International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 3, Issue 12, pp 59-62, December-2019

Image Cartoonization Using Data Augmentation in Deep Learning's Methodology

Dr. Anbunathan

Professor, Department of Computer Science and Engineering, Malla Reddy College of Engineering for Women,
Hyderabad -500100, Telangana, India

Abstract - In this paper, we have discussed various methodologies for transforming an actual real-world image into a cartoon-effect image. Despite the pre-existing traditional image-editing softwares and algorithms which provide standard features, they still fail to fabricate satisfactory results for cartoonization. Hence, to provide the painterly effect onto the images, several methods have been applied on the basis of rendering effects and diverse expressive styles generated. The outcome obtained from the analysis can be integrated with various other useful applications like image processing and transformation, object detection etc.

Keywords: Image processing and transformation, object detection, cartoonization.

1. INTRODUCTION

Cartoons are a type of illustration that are typically drawn, are sometimes animated in an unrealistic or semi-realistic style; it is a form of art. With the advancement in times and the consumption of cartoons as a product, developers and researchers around the world have grown the skills of developing algorithms and techniques to create cartoons and overcoming the primitive methods of cartoonification. All of us are aware that creating a cartoon in a hand-drawn way is not the single option now. There are so many mechanisms present with which one can transform real images into cartoonized images. The technologies are being used to indemnify everyone's needs and these advancements are the aid to further development in this domain.

With newer technologies coming into the forefront, the real images can be easily modified into cartoonified ones with simple tools, algorithms and softwares. Cartoonification is quite a gravitating subject and is being applied in various domains like in animated movies, social media, and short films and for countless fun purposes. Since the primitive methods for cartoonifying images are not at par with the current requirements, it was a tedious as well as time-consuming task, in order to overcome these situations there descends the furtherance in algorithms with time. In this paper, we have analyzed and put together some algorithms and techniques along with their advantages and limitations which would help us answer the fact that what led to the progression or evolution of these algorithms [10].

2. PROPOSED SYSTEM

2.1 Bilateral Filtering

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images, another [2][11] fun application of a bilateral filter is to "cartoonize" an image. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges [7][8]. Several trails of bilateral filter that specify its success: • The formulation is simple: each pixel is replaced by the average value of its neighboring pixels. This aspect is important as a result of it makes it simple to accumulate intuition regarding its behavior, to adapt it to application-specific requirements and to implement it. • It depends solely on two parameters that indicate the size and contrast of the property to preserve. • It is utilized in a non-iterative manner. This makes the parameters easy to line since their result isn't accumulative over multiple iterations [9]. Image cartoonization steps using BLF method are: 1) Feature space conversion is performed to extract the contrasts in the given image. Feature space such as CIE Lab so that image contrast is adjusted depending on just noticeable differences. We follow this advice and our parameter values assume that $L \in [0,100]$ and $(a,b) \in [-127,127]$ [5]. 2) Second step in the process is image smoothing which is done by using a filter called bilateral filter to smoothen the input image.

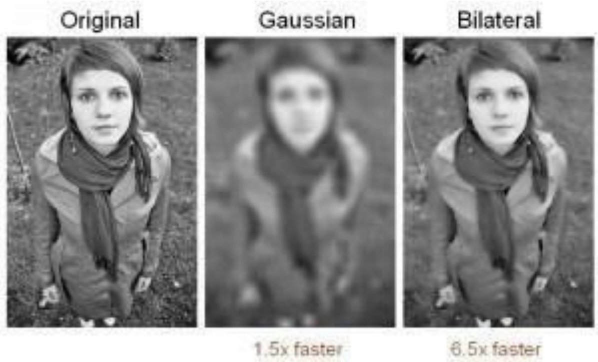

Figure 1: Comparing Gaussian Distribution method with Bilateral Filtering method

Limitations

- The bilateral filter in its direct form can produce many types of image artefacts:
- Staircase effect - Intensity plateaus that cause images to seem like cartoons.
- Gradient reversal - Introduction of false edges within the image
- Its procedural cost is relatively high compared with that of edge-preserving smoothing.

