CS 432: GPU Accelerated Computing
Project Research Paper : Cartoonify Images

By Alisha Momin and Ayeza Nasir

Dhanani School of Science and Engineering, Habib University , PK

Abstract The aim of the project is to propose a method for converting real-world images into cartoon images. The notion of the paper revolves around snapshots that are converted into an artistic medium such as painting. This paper demonstrates various methods for transforming an image into a cartoon. You can transform any type of image into a cartoon, including images of people, hills, plants, animals and so on. The previous method of conversion an image to cartoon uses complex computer graphics and expertise such as Photo-shop, Adobe Illustrator, Windows, MAC etc. We aim to provide an algorithm which is time efficient and have high performance.

Keywords · Computer Vision · Image Processing · CV2 · Cartoon Effects ·

Introduction Cartoons are depictions of fictional or fictitious characters. They are semi-realistic or non-realistic, and are frequently used to describe a scenario or event in a humorous, sarcastic manner.

One of the first traditional animated films, Fantasmagorie (1908), where all the film images were drawn by hand. Until then, the animation tradition was all about hand-drawn frames. Previously, cartoonists drew such cartoons by hands, and as Anime gained popularity, it became more challenging for them to create each movie frame by hand, that is time consuming and was not fixable if an error occurred.

With the advancement of technology, various software systems were designed to digitally create the visuals and illustration which minimized the human involvement and becoming more efficient than hand-drawn illustration. Which was also enhanced over time with the addition of advanced tools. Toy Story (1995), the world's first fully computer-animated movie, was a phenomenal success. It depicted the characters as extremely interactive, and the outstanding animation brought them to life.

Nowadays, automation technology is expanding rapidly. Human effort is minimized; with a collection of programs performing the tasks for us, animation technology has advanced significantly. Thus, this is the project that converts regular camera-captured images into cartoons, using the C++ language and associated libraries. Nowadays, when automation takes huge steps with a great minds enhancing them daily, the animation industry's primary issue is quality and other ultra-high-definition films gaining popularity. Due to the fact that their cameras are unique, animating these films while maintaining a high level of quality is a monumental task. Toy Story 4, The Garden of Words, stands out as one of the most precise animated films of all time. Anime films take significantly longer to make than conventional films, especially in the age of visual effects and graphics. Classic movies require more in terms of personnel, expertise, and equipment and investment, so the amount of time spent on production is critical. AI and machine learning have made every systems more efficient, which will enhance the quality of the work.

In addition to being a fun pastime, manipulating images has recently become a popular career choice for many people, with several individuals making a full-time living out of it. To learn more about the inner workings of image processing and computer vision, we can turn to the OpenCV library. With the help of a few different filters, a grey image can be transformed into a coloured one. These methods are also used to restore lost details in old pictures.

The best way to turn a photo into a cartoon is to use editing software like Photoshop etc but this isn't something that everyone has access to and it is limited. Even though manual processing yields the best results yet it is time consuming, slow and inefficient when applied to large numbers of images. Therefore, our project goal is to design an algorithm which will convert a large amount of pictures by increasing the performance of GPU by reducing the time and making it efficient.

Objectives:

The objective of this project is to make an application with a basic UI permitting users to apply animation to pictures so that they may produce a cartoony image. The project is intended to give masterful and cleverly engaging outcomes on as wide a scope of pictures as could really be expected, in spite of the fact that not all data sources will yield similarly fulfilling results. The filters are intended to produce artistic output on a wide variety of images. The system is designed with simplicity as the code is developed in C++ programming language, which is simple to grasp yet has a wide range of applications in every industry.

Literature Review Amongst the existing projects that we reviewed , we only came across python projects, This project has not been attempted in C++ before. Amongst the already exiting projects we can conclude the following to be the common areas. Most

projects use the CV2 library , by importing this library they can automatically use predefined functions to perform tasks such as filtering in to median , grey or resizing the image without utilizing the threading methodology. In our project we have not employed any built in functions , we have constructed specific functions to perform these tasks via threading and individual work on each pixel.The algorithm mainly constitutes of 2 phases of processing to create the animation is. The first is distinguishing and boldening the edges, and the second is for smoothing and quantize the colors in the picture. In the end, the subsequent pictures are consolidated to accomplish the final image.Over the past thirty years, the subject of the processing of images has acquired an indispensable name and acknowledgment amongst scientists as a result of their incessant glance at changed and boundless applications inside the field of different parts of science and design.
The following section discusses the course of action opted for in the already existing projects:
Prior models that proposed a similar methodology utilized black-box models, the previous model accomplishes extraordinary correctness yet slumps the stylizing quality causing a few terrible cases

Non-photorealistic rendering (NPR):
A few works done previously render 3D shapes in straightforward shading procedures, which makes an animation like impact. Such methods are called cel-shading which are able to greatly optimize timing, for artists and designers and have been utilized in the making of games as well as animation recordings and films.
An assortment of strategies has been created to make pictures with flat shading, impersonating animation, and cartoonification styles. Such techniques use either filter on images or optimization of other numerical formulae. In any case, it is hard to catch rich imaginative styles utilizing basic numerical equations. Specifically, applying filtering or optimization to the whole picture doesn't give the significant level of abstraction that a craftsman would typically do, like making object edges and boundaries clear and more identifiable. To improve outcomes, alternative techniques depend on the segmentation in images. Strategies have additionally been created for representations, where semantic segmentation can be inferred consequently by distinguishing facial parts. Nonetheless, such techniques can't adapt to general pictures.

