WHITE BOX CARTOONIZATION USING AN EXTENDED GAN FRAMEWORK

Mini- Project

Project by:

AMEY THAKUR	TE COMPS B-50	TU3F1819127
HASAN RIZVI	TE COMPS B-51	TU3F1819130
MEGA SATISH	TE COMPS B-58	TU3F1819139

A. A frame in animation “Garden of words”

B. A real photo processed by WBC method

Comparison between a real cartoon image and an image processed by our method.

What WBC model is capable of?

Original
Video

Cartoonized
Version

ABSTRACT

- Implement a GAN framework and develop a white-box controllable image cartoonization, which can generate high-quality cartoonized images from real-world photos.

- We suggest to separately identify three white-box representations from images:
  1. Surface representation - contains a smooth surface of cartoon images.
  2. Structure representation - refers to the scattered color-blocks and flatten global content.
  3. Structure representation - reflects high frequency texture, contours, and details in cartoon images.

- The learning objectives are separately based on each extracted representations, making our framework controllable and adjustable.

INTRODUCTION

- Wide applications of Cartoons in diverse scenes such as: illustrations for children's books, anime, movies, video games.
- Some cartoon artwork were created based on real world scenes. However, manual re-creation of real life based scenes can be very laborious and requires refined skills.
- The evolution in the field of Machine Learning has expanded the possibilities of creating visual arts. Some famous products have been created by turning real-world photography into usable cartoon scene materials.

LITERATURE SURVEY

- CartoonGAN is one of the existing technology to generate cartoonized images but it adds noise and reduce quality of image. On the other hand White Box Cartoonization overcomes these problems and results in more precise and sharp images.

- This project will be based on the model proposed by students of University of Tokyo, Xinrui Wang and Jinze Yu - Learning to Cartoonize Using White-Box Cartoon Representations.

PROBLEM STATEMENT

- Many black-box cartoonization frameworks exist, although they provide users with limited control or adjustability when rendering real-world photography into cartoon scenes.

- So we are proposing a less time consuming web app capable of cartoonize images with precision and accuracy just by giving input image into a GAN framework which will result a cartoon image.

OBJECTIVES

The objectives of this project are:

- Develop modern cartoon animation workflows which allow artists to use a variety of sources to create content without the need of drawing.

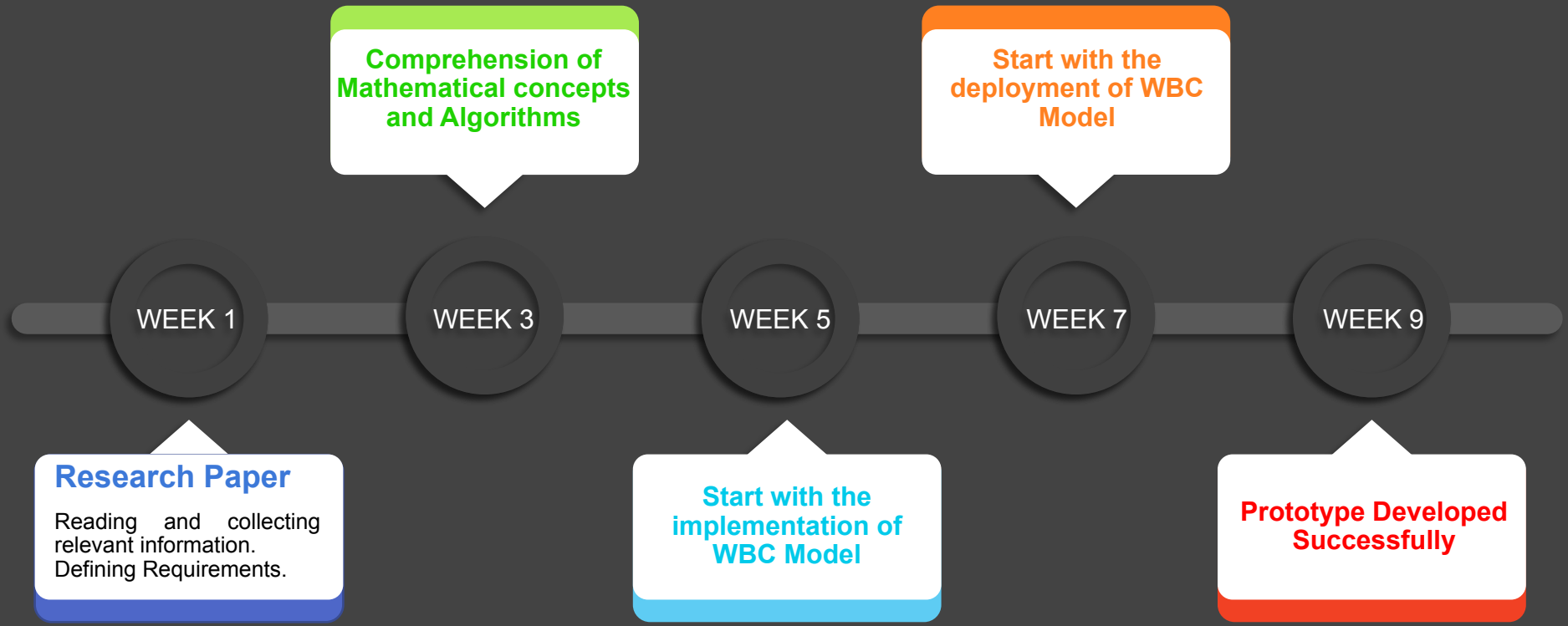
- Deploy GAN framework and create a Web application that can convert any images into high quality cartoon images by simply uploading the picture online and getting the cartoon image.

