Anime Scene Generator from Real-world Scenario using Generative Adversarial Networks

LE XUAN HUY
BUI THI BICH NGOC
SUPERVISOR: DR. PHAN DUY HUNG

Content

Introduction

This thesis presents an approach for image cartoonization and style transferring: translating an image or video in real life into an aesthetic, anime-like frame.

Original

After transfering

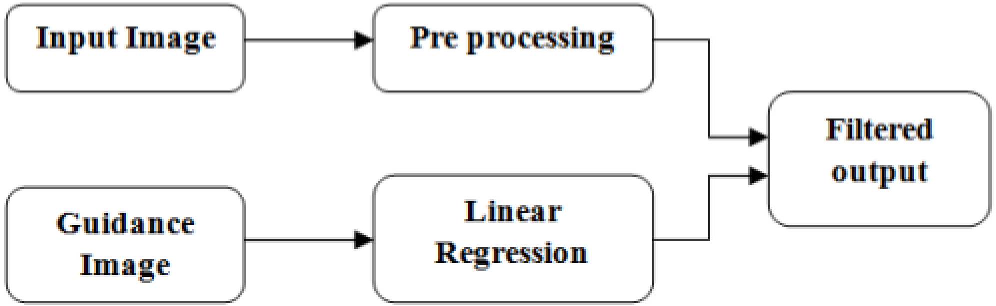
Project Objectives

- Reduce time needed to produce anime/cartoon episodes for artists and studio
- Build an entertainment/business web or mobile application of auto cartoonizing images and videos
- Contribute our results and studies to the image processing field and further research.

Related Works

Image Smoothing

Image smoothing is an image enhancement process, which is usually applied as one module of image preprocessing in various projects. Smoothing is often used to reduce noise in images and give us a more accurate intensity surface.



Guided Image Filter

Original image, guided filter result, and fast guided filter result

Related Works

Image Segmentation

Image segmentation aims to separate images into different regions. And superpixels were created to group pixels similar in color and other low-level properties. Various well-known segmentation and grouping methods exploit the power of superpixel algorithms to perform in a faster and more memory-efficient manner.

Outdoor & indoor scene segmentation results produced by Felzenszwalb's algorithm

Related Works

Generative Adversarial Networks (GANs)

Generative Adversarial Networks is a data generation method, which introduces adversarial training between the *G* - Generator and the *D* - Discriminator to achieve impressive results.

⇒ GANs is basically a min-max game between the *G* and *D* with a value function *V (G, D)*.

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

Character creation using GANs

Related Works

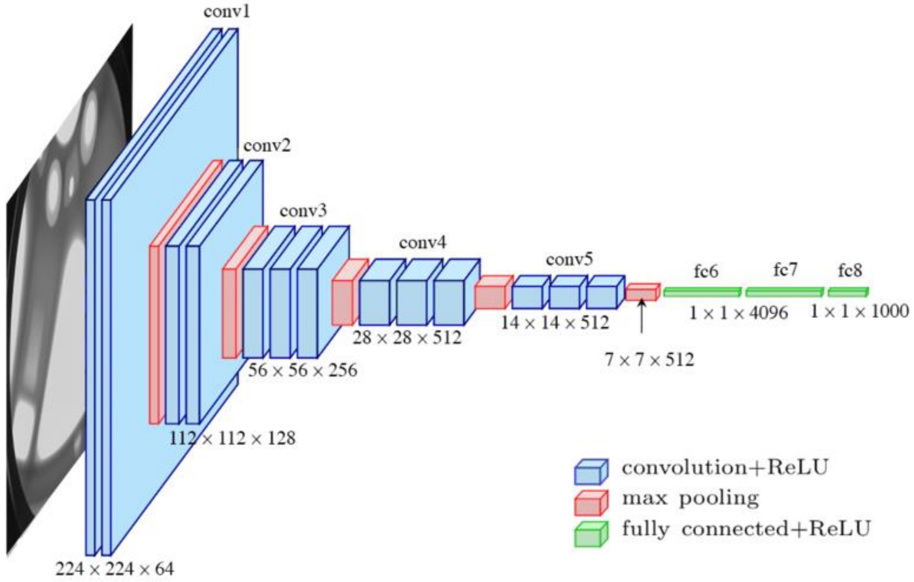
Non-Photorealistic Rendering (NPR)

Non-Photorealistic Rendering is a computer graphic area that focuses on various styles of digital art. NPR algorithms, especially Neural Style Transfer, have been developed to generate/translate images with different artistic styles, such as painting, drawing, animation, and architecture illustration, etc.


