV is Associate in Nursing ASCII text file python library used for pc vision and machine learning. it's principally geared toward time period pc vision and image process. it's wont to perform diferent operations on pictures that rework them victimisation diferent techniques. Numpy could be a library for scientific computing in Python. It provides a superior flat array object and tools for operating with these arrays. A NumPy array is analogous to an inventory. we are able to solid an inventory to a NumPy array by frst importation it. Numpy arrays contain information of a similar type; we are able to use the attribute "dtype" to get the info style of the array's components. The algorithms utilized in image process area unit morphological Image process, mathematician Image process, Fourier rework in image process, Edge Detection in image process, rippling Image process, Convolutional Neural

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 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 diferent methods of image synthesis such as direct method,

LITERATURE SURVEY

To improve the performance of GAN and enhance output in the task they trained diferent 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. Jahnvi 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 an 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 Cartoonify 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 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 model. 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. The a lot of each networks area unit trained, the higher pictures we have a tendency to get.

MEDIAN FILTER – This filter helps in reducing the noise created throughout the downscaling of the image and later changing the initial image to cartoon image by applying the bilateral filter. Any extreme specks area unit ironed over.

EDGE DETECTION – initially the noise of the image is removed inside the image. Later the smoothed image is filtered victimisation horizontal and vertical direction by dividing the cells of the image part.

MORPHOLOGICAL OPERATIONS – This serves the aim to Bolden and smoothen the define of the perimeters

variably. The pixels that area unit highlighted however appear way area unit removed. therefore the sting lines cut back to a agent define.

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.

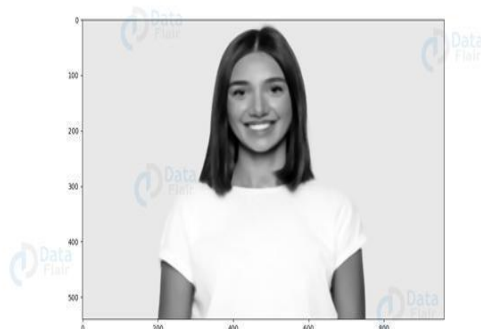
Step 3:

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 color image and mask it with edges. This creates a beautiful cartoon image with edges and lightened color of the original image.

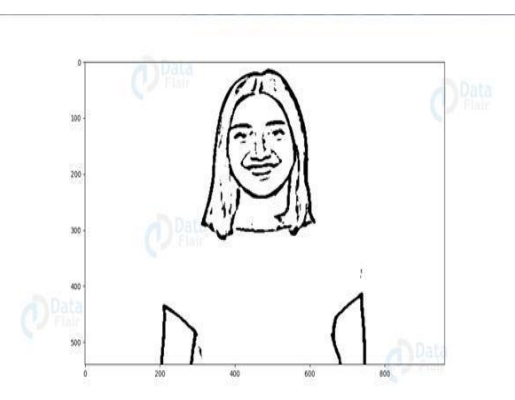
Step 4: Transforming an image to grayscale

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



Step 6: Retrieving the edges of an image

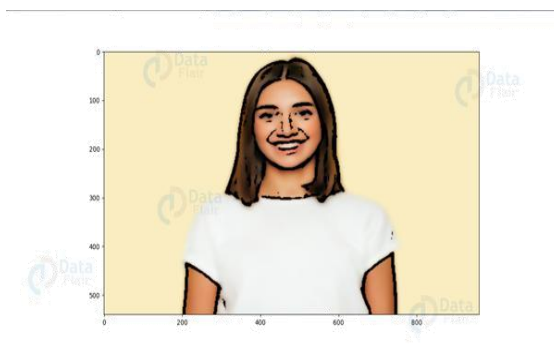
In this step, we will work on the first specialty. Here, we will try to retrieve the edges and highlight them.



