Cartoonize Image Using Deep Learning

Sk. Gouse Mastan Vali[1a], D. Vamsi Krishna[1b], A. Sairam[1c], K. S. Lakshmi Narayana[1d], P. Lakshmi Narayana[2]
[1]*Department of CSE, Narasaraopet Engineering College, Narasaraopet, A.P, India*
[a] gousemastan7867@gmail.com
[b] vamsi.9182049979gmail.com
[c] sairam55@gmail.com
[d] nanisreeram1999@gmail.com
[2]*Asst. Prof, Department of CSE, Narasaraopet Engineering College, Narasaraopet, A.P, India*
[2] lakshmi.778@gmail.com

***Abstract*-- This paper presents an approach for image cartooniza- tion. By observing the cartoon painting behavior and consulting artists, we propose to separately identify three white-box representations from images: the surface rep- resentation that contains a smooth surface of cartoon im- ages, the structure representation that refers to the sparse color-blocks and flatten global content in the celluloid style workflow, and the texture representation that reflects high- frequency texture, contours, and details in cartoon im- ages. A Generative Adversarial Network (GAN) framework is used to learn the extracted representations and to cartoonize images. The learning objectives of our method are separately based on each extracted representations, making our frame- work controllable and adjustable. This enables our ap- proach to meet artists' requirements in different styles and diverse use cases.**

***Keywords*-- Deep Learning, Convoulution Neural Networks, Cartoonization, White box representation, Cartoon Generative Adversarial Network (cartoonGAN).**

I. INTRODUCTION

Cartoon is a popular art form that has been widely ap- plied in diverse scenes. Modern cartoon animation work- flows allow artists to use a variety of sources to create con- tent. Some famous products have been created by turning real- world photography into usable cartoon scene materials, where the process is called image cartoonization.

The variety of cartoon styles and use cases require task- specific assumptions or prior knowledge to develop usable algorithms. For example, some cartoon workflows pay more attention to global palette themes, but the sharpness of lines is a secondary issue. In some other workflows, sparse and clean color blocks play a dominant role in artistic ex- pression, but the themes are relatively less emphasized.

CartoonGAN is designed for image cartoonization, in which a GAN frame- work with a novel edge loss is proposed and achieves good results in certain cases. But using a black- box model to directly fit the training data decreased its generality and stylization quality, causing bad cases in some situations.

we propose to decompose images into several cartoon representations, and list them as follows.

Firstly, we extract the surface representation to repre- sent the smooth surface of images. we extract a weighted low- frequency component, where the color composition and surface texture are preserved with edges, textures and details ignored.

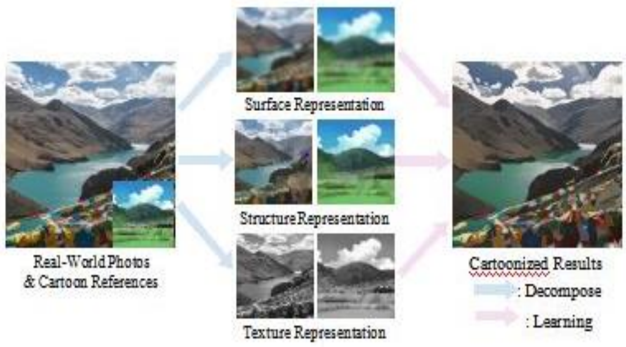

Fig. 1 Cartoonization Process

Figure 1: A simple illustration of our method. Images are decomposed into three cartoon representations, which guide the network optimization to generate cartoonized photos. Behavior where artists usually draw composition drafts before the details are retouched and is used to achieve a flexible and learnable feature representation for smoothed surfaces.

Secondly, the structure representation is proposed to ef- fectively seize the global structural information and sparse color blocks in celluloid cartoon style. We extract a seg- mentation map from the input image and then apply an adaptive coloring algorithm on each segmented regions to generate the structure representation. This representation is motivated to emulate the celluloid cartoon style, which is featured by clear boundaries and sparse color blocks.

The structure representation is of great significance for generating the sparse visual effects, as well as for our method to be embedded in the celluloid style cartoon workflow.

Thirdly, we use the texture representation to contain painted details and edges. The input image $I \in RW\times H\times 3$ is converted to a single-channel intensity map $It \in RW\times H\times 1$, where the color and luminance are removed and relative pixel intensity is preserved. This feature representation is motivated by a cartoon painting method where artists firstly draw a line sketch with contours and details, and then apply color on it. It guides the network to learn the high-frequency textural details independently with the color and luminance patterns excluded.