SCOPE

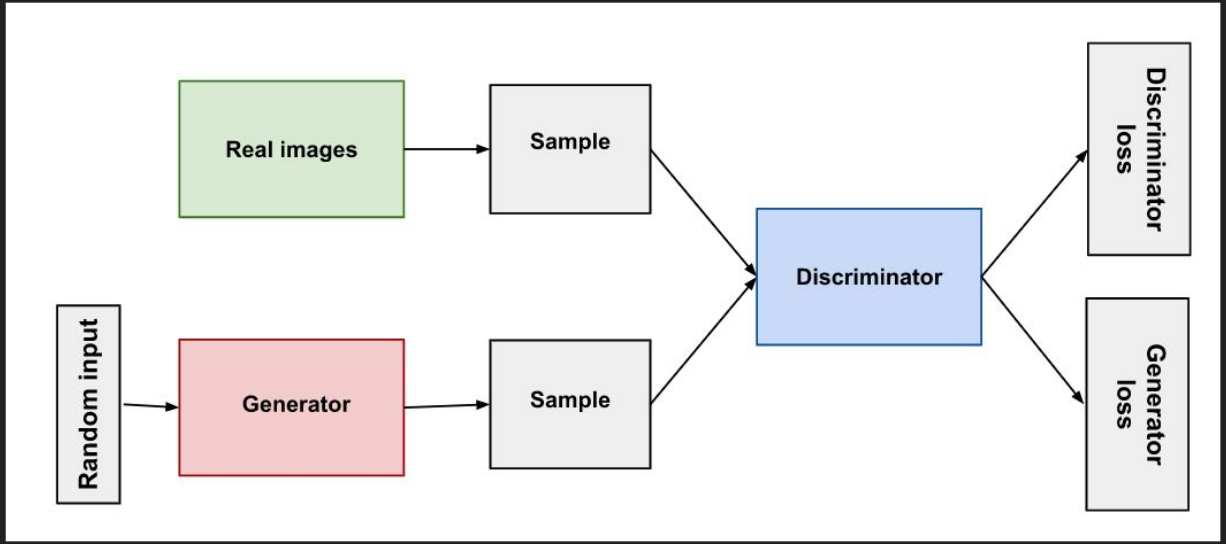
More deep work and development in this white box cartoonization model can lead to various applications:

1. Generation of quick prototypes for anime, cartoons and games.
2. Generation of minimal art, since it subdues facial features and information.
3. Games can import shortcut scenes very easily without using motion-capture.
4. Modelled as an assistant to graphic designers or animators.