L1 Transformation for Image Flattening :
This methodology utilized image smoothing by reducing a quadratic energy function.

Image Smoothing Via End-to-End networks:
This methodology is has been utilised in previous projects to produce and image with less pixels and reduced noise which will contribute to the a single value representing the image, as in an average or median value.

Machine Learning algorithm applied with 2 input images :
Another attempt at this project was made in 2019 by Chinmay Joshi, Devendra Jaiswal, Akshata Patil from KC College [4] , their proposed solution would 2 images with neural style transfer algorithm from machine learning which would take 1 image that needs to be converted a sin input from the user and another image selected by the user that will be the style needed to be applied to the image. In this solution user is provided with a set of pre-trained images to choose from. The weaknesses in this model are that there is no finite number of possible outcomes and the processing requires a lot of GPU hardware. The implementation uses TensorFlow , loss functions and shallower layers than the ones in Johnson's implementation

GAN Model Based Conversion : This approach was also discussed in the IRJET Volume 07 Issue 1 in January 2020.Which details that the GAN algorithm is is a blend of a generative model and a discriminative model. The generative model makes new examples of data that are similar to the data used for training. The discriminator is the model on the other hand is utilized for testing the data and contrasting it and the picture from the generator. The discriminator chooses whether the yielded picture is fake or real. The Generator and Discriminator both are neural network-based and both the algorithms run in competition with one another in the preparation stage.Transforming images (snapshots) to the finest cartoonified images(animated images), with the help of the loss function and its two types named as Adversarial loss and Content Loss, we are able to achieve a flexible as well as a clear edge defined images. This framework experienced challenges of readjusting parameters similar to other deep learning models. Additionally, the network structure for combination brought about less speed whilst training.

Algorithm
In this paper, we proposed an image cartoonification Algorithm that transforms real-world photos into high-quality cartoon-style images.

Following is the detail of the High Level implementation of our program. In the GPU code:

For the conversion of the image multiple transformations need to be done.
Firstly, an image is changed to a Gray scale image.
Then, the Gray scale image is smoothed, the noise is reduced in the image.

To smoothen a picture, we basically apply a blur impact to it. This is finished utilizing median blur.
Here, the middle pixel is allocated a average numerical value of the relative multitude of pixels which fall under the kernel. Thus a blurry visual in achieved 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.
Elaboration on mathematical methodology: **Guassian Filter** Gaussian Filter is a direct approach. Be that as it may, it doesn't secure the edges in the picture - the magnitude of sigma oversees the level of smoothing, and in the end, how the edges are safeguarded. It is utilized to obscure the picture and eliminate commotion and detail.
Representation in 1D:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{x^2}{2\sigma^2})$$

Representation in 2D:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2+y^2}{2\sigma^2})$$

Laplacian Filter

This is used for edge detection it identifies and alters adjacent pixel values and utilises the second derivative for the determination of this value.

Mathematical representation:

$$\nabla^2 f = \nabla \cdot \nabla f$$

Median Filter Utilised for extreme values that are not similar to the median. This can be used to remove salt and pepper noise and correct intensity values by replacing intense pixel values with median values in the central pixel.

Final Observation

The elapsed time in GPU was **0.135538 ms**
The elapsed time in CPU was **0.701592 ms**

Challenges We struggled with learning about adaptive threshold during this project as well as CV2 installation and running on our machines.

Summary Observation industry is one of the most buzzing fields , it is projected to grow bigger and better int he future with the coming of metaverse, NFTs and digital property holdings etc. Hence our project shows a great margin for expansion. Our framework can undoubtedly be fixed up with a mechanization framework and will give expected results where picture are less pixelated and more refined. We observed that the following were the time differences in the

Future Scope The results of our project demonstrate that the image was effectively transformed into a cartoon effect. In future, we'd like to concentrate on face and object identification, followed by generating cartoonified portrait and video. Additionally, we plan to devote more time and effort to video conversion in order to obtain a higher-quality HD or 4K video that will be more efficient.

References

[1] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge : Image Style Transfer Using Convolutional Neural Networks, 2016

Justin Johnson, Alexandre Alahi, Li Fei-Fei : Perceptual Losses for Real-Time Style Transfer and Super-Resolution, 2016

R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A MATLAB-like environment for machine learning. In NIPS Workshop on BigLearn, 2011.

V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. In International Conference on Learning Representations (ICLR), 2017.

2022. [online] Available at: <https://www.researchgate.net/public
$[Accessed 9 May 2022]$.