The separately extracted cartoon representations enable the cartooniaztion problem to be optimized end-to-end within a Generative Neural Networks (GAN) framework, making it scalable and controllable for practical use cases and easy to meet diversifified artistic demands with task specifific fifine-tuning. We test our method on a variety of realworld photos on diverse scenes in different styles. Experimental results show that our method can generate images with harmonious color, pleasing artistic styles, sharp and clean boundaries, and signifificantly fewer artifacts as well. We also show that our method outperforms previous stateof-the-art methods through qualitative experiments, quantitative experiments, and user studies. Finally, ablation studies are conducted to illustrate the inflfluence of each representation. To conclude, our contributions are as follows:

- We propose three cartoon representations based on our observation of cartoon painting behavior: the surface epresentation, the structure representation, and the texture representation. Image processing modules are then introduced to extract each representation.
- A GAN-based image cartoonization framework is optimized 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 our method can generate high-quality cartoonized images. Our method outperforms existing methods in qualitative comparison, quantitative comparison, and user preference.

II. RELATED WORK

A. Image Smoothing and Surface Extraction

Image smoothing is an extensively studied topic. Early methods are mainly filtering based and optimization-based methods later became popular. Farbman et al. utilized weighted least square to constrain the edge-preserving operator, Min et al. [29] solved global image smoothing by minimizing a quadratic energy function, and Bi et al. [5] proposed an L1 transformation for image smoothing and flattening problem. Xu and Fan et al. [44, 9] introduced end-to-end networks for image smoothing. In this work, we adapt a differentiable guided filter to extract smooth, cartoon-like surface from images, enabling our model to learn structure-level composition and smooth surface that artists have created in cartoon artworks.

B. Super pixel and structure Extraction

Super-pixel segmentation groups spatially connected pixels in an image with similar color or gray level. Some popular super pixel algorithms are graph-based, treating pixels as nodes and similarity between pixels as edges in a graph. Gradient ascent based algorithms initialize the image with rough clusters and iteratively optimize the clusters with gradient ascent until convergence. In this work, we follow the felzenszwalb algorithm to develop a cartoon-oriented segmentation method to achieve a learnable structure representation. This representation is significant for deep models to seize global content information and produce practically usable results for celluloid style cartoon workflows.

C. Non photorealistic Rendering (NPR)

Non-photorealistic Rendering (NPR) methods represent image content with artistic styles, such as pencil sketch- Ing, paints, watercolor. Image cartoonization is also extensively studied from filtering based method to end-to-end neural network.

Neural Style Transfer methods are popular among NPR algorithms, which synthesis images with artistic style by combining the content of one image and the style of another image. Gatys et al. jointly optimized a style loss and a content loss to generate stylize images with a style-content image pair. Johnson et al. accelerated stylization by training an end-to-end network with perception loss. Several works later proposed different methods to stylize images.

NPR methods are also widely used in image abstraction. These methods highlight semantic edges while filtering out image details, presenting abstracted visual in- formation of original images, and are commonly used for cartoon related applications. Our method, different from style transfer methods that use a single image as reference or image abstraction methods that simply consider content images, learns the cartoon data distribution from a set of cartoon images. This allows our model to synthesis high- quality cartoonized images on diverse use cases.

D. Generative Adversarial Networks

Generative Adversarial Network (GAN) is a state-of-the-art generative model that can generate data with the same distribution of input data by solving a min-max problem between a generator network and a discriminator net- work. It is powerful in image synthesis by forcing the gener- ated images to be indistinguishable from real images. GAN has been widely used in conditional image generation tasks, such as image in painting, style transfer, image car- toonization, image colorization. In our method, we adopt adversarial training architecture and use two discrim- inators to enforce the generator network to synthesize im- ages with the same distribution as the target domain.

E. Image to Image Translation

Image-to-Image Translation tackles the problem of translating images from a source domain to an- another target domain. Its applications include image quality enhancement, stylizing photos into paints, cartoon images and sketches, as well as grayscale photo colorization and sketch colorization. Recently, bi-directional models are also introduced for inter-domain translation. Zhu et al. performs transformation of unpaired images (i.e., summer to winter, photo to paints). In this paper, we adopt an unpaired image-to-image translation framework for image cartoonization. Unlike previous black-