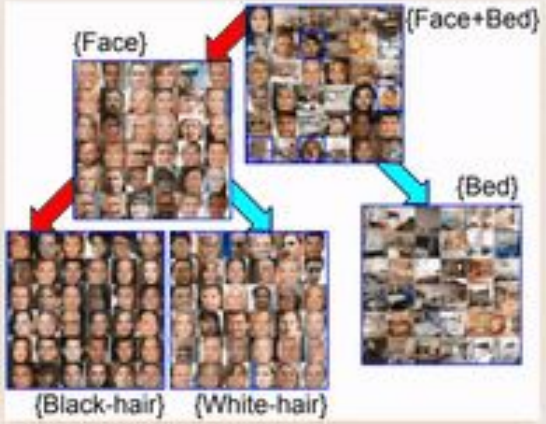
TIMELINE


Comprehension of Mathematical concepts and Algorithms
Start with the deployment of WBC Model
WEEK 1
WEEK 3
WEEK 5
WEEK 7
WEEK 9
Research Paper
Reading and collecting relevant information.
Defining Requirements.
Start with the implementation of WBC Model
Prototype Developed Successfully

GAN-TREE: A GAN FRAMEWORK FOR GENERATING DIVERSE DATA



Architecture of GAN


{Face}
{Face+Bed}
{Bed}
{Black-hair}
{White-hair}

PROS OF GAN FRAMEWORK

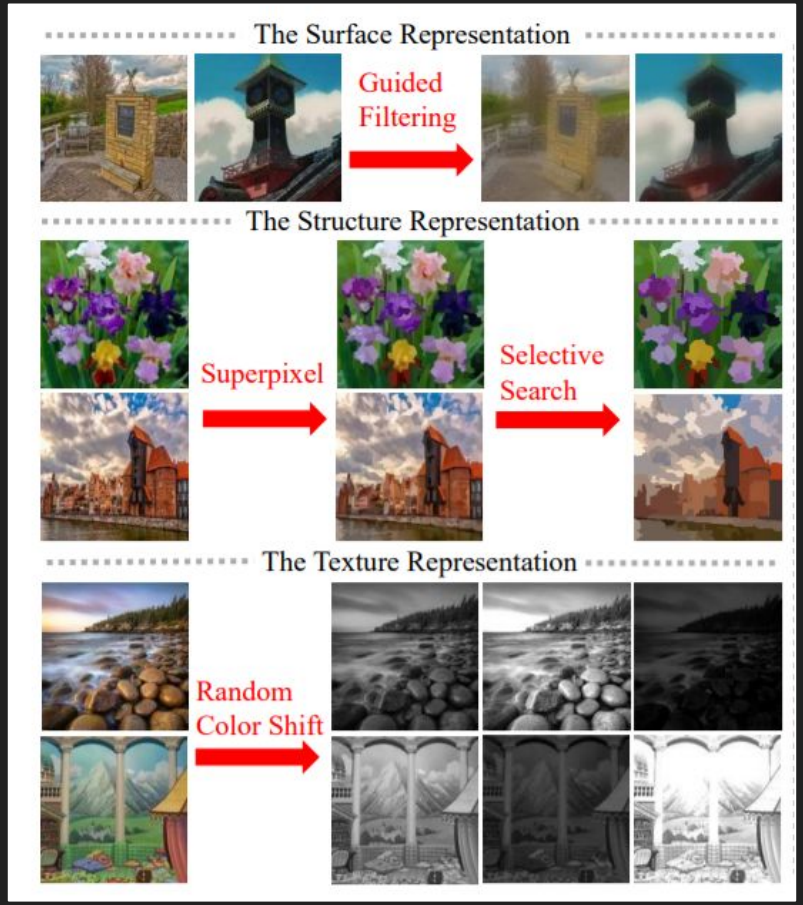
+ Very impressive result related to content generated, and realistic visual content.
+ Also can be modelled for other applications like image restoration, etc.