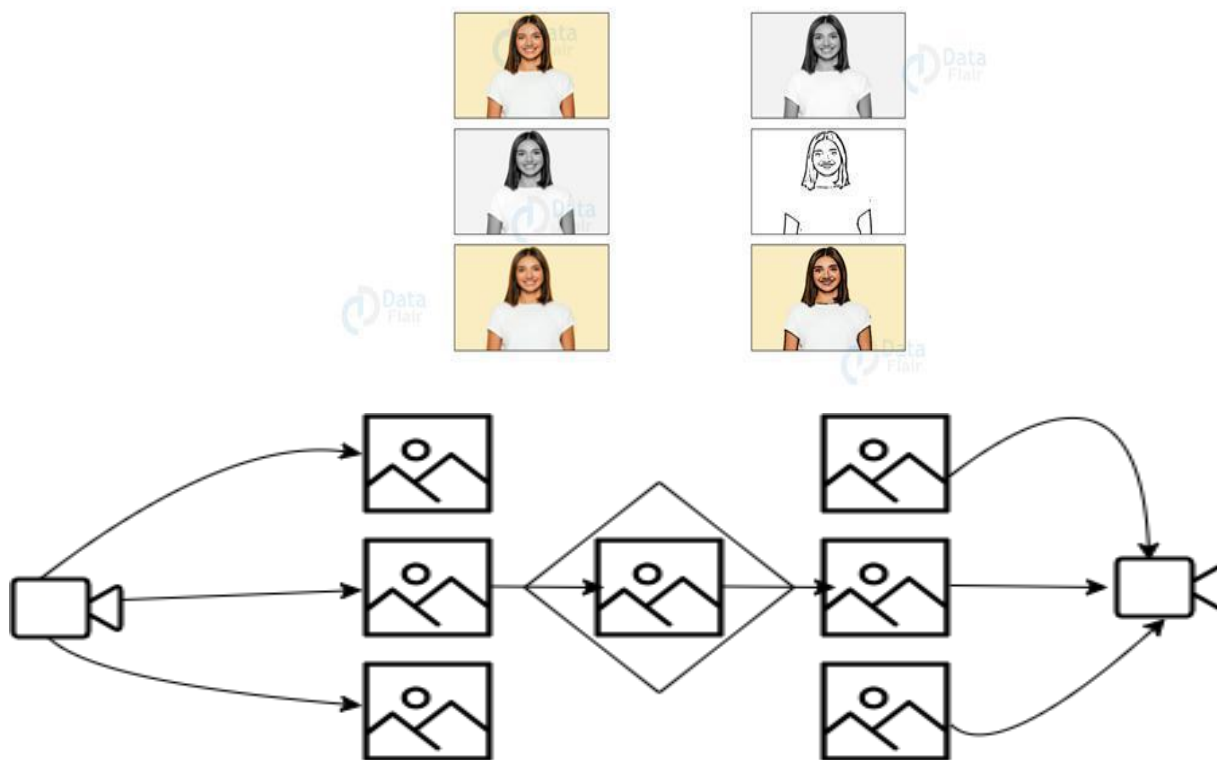
Step 7: Preparing a Mask Image



Step 8: Giving a Cartoon Effect



Step 9: Plotting all the transitions together



Video Processing

Step 1:- The video is first received by the fastapi web server which help to communicate between frontend and backend let the frontend communicate with the machine learning gans module which convert the image and video to cartoonization

Step 2:- when the video is received in the backend then it being . Processed by the opencv to load up into memory

Code

```
vidcap = cv2.VideoCapture(saveFile)
```

Step 3:- Then the video is break into several frame each frame is recorded every second and saved as an image . You can also tunned the frame recorder parameter to adjust the frame

Code

```
sec = 0
```

```
    frameRate = 0.1
```

```
    success = getFrame(vidcap,filename,sec)
```

```
    while success:
```

```
        sec = sec + frameRate
```

```
        sec = round(sec, 2)
```

```
        success = getFrame(vidcap,filename,sec)
```

```
vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
```

```
hasFrames,image = vidcap.read()
```

```
if hasFrames:
```

```
    img = Image.fromarray(image)
```

```
    img.save(f"{FILEPATH}{str(sec)}.jpg")
```

```
    global tempframes
```

```
    tempframes.append(f"{str(sec)}.jpg")
```

```
return hasFrames
```

Step 4:- Then each image is put into the machine learning module gans cartoonization and then processed and saved

Step 5:- Then the image are merged into one with the help of movies.py library . which take all the image and merged them into a video and return it to the end user

Code

```
img_array = []
```

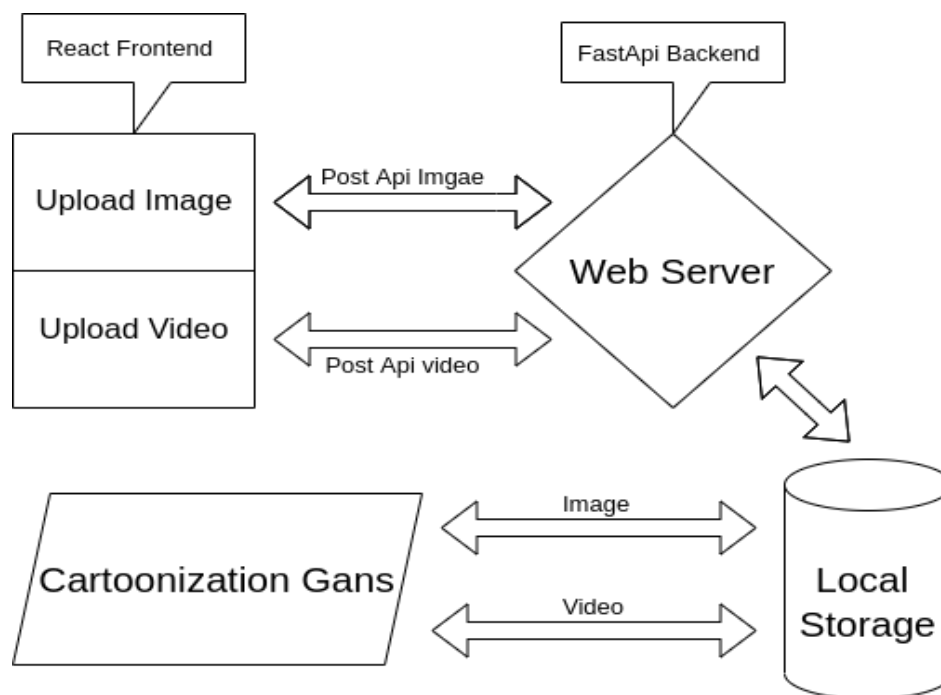
```
for filename in tempframes:
```

```
    image = ImageClip(VIDEOFRAEMEPATH+filename).set_duration(0.1)
```

```
    img_array.append(image)
```

```
video = concatenate_videoclips(img_array, method="compose")
```

```
video.write_videofile(path,fps=
```