There are several filter extensions that can handle these artefacts. Alternative filters such as guided filters are also suggested as effective alternatives without these limitations [9].

2.2 Linde Buzo Gray Vector Quantization

Vector quantization (VQ) is amongst one of the lossy information compression techniques and has been employed in range of applications like pattern recognition, speech recognition and face detection, Content primarily based Image Retrieval (CBIR) etc. The vector quantization once applied on an input image forms the clusters of colours that cause colour reduction. The output created by vector quantization once mapped with the perimeters extracted from the input image provides a border like impact on the image that makes the image look cartoonized [4].

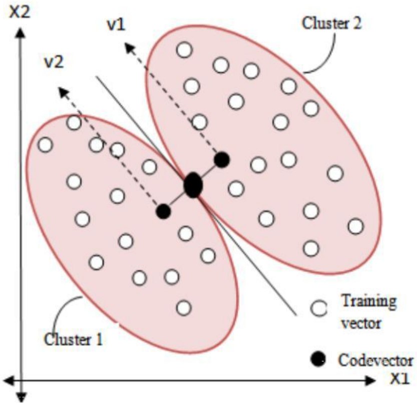

Figure 2: LBG codevectors V1 and V2

2.3 Kekre's Proportionate Error Vector Quantization

The KPE VQ algorithmic rule is somewhat almost alike the previous VQ algorithm. However, rather than adding a constant/continuing error to the centroid, a proportionate error value is added which is decided based on the components of the centroid. For instance, let k be the length of codevector, C = {c1, c2, c3, …, ck} be the code vector generated from the mean of training vectors, and E = {e1, e2, …., ek} be the error vector. After computing cj as min { ci /i = 1,2, ... k} where j is the index of the vector, we then assign ei = 1and if ci /cj <=10 then set ei = ci /cj else, ei = 10 for every i ≠ j and i=1,2,........., k [16]. The error vector is generated every time the new clusters are formed and is added to the codevector to form the new codevector.

The proportionate error added in both positive and negative direction of the centroid in order to get initial two code vectors in codebook. The error ratio ei is decided by the magnitude of coordinates of the centroid. Hence, the procedure is same as that of LBG.

3. RESULT ANALYSIS

Referring to Figure 2.4.1 in GAN architecture, the generator network G uses the mapped input images to the cartoon manifold. Cartoonified image is fabricated once our model is trained. G initiates with a flat convolutional stage followed by two down-convolutional blocks to spatially flatten and encode the photos. Required local signals are obtained in this stage for downstreaming the transformation [19].

Subsequently, these eight residual blocks with identical configuration are used to build the content and the manifold aspect here. Lastly, the output cartoonized images are restructured by two up-convolutional blocks which consist of a fractionally-strided convolutional layer with a stride 1/2 and a final convolutional layer of 7 × 7 kernels [18][19].