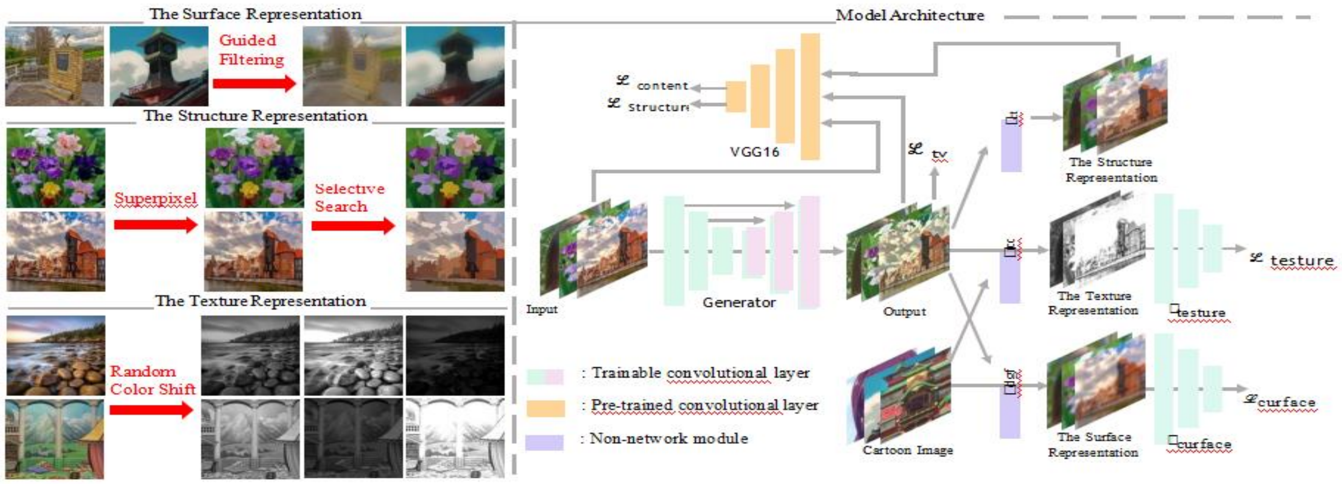

Fig. 2 Proposed image cartoonization system

box models that guide network training with loss terms, we decompose images into several representations, which enforces network to learn different features with separate objectives, making the learning process controllable.

III. PROPOSED WORK

We show our proposed image cartoonizaiton framework in Figure 2. Images are decomposed into the surface representation, the structure representation, and the texture representations, and three independent modules are introduced to extract corresponding representations. A GAN frame- work with a generator G and two discriminators Ds and Dt is proposed, where Ds aims to distinguish between surface representation extracted from model outputs and cartoons, and Dt is used to distinguish between texture representation extracted from outputs and cartoons. Pre-trained VGG network is used to extract high-level features and to im- pose spatial constrain on global contents between extracted structure representations and outputs, and between input photos and outputs.

A. Learning From the Surface Representation

The surface representation imitates cartoon painting style where artists roughly draw drafts with coarse brushes and have smooth surfaces like cartoon images. To smooth images and meanwhile keep the global semantic structure, a differentiable guided filter is adopted for edge- preserving filtering. Outputs and reference cartoon images have similar surfaces, and guide the generator to learn the information stored in the extracted surface representation.

B. Learning From the Structure representation

The Structure representation emulates flattened global content, sparse color blocks, and clear boundaries in celluloid style cartoon workflow. We at first use felzenszwalb algorithm to segment images into separate regions. As super pixel algorithms only consider the similarity of pixels and ignore semantic information, we further introduce selective search to merge segmented regions and extract a sparse segmentation map.

Standard super pixel algorithms color each segmented region with an average of the pixel value. By analyzing the processed dataset.

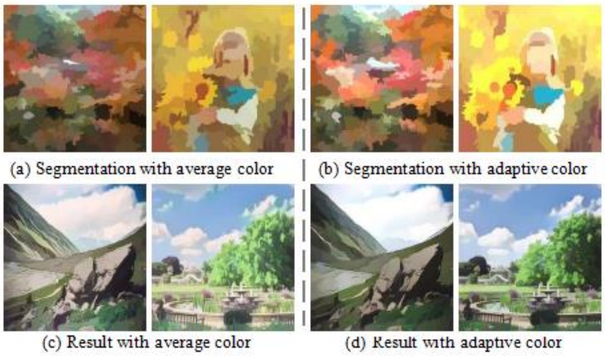

Fig. 3 Adaptive coloring algorithm.

Figure 3: Adaptive coloring algorithm. (a) and (b) show segmentation maps with different coloring method, while (c) and (d) shows results generated with different coloring method. Adaptive coloring generates results that are brighter and free from hazing effects.