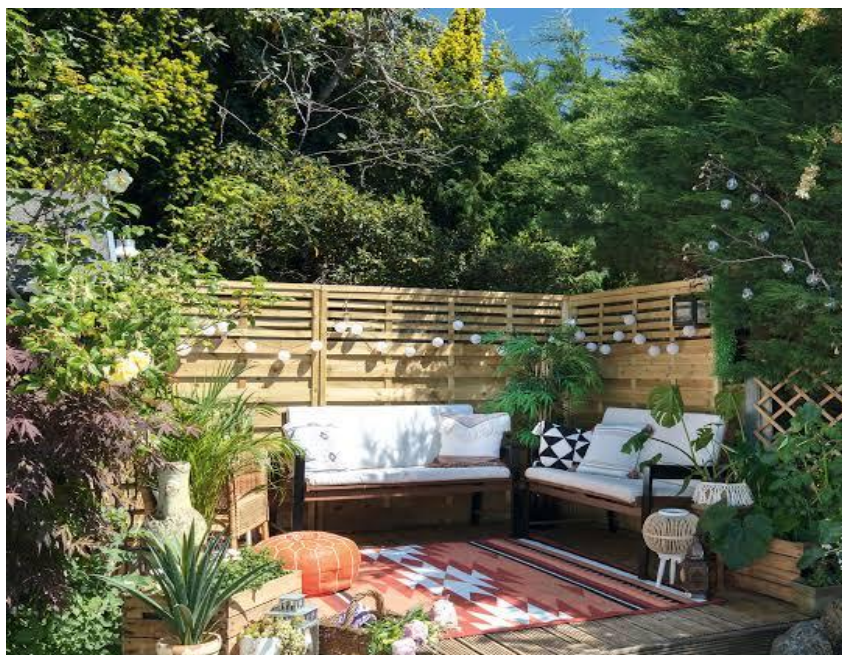


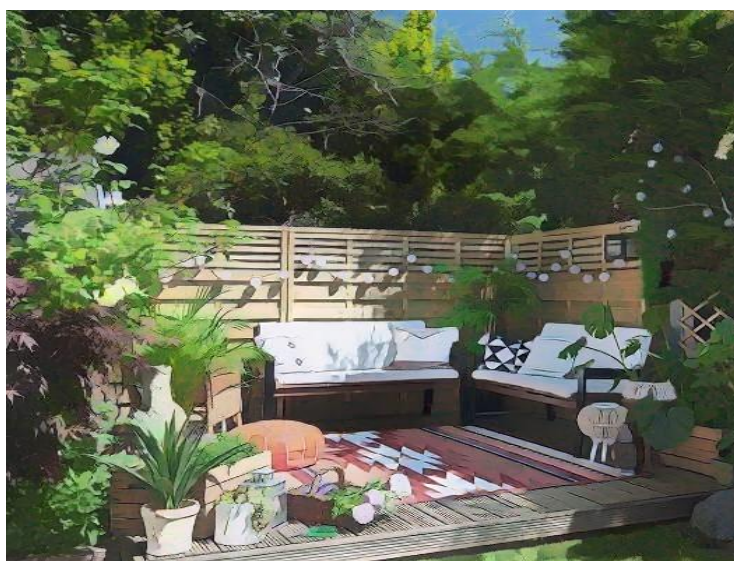
PROPOSED SYSTEM

- The system is capable of converting the image and video to cartoonization
- The system can handle png jpg jpeg mp4 and can success convert into the cartooned
- The frontend is Asyns with the help of react
- There are two Rest Post Api one take 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
- It handle 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 fetch 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 fetch 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 regrading the error occurred in the system
- One Image two processes it take 1.2 sec with cpu and with gpu 0.8 sec depend on your hard ware

Result

Input → output





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 need 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.