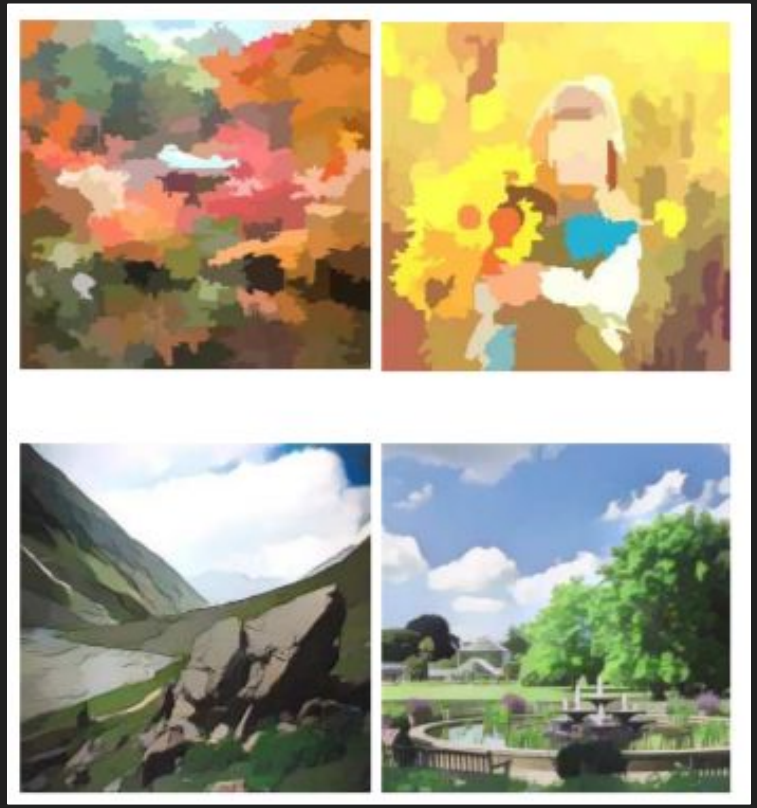
CONS OF GAN FRAMEWORK

Despite the success of GAN, the potential of such a framework has certain limitations.

- Difficult for training: huge computation.
- "Realistic" but not Real. A fake pattern can be created, especially for small scale structure or non-nature object like text ---> problem of the loss function.
- Some physical effects in the picture like shadow will become unrealistic.

METHODOLOGIES OF WHITE BOX CARTOONIZATION





Adaptive coloring generates results that are brighter and free from hazing effects.

SURFACE LOSS FORMULA

$$L_{surface}(G, D_s) = \log D_s(F_{dgf}(I_c, I_c)) + \log(1 - D_s(F_{dgf}(G(I_p), G(I_p))))$$

Where,

G = Generator,

D_s = Discriminator,

I_c = Reference Cartoon Image,

I_p = Input Photo,

F_{dgf} = It takes an image I as input and itself as guide map, returns extracted surface representation Fdgf (I, I) with textures and details removed.

Note: A discriminator Ds is introduced to judge whether model outputs and reference cartoon images have similar surfaces, and guide the generator G to learn the information stored in the extracted surface representation.

STRUCTURE LOSS FORMULA

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

Where,

G = Generator,

I_p = Input Photo,

F_{st} = Structure Representation Extraction.

Note: We use high-level features extracted by a pre-trained VGG16 network to enforce spatial constraint between our results and extracted structure representation.

TEXTURAL REPRESENTATION FORMULA

$$L_{texture}(G, D_t) = \log D_t(F_{rcs}(I_c)) + \log(1 - D_t(F_{rcs}(G(I_p))))$$