We use high-level features extracted by pre-trained VGG16 network to enforce spatial constrain between our results and extracted structure representation. Let Fst denote the structure representation extraction, the structure loss Lstructure is formulated as:

$$L_{structure} = \| VGG_n(G(I_p)) - VGG_n(F_{st}(G(I_p))) \| \quad (1)$$

C. Learning From the Textural Representation

The high-frequency features of cartoon images are key learning objectives, but luminance and color information make it easy to distinguish between cartoon images and real-world photos. We thus propose a random color shift algorithm Frcs to extract single-channel texture representation from color images, which retains high-frequency textures and decreases the influence of color and luminance.

$$Frcs(Irgb) = (1-\alpha)(\beta 1*Ir+\beta 2*Ig+\beta 3*Ib)+\alpha*Y \quad (2)$$

Color image, the random color shift can generate random intensity maps with luminance and color information removed. A discriminator Dt is introduced to distinguish texture representations extracted from model outputs and cartoons, and guide the generator to learn the clear con- tours and fine textures stored in the texture representations.

$$Ltexture(G, Dt) = \log Dt(Frcs(Ic)) \quad (3)$$

D. Full model

Our full model is a GAN based framework with one generator and two discriminators. It is jointly optimized with features learned from three cartoon representations and could be formulated in Equation 4. By adjusting and bal- ancing λ1, λ2, λ3 and λ4, it could be easily adapted to vari- ous applications with different artistic style.

$$Ltotal = \lambda 1 * Lsurface + \lambda 2 * Ltexture$$
$$+ \lambda 3 * Lstructure + \lambda 4 * Lcontent + \lambda 5 * Ltv$$

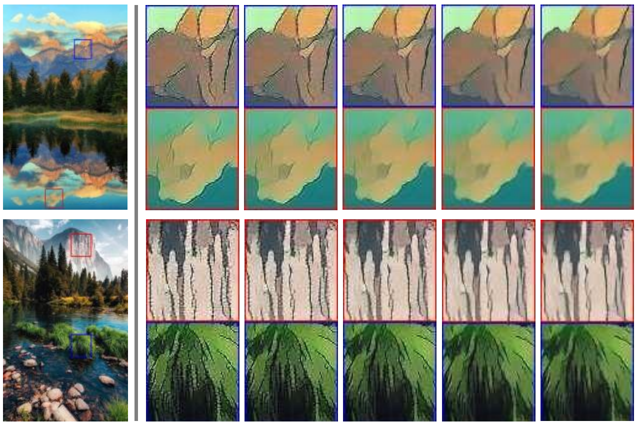

Fig. 4 sharpness adjusted by style interpolation.

Figure 4: The sharpness of details could be adjusted by style interpolation. δ = 0.0, 0.25, 0.5, 0.75, 1.0 from left to right.

The total-variation loss Ltv is used to impose spatial smoothness on generated images. It also reduces high-frequency noises such as salt-and-pepper noise. In Equation 5, H, W, C represent spatial dimensions of images.

$$L_{tv} = \frac{1}{H*W*C} \| \nabla_x (G(I_p)) + \nabla_y (G(I_p)) \| \quad (5)$$

The content loss $L_{content}$ is used to ensure that the cartoonized results and input photos are semantically invariant, and the sparsity of L1normallowsforlocalfeaturesto be cartoonized. Similarto the structure loss, itis calculated on pre-trained VGG16 featurespace:

$$Lcontent = \|VGGn(G(Ip)) - VGGn(Ip)\| \quad (6)$$

To adjust the sharpness we have adapted the differentiable guided filter Fdgf for style interpolation. Shown in Figure 6, it can effectively tune the sharpness of details and edges without fine-turning the network parameters. Denote the network input as Iin and network output as Iout, we formulated the post-processing in Equation 9, where Iin isused as guidemap:

$$I_{interp} = \delta * F_{dgf}(I_{in}, G(I_{in})) + (1-\delta) * G(I_{in}) \quad (7)$$

IV. EXPERIMENTAL RESULT

A. Experimental Setup

Implementation. We implement our GAN method with TensorFlow. The generator and discriminator architec- tures are described in the supplementary material. Patch discriminator is adopted to simplify calculation and en- hance discriminative capacity.

TABLE 1
PERFORMANCE AND MODEL SIZE COMPARISION

Methods	[20]	[6]	[48]	Ours
LR, CPU(ms)	639.31	1947.97	1332.66	64.66
LR, GPU(ms)	16.53	13.76	9.22	3.58
HR, GPU(ms)	48.96	148.02	106.82	17.23
Parameter(M)	1.68	11.38	11.13	1.48

Table 1: Performance and model size comparison, LR means 256*256 resolution, HR means 720*1280 resolution.