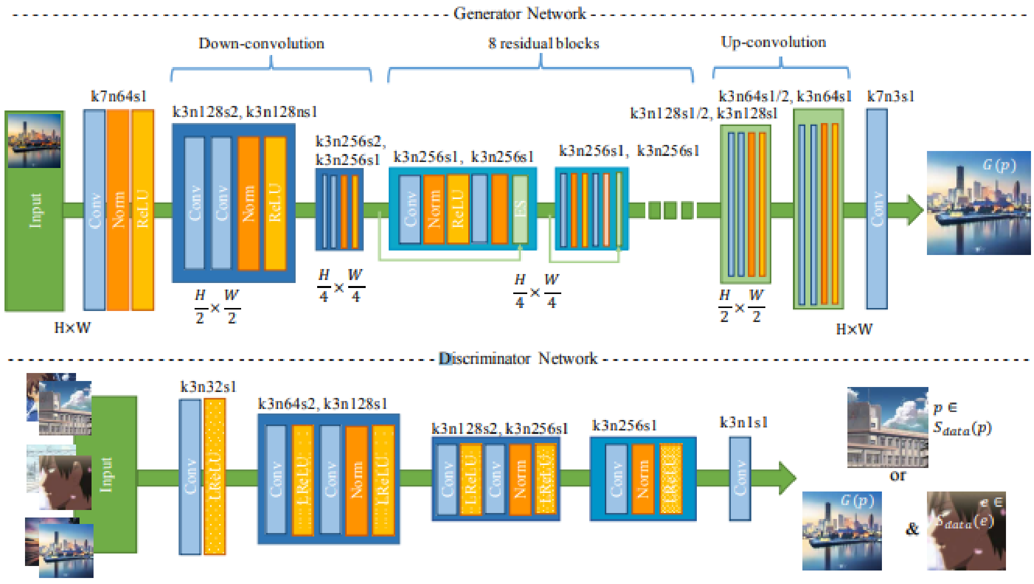

Figure 3: Architecture of the generator and discriminator networks in our reference model

4. CONCLUSIONS

With the advancements in technology and methods related to cartoonification of images we have come a long way. Every algorithm/technique came up with its own improvements and versions to create the best possible cartoonified image.

A technique called data augmentation in deep learning's methodology in the computer vision domain has been one of the greatest achievements of the time. Data augmentation results us in a way better way with performing models, both by increasing model skill and by providing a regularized effect reducing the generalization error. It works by creating latest, artificial but credible examples from the input problem domain on which the model was trained. In compound domains or domains with a limited range of data, generative modeling provides us a way towards more training for modelling the data. GANs have been quite successful in these use cases in domains like deep reinforcement learning.

There are many research reasons why GANs are engaging, vital and require further study. Ian Goodfellow outlines a number of these in his 2016 conference keynote and associated technical report titled "NIPS 2016 Tutorial: Generative Adversarial Networks."

Amongst these reasons, he highlighted GANs' successful ability to model high-dimensional data, handle missing data and the capacity of GANs to provide multi-modal outputs or multiple plausible answers [15]. Possibly the most compelling reasons that GANs [3] are majorly considered, developed and is used because of their success. GANs have been able to spawn images so realistic that humans are unable to tell that they are of objects, scenes and people that do not exist in real lives. Stupefying is not an apt adjective for their capability and triumph [18].

REFERENCES

[1] Cartoonization Using LBG And KPE Vector Quantization Dr. Archana B. Patankar*, Ms. Purnima Tawde.

[2] Patankar, A. B., Kubde, P. A., & Karia, A. (2016). Image cartoonization methods. 2016 International Conference on Computing Communication Control and Automation (ICCUBEA). doi:10.1109/iccubea.2016.7860045.

[3] CartoonGAN: Generative Adversarial Networks for Photo Cartoonization Yang Chen Tsinghua University, China, Yu-Kun Lai Cardiff University, UK, Yong-Jin Liu∗ Tsinghua University, China.

[4] Dr.Archana B. Patankar, Purnima Tawde," Cartoonization using LBG and KPE Vector Quantization", International Journal of Modelling and Simulation, 18(2), 1998, 112-116.

[5] By Sylvain Paris, Pierre Kornprobst, Jack Tumblin and Fr´edo Durand, "Bilateral Filtering: Theory and Applications",Foundations and Trends R in Computer Graphics and Vision Vol. 4, No. 1 (2008) 1–73 c 2009.

[6] T. G. Stockham, "Image processing in the context of a visual model," Proceed-ings of the IEEE, vol. 60, no. 7, pp. 828–842, 1972.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images,"in Proceedings of the IEEE International Conference on Computer Vision,pp. 839–846, 1998.

[8] C. C. Wang, "Bilateral recovering of sharp edges on feature-insensitive sampled meshes," IEEE Transactions on Visualization and Computer Graphics, vol. 12,no. 4, pp. 629–639, 2006.

****