International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Advancement of Techniques for Image Cartoonization

Aditi Lohomi[1], Rashmi Singh[2], Stuti Mittra[3]

[1-3]Dept. of Computer Science and Engineering, Rajasthan Institute of Engineering and Technology, Rajasthan, 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*.

***Key Words*: 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. RELATED WORK

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 ∈ [0,100] and (a,b) ∈[−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.

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|),$$

And normalization term W_p

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

where,

I^{filtered} is the filtered image;

I is the original input image to be filtered;

x are the coordinates of the current pixel to be filtered;

Ω is the window centered in x, so $x_i \in \Omega$ is another pixel;

f_r is the range kernel for smoothing differences in intensities (this function can be a Gaussian function);

g_s is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function).

Figure 2.1 Comparing Gaussian Distribution method with Bilateral Filtering method

2.1.1 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].
This Vector Quantization algorithm is an iterative algorithm; the algorithm iteratively minimizes the total distortion by representing the training vectors by their corresponding codevectors. This requires an initial codebook C obtained by taking the average mean of the training vector X [1].
In this method, centroid of the entire training set is computed first and then to the centroid a constant error {1,1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1} is added and two code vectors V1 and V2 are generated as shown in the figure 2.2.1. After the codevector is initiated, clustering is performed based on the Euclidean distance for all the training vectors computed with codevectors V1 and V2. Training vectors are then clustered on the basis of smallest distance calculated into either of the clusters; V1 or V2.
These steps are repeated until the desired size of codebook is generated.

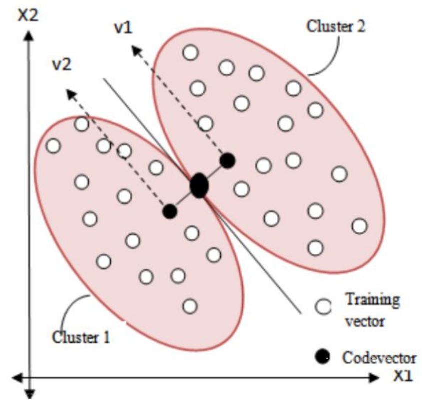

Figure 2.2.1 LBG codevectors V1 and V2

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, $C = \{c_1, c_2, c_3, \ldots, c_k\}$ be the codevector generated from the mean of training vectors, and $E = \{e_1, e_2, \ldots, e_k\}$ be the error vector. After computing c_j as min { c_i /i = 1,2, ... k} where j is the index of the vector, we then assign e_i = 1and if $c_i / c_j <= 10$ then set $e_i = c_i / c_j$ else, $e_i = 10$ for every $i \neq j$ 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 e_i is decided by the magnitude of coordinates of the centroid. Hence, the procedure is same as that of LBG.

2.4 Generative Adversarial Networks

GAN is an amalgamation of a generative model and a discriminative model, i.e is 2 CNNs. The generative model creates new occurrences of the data that simulate the training data.

The discriminator model is used for testing the data and contrasting it with the image from the Generator. Discriminator implies whether the output generated is fake or real. Both the generator and the discriminator are neural networks which run in competition (or against) with each other in the training phase. These steps are repeated multiple times so that the generator model and the discriminator model can reach a better outcome after repetitive steps.

We have formulated the procedure of studying algorithms to transform the real world photos into cartoon images as a mapping function which plots the image's numerous P to the cartoon's numerous C.

The mapping functions are learned with the help of the training data $S_{data}(p) = \{p_i \mid i = 1 \ldots N\} \subset P$ and $S_{data}(c) = \{c_i \mid i = 1 \ldots M\} \subset C$, where N and M are the numbers of photo and cartoon images in the training set, respectively. Referring to other GAN skeletons, a discriminative model function D is trained for pushing G to extend its purpose by differentiating the images in the cartoon manifold from other images and catering the adversarial loss for G.

Assuming L to be the loss function for the same, G∗ and D∗ be the weights of our networks. The prime motive is to solve the min-max problem here:

$$(G*, D*) = \arg\min G \max D\ L(G, D) \quad (1)$$

2.4.1 Network Architecture

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].

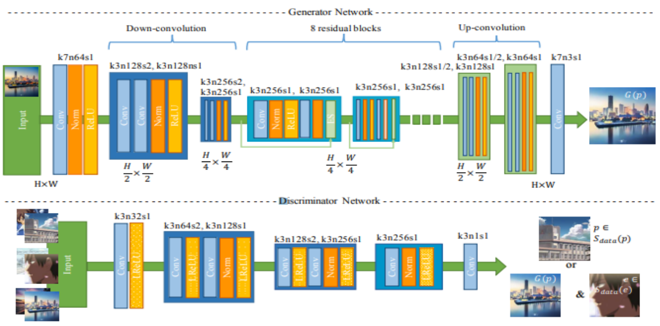

Figure 2.4.1 Architecture of the generator and discriminator networks in our reference model

where, k is the kernel size,
n is the number of feature maps,
s is the stride in each convolutional layer,
'norm' indicates a normalization layer,
'ES' indicates element wise sum.