Human face and landscape data are collected for generalization on diverse scenes. For real-world photos, we collect 10000 images from the FFHQ dataset [22] for the human face and 5000 images from the dataset in [48] for landscape. For cartoon images, we collect 10000 images from animations for the human face and 10000 images for landscape. Producers of collected animationsinclude Kyoto animation, P.A.Works, Shinkai Makoto, Hosoda Mamoru, and Miyazaki Hayao. For the validation set, we collect 3011 animation images and 1978 real-world photos. Images shown in the main paper are collected from the DIV2K dataset [3], and images in user study are collected from the Internet and Microsoft COCO [27] dataset. During training, all images are resized to 256*256 resolution, and face images are feed only once in every five iterations. Previous Methods. We compare our method with four algorithms that represent Neural Style Transfer [20], Imageto-Image Translation [48], Image Abstraction [21] and Image Cartoonization [6] respectively. Evaluation metrics. In qualitative experiments, we present results with details of four different methods and original images, as well as qualitative analysis. In quantitative experiments, we use Frechet Inception Distance (FID) [15] to evaluate the performance by calculating the distance between source image distribution and target image distribution. In the user study, candidates are asked to rate the results of different methods between 1 to 5 in cartoon quality and overall quality. Higher scores mean better quality. Time Performance and Model Size. Speeds of four methods are compared on different hardware and shown in Table 1. Our model is the fastest among four methods on all devices and all resolutions, and has the smallest model size. Especially, our model can process a 720*1280 image on GPU within only 17.23ms, which enables it for real-time.

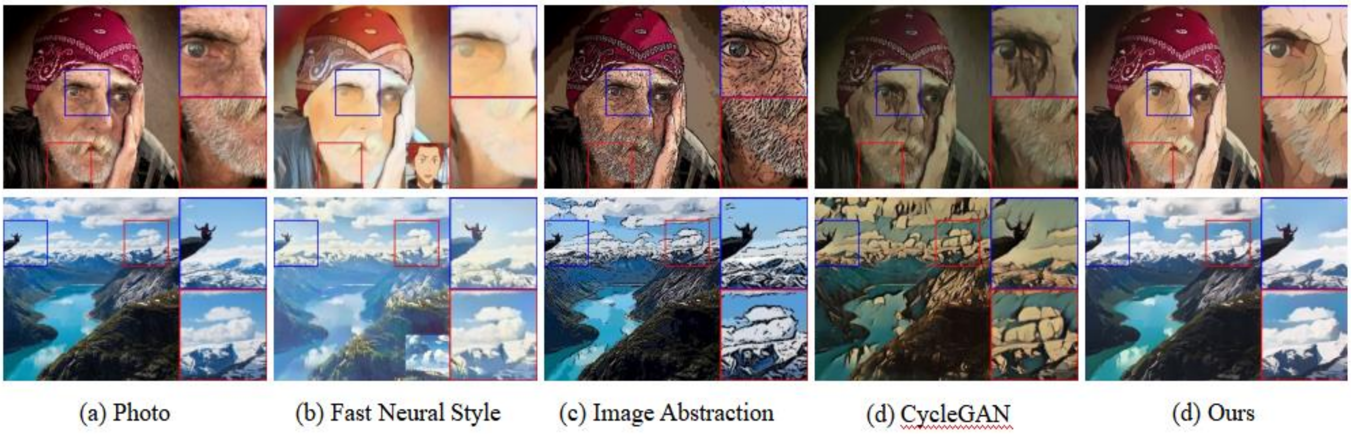

Fig. 5 Qualitative comparison, second raw shows 4 different styles of Cartoon GAN

TABLE 2
PERFORMANCE EVALUTION BASED ON FID

Methods	Photo	Fast Neural Style	CycleGAN	Image Abstraction	Ours
FID to Cartoon	162.89	146.34	141.50	130.38	101.31
FID to Photo	N/A	103.48	122.12	75.28	28.79
Methods	Shinkai style of	Hosoda style of	Hayao style of	Paprika style of	Ours
FID to Cartoon	135.94	130.76	127.35	127.05	101.31
FID to Photo	37.96	58.13	86.48	118.56	28.79

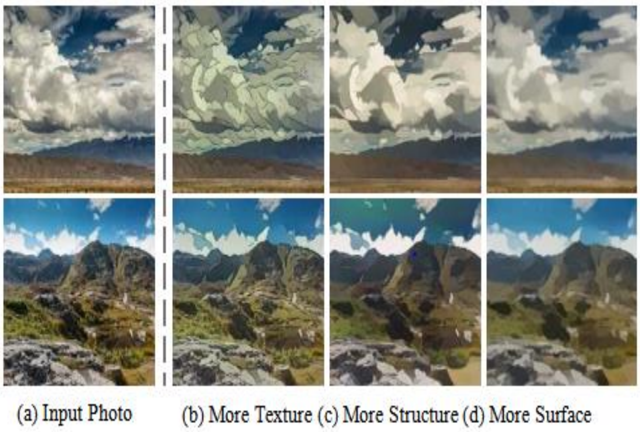

Fig. 6 Output quality could be controlled by adjusting weight of each representation. Zoom in for details.