Where,

G = Generator,

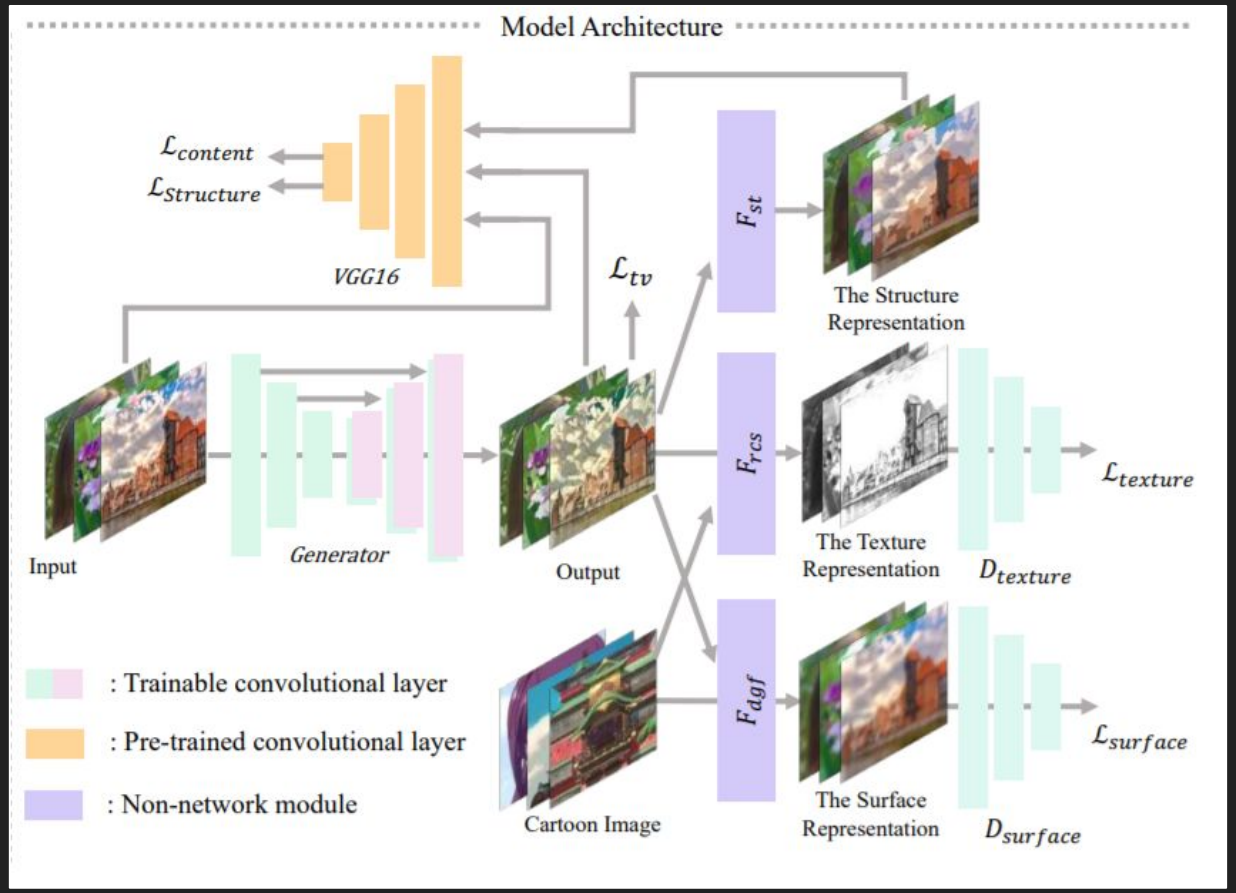
D_t = Discriminator,

I_c = Reference Cartoon Image,

I_p = Input Photo,

F_{rcs} = Extract single-channel texture representation from color images, which retains high-frequency textures and decreases the influence of color and luminance.

ARCHITECTURE OF THE PROPOSED WBC MODEL



Architecture of WBC Model

TOOLS AND TECHNOLOGIES USED

TOOLS USED

- Anaconda
- Jupyter Notebook
- TensorFlow.js
- Bootstrap
- Flexbox
- GitHub Pages
- Colab

TECHNOLOGIES USED

- HTML
- CSS
- JavaScript
- Python

IMPLEMENTATION

GitHub Repository - https://github.com/rizvihasan/whitebox_cartoonisation-webapp

- We implemented our GAN method with TensorFlow.
- Patch discriminator is adopted to simplify calculation and enhance discriminative capacity.
- We use Adam algorithm to optimize both Generator and Discriminator networks.
- Learning rate and batch size are set to 2 ∗ 10^−4 and 16 during training. We at first pre-train the generator with the content loss for 50000 iterations, and then jointly optimize the GAN based framework. Training is stopped after 100000 iterations or on convergence.