The discriminator network D hereby is used to conclude whether the given input image is real. Since judging whether an image is cartoon or not is a less challenging job, hence, instead of going for a regular full-image discriminator model, a simplified patch-level discriminator model with fewer parameters in D will be appropriate. The pattern of implementing cartoon styled discrimination depends on the local features of an image which is different from object classification. Therefore, the network D is fabricated to be a shallow one. After the stage with flat layers, the network engages two strided convolution blocks to lessen the resolution and encode crucial local features for the classification. Later a feature construction block and a 3 × 3 convolutional layer are used to obtain the classification output. A Leaky ReLU (LReLU) with α = 0.2 is generally used after each normalization layer.

3. ALGORITHMIC STUDY TABLE

Year coined in	Algorithms used	Advantages of algorithm	Limitations of algorithm
1995	Bilateral filter	The Bilateral Filter is a Robust Filter - Expressed as optimization problems in a discretized space, it is possible to define some edge-preserving restoration formulations.	The bilateral filter in its direct form can introduce several types of image artefacts.
1980	Linde Buzo Gray	The centroid is calculated by taking the mean as the first code vector for the training set. Two centroids are generated by using constant error-addition to the codevector. Euclidean distances of all the training vectors are computed with	A constant error is added every time to divide the clusters in LBG, resulting in formation of clusters in one direction which is 1350 in 2-

		centroids and two clusters are formed based on the closest. Hence, this is a similar method to clustering and is easier to implement.	dimensional case. Because of this reason clustering is inefficient resulting in high MSE in LBG.
-	Kekre's Proportionate Error	Magnitude of elements of the codevector decides the error ratio. Hereafter the procedure is similar to that of LBG which is easier to perform.	The cluster orientation in KPE changes its variation limited to +/- 450 over 1350.
2014	Generative Adversarial Networks	The potential to generate high-resolution versions of input images. The ability to design and develop new and artistic images, sketches, paintings. The ability to transform photographs across domains, such as day to night, summer to winter etc.	Difficult to train: One needs to provide different types of data continuously to verify if it works precisely. Generating these results from text or speech is complex [3].

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].

Perhaps the most compelling application of GANs is in conditional GANs for tasks that require the generation of new examples. Here, Goodfellow indicates three main examples:

- **Image Super-Resolution-** The ability to generate high-resolution versions of input images.
- **Creating Art-** The ability to create great, new and artistic images, sketches, painting and more.
- **Image-to-Image Translation-** The ability to translate photographs across domains, such as day to night, summer to winter and more.

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.

[9] B. Weiss, "Fast median and bilateral filtering," ACM Transactions on Graphics,vol. 25, no. 3, pp. 519–526, Proceedings of the ACM SIGGRAPH conference,2006.

[10] H. Winnem¨oller, S. C. Olsen, and B. Gooch, "Real-time video abstraction,"ACM Transactions on Graphics, vol. 25, no. 3, pp. 1221–1226, Proceedings ofthe ACM SIGGRAPH conference, 2006.K. Elissa, "Title of paper if known," unpublished.

[11] https://en.wikipedia.org/wiki/Bilateral_filter.

[12] NIPS 2016 Tutorial: Generative Adversarial Networks conference https://arxiv.org/abs/1701.00160.

[13] H. B. Kekre, Tanuja K. Sarode, Kavita Raut and Rohan Tahiliani, "Performance Comparison of LBG KPE KFCG and KMCG for Global Codebook Technique", International Journal of Computer Applications, vol. 30, no. 10, pp. 42-50, Sep. 2011.

[14] Bang Huang, Linbo Xie, "An Improved LBG Algorithm for Image Vector Quantization", Computer Science and Information Technology (ICCSIT), vol. 6, 3rd IEEE International Conference, pp. 467- 471, July 2010.

[15] Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks, 2016 - Chuan Li, Michael Wand.

[16] Technical Paper Presentation on Application of Cartoon like Effects to Actual Images by Chinmay Joshi, Devendra Jaiswal, Akshata Patil.

[17] J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, (6):679–698, 1986.

[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In International Conference on Computer Vision, 2017.

[19] https://medium.com/analytics-vidhya/generating-japanese-style-animation-using-deep-learning-not-cyclegans-or-neural-style-transfer-1b48faeba931.