Generality to diverse use cases. We apply our model on diverse real-world scenes, including natural landscape, city views, people, animals, and plants, and show the results Validation of Cartoon Representations.

B. Validation of Cartoon Representations

To validate our proposed cartoon representations reason able and effective, a classification experiment and a quantitative experiment based on FID are conducted, and the results are shown in Table 2. We train a binary classifier on our training dataset to distinguish between real-world photos and cartoon images. The classifier is designed by adding a fully-connected layer to the discriminator in our framework. The trained classifier is then evaluated on the validation set to validate the inflfluence of each cartoon representation.

TABLE 3
CLASSIFICATION ACCURACY AND FID EVALUTION OF OUR PROPOSED CARTOON REPRESENTATION.

No.	Surface	Structure	Texture	Original
Acc	0.8201	0.6342	0.8384	0.9481
FID	113.57	112.86	112.71	162.89

We fifind the extracted representations successfully fool the trained classififier, as it achieves lower accuracy in all three extracted cartoon representations compared to the original images. The calculated FID metrics also support our proposal that cartoon representations help close the gap between real-world photos and cartoon images, as all three extracted cartoon representations have smaller FID compared to the original images.

C. Illustration of Controllability

As is shown in Figure 8, the style of cartoonized results could be adjusted by turning the weight of each representation in the loss function. Increase the weight of texture representation adds more details in the images, rich details such as grassland and stones are preserved. This is because it regulates dataset distributions and enhances high-frequency details stored in texture representation. Smoother textures and fewer details are generated with a higher weight of surface representation, the details of the cloud and the mountain are smoothed. The reason is that guided fifiltering smooths training samples and reduces densely textured patterns. To get more abstract and sparse features, we can increase the weight of structure representation, and the details of the mountains are abstracted into sparse color blocks. This is because the selective search algorithm flflattens the training data and abstract them into structure representations. To conclude, unlike black-box models, our white-box method is controllable and can be easily adjusted.

D. Qualitative Comparison

Comparisons between our method and previous methods are shown in Figure 9. The white-box framework helps generate clean contours. Image abstraction causes noisy and messy contours, and other previous methods fail to generate clear borderlines, while our method has clear boundaries, such as human face and clouds. Cartoon representations also help keep color harmonious. CycleGAN generates darkened images and Fast Neural Style causes oversmoothed color, and CartoonGAN distorts colors like human faces and ships. Our method, on the contrary, prevents improper color modififications such as faces and ships. Lastly, our method effectively reduces artifacts while preserves fifine details, such as the man sitting on the stone, but all other methods cause over-smoothed features or distortions. Also, methods like CycleGAN, image abstraction and some style of CartoonGAN cause high-frequency artifacts. To conclude, our method outperforms previous methods in generating images with harmonious color, clean boundaries, fifine details, and fewer noises.

E. Quantitative Evaluation

Frechet Inception Distance (FID) [15] is wildly-used to quantitatively evaluate the quality of synthesized images. Pre-trained Inception-V3 model [36] is used to extract highlevel features of images and calculate the distance between two image distributions. We use FID to evaluate the performance of previous methods and our method. As CartoonGAN models have not been trained on human face data, for fair comparisons, we only calculate FID on scenery dataset. As is shown in Table 3, our method generates images with the smallest FID to cartoon image distribution, which proves it generates results most similar to cartoon images. The output of our method also has the smallest FID to realworld photo distribution, indicating that our method loyally preserves image content information.

TABLE 4
RESULT OF USER STUDY

Methods	[20]	[6]	[48]	Ours
Cartoon quality, mean	2.347	2.940	2.977	4.017
Cartoon quality, std	1.021	1.047	1.437	0.962
Overall quality, mean	2.38	2.937	2.743	3.877
Overall quality, std	0.993	1.046	1.321	0.982