DATASET

- Human face and landscape data are collected for generalization on diverse scenes.
- For real-world photos, we collect 10000 images from the FFHQ dataset for the human face and 5000 images from the dataset in for landscape.
- For cartoon images, we collect 10000 images from animations for the human face and 10000 images for landscape.
- 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, and images in user study are collected from the Internet and Microsoft COCO dataset.
- During training, all images are resized to 256*256 resolution, and face images are feed only once in every five iterations.

TIME PERFORMANCE AND MODEL SIZE

- 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 High-Resolution video processing tasks.
- Generality to diverse use cases: We apply our model on diverse real-world scenes, including natural landscape, city views, people, animals, and plants.

CartoonGAN

- CartoonGAN, a generative adversarial network (GAN) framework for cartoon stylization.This method takes a set of photos and a set of cartoon images for training.
- CartoonGAN is basically a dedicated GAN-based architecture together with two simple yet effective loss functions.
- When cartoon images from individual artists are used for training, our method is able to reproduce their styles.

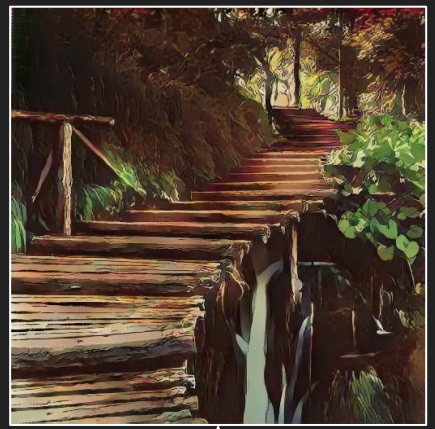
GANILLA

- Currently, state-of-the-art image-to-image translation models successfully transfer either the style or the content, they fail to transfer both at the same time.

- GANILLA is a new network to address this issue and show that the resulting network strikes a better balance between style and content, where both content and style are taken into account using separate classifiers.

COMPARISON BETWEEN CARTOONGAN, GANILLA AND WBC

Input Image



CartoonGAN

GANILLA

WBC

COMPARISON BETWEEN CARTOONGAN, GANILLA AND WBC

Input Image

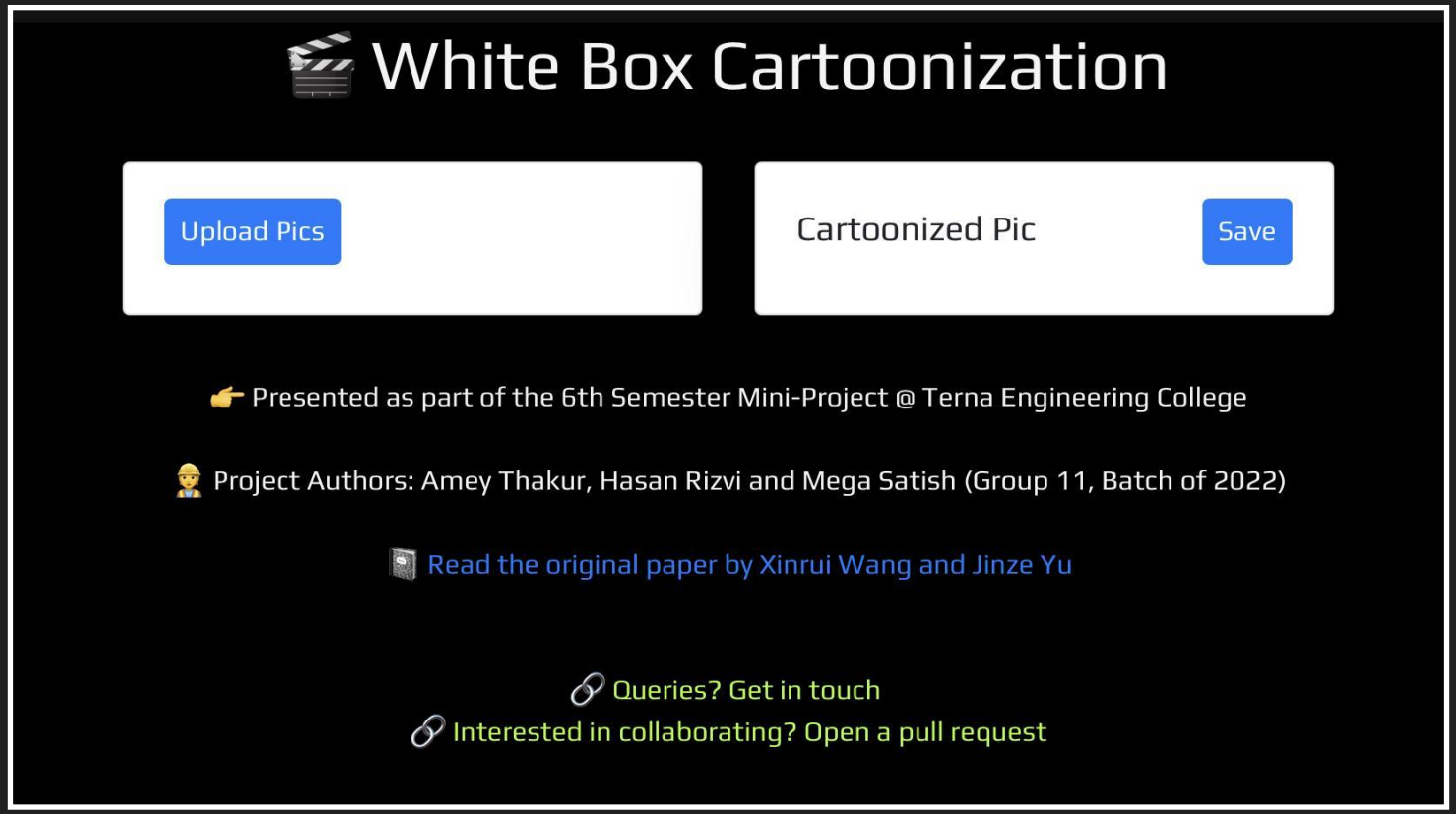
CartoonGAN

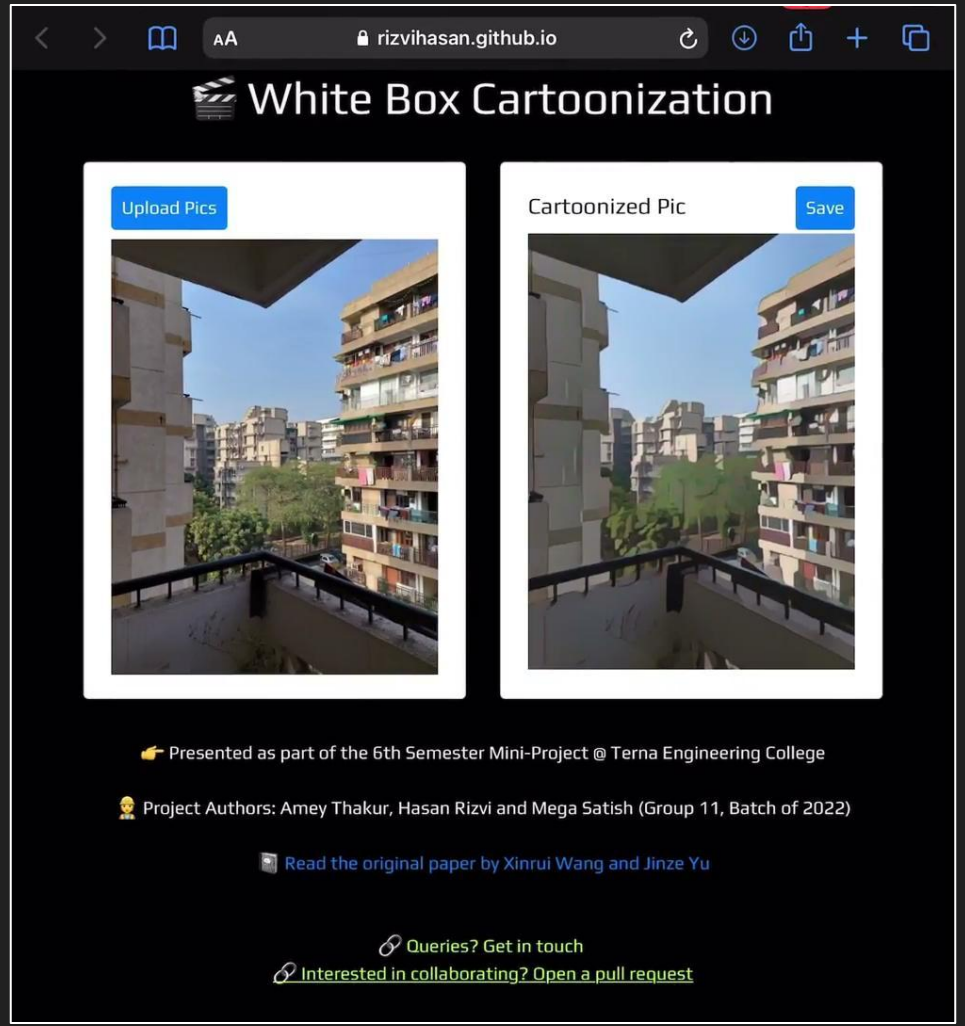
GANILLA

WBC

RESULTS

Web App - https://rizvihasan.github.io/whitebox_cartoonisation-webapp




AA
rizvihasan.github.io
White Box Cartoonization
Upload Pics
Cartoonized Pic
Save
Presented as part of the 6th Semester Mini-Project @ Terna Engineering College
Project Authors: Amey Thakur, Hasan Rizvi and Mega Satish (Group 11, Batch of 2022)
Read the original paper by Xinrui Wang and Jinze Yu
Queries? Get in touch
Interested in collaborating? Open a pull request

Cartoonized Image →

← Test Image

🎬 White Box Cartoonization

Upload Pics

Cartoonized Pic

Save

👉 Presented as part of the 6th Semester Mini-Project @ Terna Engineering College

👷 Project Authors: Amey Thakur, Hasan Rizvi and Mega Satish (Group 11, Batch of 2022)

Cartoonized Image →

← Test Image

🎬 White Box Cartoonization

Upload Pics

Cartoonized Pic

Save

👉 Presented as part of the 6th Semester Mini-Project @ Terna Engineering College

👷 Project Authors: Amey Thakur, Hasan Rizvi and Mega Satish (Group 11, Batch of 2022)

📓 Read the original paper by Xinrui Wang and Jinze Yu

🔗 Queries? Get in touch
🔗 Interested in collaborating? Open a pull request

Test Image

Cartoonized Image

🎬 White Box Cartoonization

Upload Pics

Cartoonized Pic

Save

👉 Presented as part of the 6th Semester Mini-Project @ Terna Engineering College