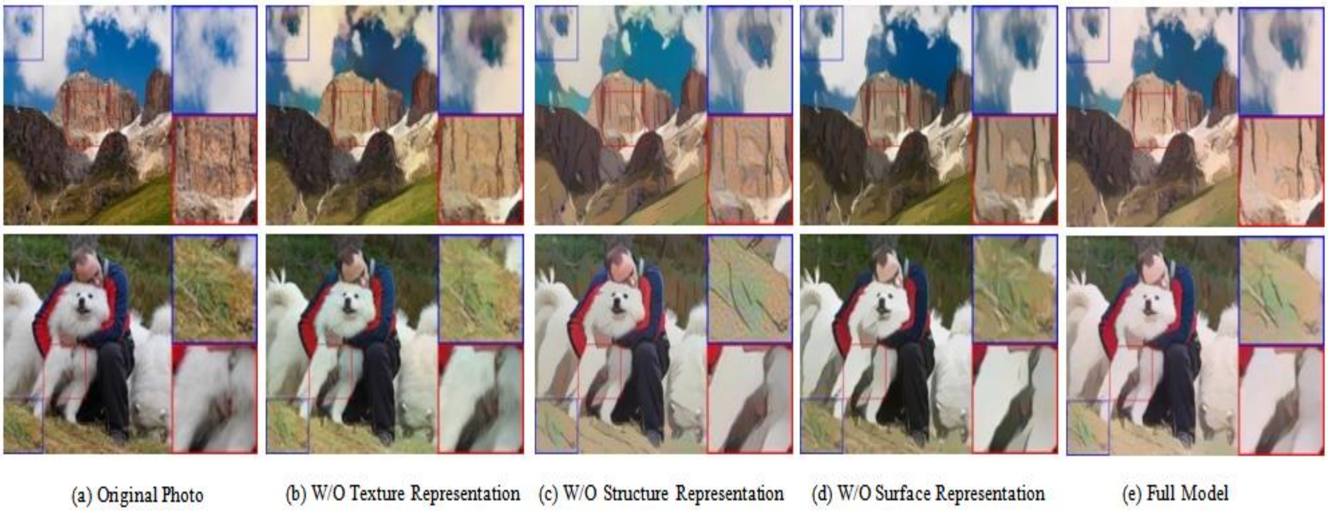

Fig. 7 Ablation study by removing each component

Table 4: Result of User study, higher score means better quality. Row 1 and 2 represent the mean and standard error of Cartoon quality score, row 3 and 4 represent the mean and standard error of Overall quality score.

F. User Study

The quality of Image cartoonization is highly subjective and greatly influenced by individual preference. We con- ducted user studies to show how users evaluate our method and previous methods. The user study involves 30 images, each processed by our proposed method and three previous methods. Ten candidates are asked to rate every image be- tween 1-5 in 2 dimensions, following the criterion below: Cartoon quality: users are asked to evaluate how similar are the shown images and cartoon images. Overall quality: users are asked to evaluate whether there are color shifts, texture distortions, high-frequency noises, or other artifacts they dislike on the images.

We collect 1200 scores in total, and show the average score and standard error of each algorithm Table 4. Our method outperforms previous methods in both cartoon qual- ity and overall quality, as we get higher scores in both criteria. This is because our proposed representations effectively extracted cartoon features, enabling the network to synthesize images with good quality. The synthesis quality of our method is also the most stable, as our method has the smallest standard error in both criteria. The reason is that our method is controllable and can be stabilized by balancing different components. To conclude, our method outperforms all previous methods shown in the user study.

G. Analysis of Each Components

We show the results of ablation studies in Figure 7. Ablating the texture representation causes messy details.Shown in Figure 7(a), irregular textures on the grassland and the dog's leg remains. This is due to the lack of high- frequency stored in the surface representation, which deteri- orates the model's cartoonization ability. Ablating the struc- ture representation causes high-frequency noises in Figure 7(b). Severe pepper-and-salt appear on the grassland and the mountain. This is because the structure representation flattened images and removed high-frequency information. Ablating the surface representation causes both noise and messy details. Unclear edges of the cloud and noises on the grassland appear in Figure 7(c). The reason is that guided filtering suppresses high-frequency information and preserves smooth surfaces. As a comparison, the results of our full model are shown in Figure 7(d), which have smooth features, clear boundaries, and much less noise. In conclusion, all three representations help improve the car- toonizaiton ability of our method.

V. CONCLUSION

In this paper, we propose a white-box controllable image cartoonization framework based on GAN, which can gener- ate high-quality cartoonized images from real-world photos. Images are decomposed into three cartoon representations: the surface representation, the structure representation, and the texture representation. Corresponding image process- ing modules are used to extract three representations for network training, and output styles could be controlled by adjusting the weight of each representation in the loss func- tion. Extensive quantitative and qualitative experiments, as well as user studies, have been conducted to validate the performance of our method. Ablation studies are also con- ducted to demonstrate the influence of each representation.

ACKNOWLEDGEMENT

This work is conducted in collaboration with the Team of Computer Science Department of narasaraopeta engineering college . We would like to thank M,sireesha for guiding during the project development . We are also grateful for the help from Dr. S N Tirumalarao, Srikanth Vemuru.