👷 Project Authors: Amey Thakur, Hasan Rizvi and Mega Satish (Group 11, Batch of 2022)

📓 Read the original paper by Xinrui Wang and Jinze Yu

🔗 Queries? Get in touch

🔗 Interested in collaborating? Open a pull request

Test Image

Cartoonized Image

CONCLUSION

- We have proposed a deployed white-box controllable image cartoonization framework based on GAN, which can generate high-quality cartoonized images from real-world photos.
- First, images are decomposed into three cartoon representations: the surface representation, the structure representation, and the texture representation. Corresponding image processing modules are used to extract three representations for network training. Finally, extensive quantitative and qualitative experiments have been conducted to validate the performance of our method.
- Our Web App is able to produce a cartoonized image from an input image. The output result is very similar to the real image yet depicts characteristics of cartoons. Also the time for conversion is less compared to the previous system and this web application works on any browser without having any specific requirements.

REFERENCES

- https://arxiv.org/abs/2002.05638
- https://arxiv.org/abs/1406.2661
- https://systemerrorwang.github.io/White-box-Cartoonization/
- https://www.researchgate.net/publication/329748755_CartoonGAN_Generative_Adversarial_Networks_for_Photo_Cartoonization
- https://www.researchgate.net/publication/343457093_Learning_to_Cartoonize_Using_White-Box_Cartoon_Representations
- https://www.tensorflow.org/js/models
- https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D

THANK YOU