REFERENCES

[1] Mart´ın Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghe- mawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In 12th Sym- posium on Operating Systems Design and Implementation, pages 265–283, 2016. 5

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Su¨sstrunk. Slic superpix- els compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2274–2282, 2012. 2

[3] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In The IEEE Conference on Computer Vision and Pattern Recogni- tion (CVPR) Workshops, July 2017. 5

[4] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9465–9474, 2018. 1, 3, 5, 7, 8

[5] Dorin Comaniciu and Peter Meer. Mean shift: A robust ap- proach toward feature space analysis. IEEE Transactions on Pattern Analysis & Machine Intelligence, (5):603–619, 2002. 2

[6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kud- lur. A learned representation for artistic style. arXiv preprint arXiv:1610.07629, 2016. 3

[7] Qingnan Fan, Jiaolong Yang, David Wipf, Baoquan Chen, and Xin Tong. Image smoothing via unsupervised learning. In SIGGRAPH Asia 2018 Technical Papers, page 259. ACM, 2018. 2

[8] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In ACM Transactions on Graphics (TOG), volume 27, page 67. ACM, 2008. 2

[9] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59(2):167–181, 2004. 2

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015. 3

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014. 3

[12] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image fil- tering. In European Conference on Computer Vision, pages 1–14. Springer, 2010. 2

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilib- rium. In Advances in Neural Information Processing Sys- tems, pages 6626–6637, 2017. 5, 7

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceed- ings of the IEEE International Conference on Computer Vi- sion, pages 1501–1510, 2017. 3

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In Proceedings of the European Conference on Computer Vi- sion (ECCV), pages 172–189, 2018. 3

[16] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni- tion Workshops, pages 691–700, 2018. 3

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adver- sarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1125– 1134, 2017. 3, 5

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, Springer, 2016. 1, 3, 5, 7, 8

[19] Henry Kang, Seungyong Lee, and Charles K Chui. Flow-based image abstraction. IEEE transactions on visualization and computer graphics, 15(1):62–76, 2008. 3, 5, 7

[20] Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, and Ming-Hsuan Yang. Im2pencil: Controllable pencil illustra- tion from photographs. In Proceedings of the IEEE Con- ference on Computer Vision and Pattern Recognition, pages 1525–1534, 2019. 3

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dolla´r, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European Conference on Computer Vision, pages 740–755. Springer, 2014. 5

[22] Cewu Lu, Li Xu, and Jiaya Jia. Combining sketch and tone for pencil drawing production. In Proceedings of the Sympo- sium on Non-Photorealistic Animation and Rendering, pages 65–73. Eurographics Association, 2012. 3

[23] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European Conference on Computer Vision, pages 649–666. Springer, 2016. 3

[24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of IEEE In- ternational Conference on Computer Vision, 2017. 1, 3, 5, 7, 8

Moturi Sir., Tirumala Rao S.N., Vemuru S. (2021) Risk https://doi.org/10.1007/978-981-15-9516-5_37

[25] M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Optimized Feature Extraction and Hybrid Classification Model for Heart Disease and Breast Cancer Prediction International Journal of Recent Technology and

Engineering Vol - 7, No 6, Mar - 2019 ISSN - 2277-3878, Pages – 1754 – 1772

[26] M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Frequent Itemset Mining Algorithms: A Survey Journal of Theoretical and Applied Information Technology Vol - 96, No .3, Feb - 2018 ISSN - 1992-8645, Pages – 744 – 755

[27] M.Sireesha, Srikanth Vemuru, S.N.Tirumala Rao "Classification Model for Prediction Of Heart Disease Using Correlation Coefficient Technique" International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 2, March - April 2020, Pages-2116 – 2123.

[28] M. Sireesha, Srikanth Vemuru and S. N. TirumalaRao, "Coalesce based binary table: an enhanced algorithm for mining frequent patterns", International Journal of Engineering and Technology, vol. 7, no. 1.5, pp. 51-55, 2018.

[29] Sireesha Moturi , Dr. S. N. Tirumala Rao, Dr. Srikanth Vemuru,. (2020). Predictive Analysis of Imbalance

[30] Cardiovascular Disease Using SMOTE. International Journal of Advanced Science and Technology, 29(05), 6301 - 6311.

Prediction-Based Breast Cancer Diagnosis Using Personal
Health Records and Machine Learning Models. In:
Bhattacharyya D., Thirupathi Rao N. (eds) Machine
Intelligence and Soft Computing. Advances in Intelligent
Systems and Computing, vol 1280. Springer, Singapore.