Original picture and non-photorealistic representation of a lake

Related Works

Recent studies on NST show that the VGG network trained for object recognition has the ability to extract semantic features of objects.

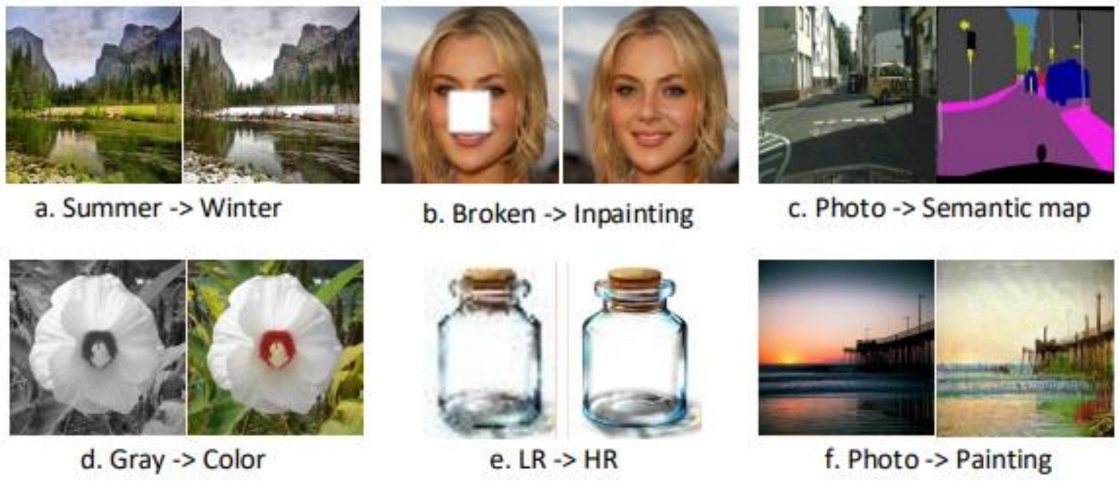


Architecture of a VGG-16 network

Related Works

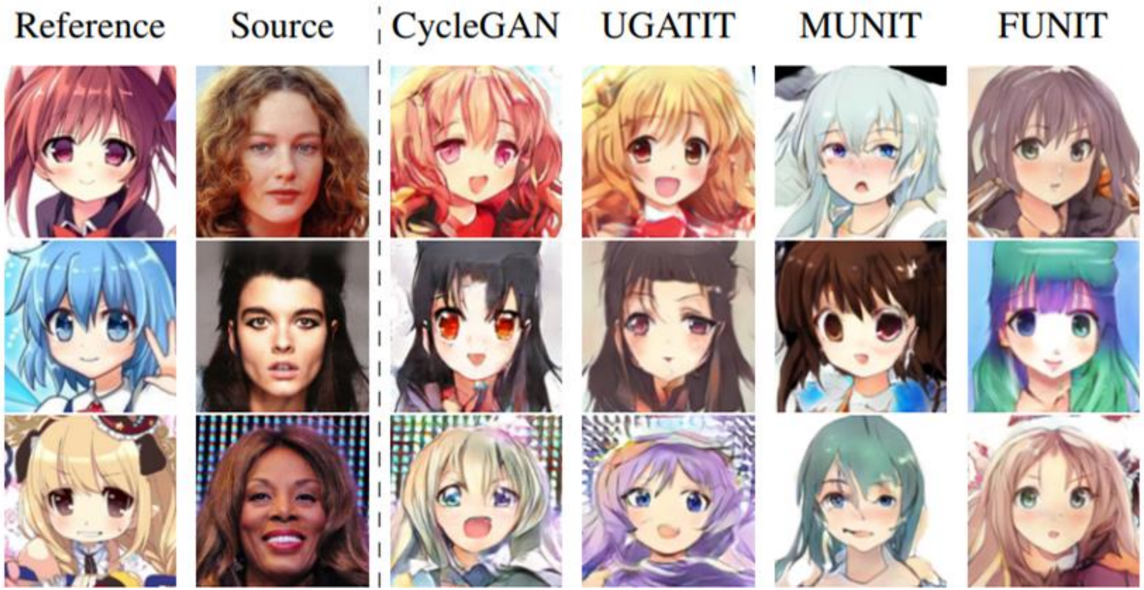
Unpaired Image-to-Image Translation

Image-to-Image Translation (I2I) focuses on translating images from a source domain to another target domain while preserving the content representation.



I2I applications in various graphic problems

Related Works

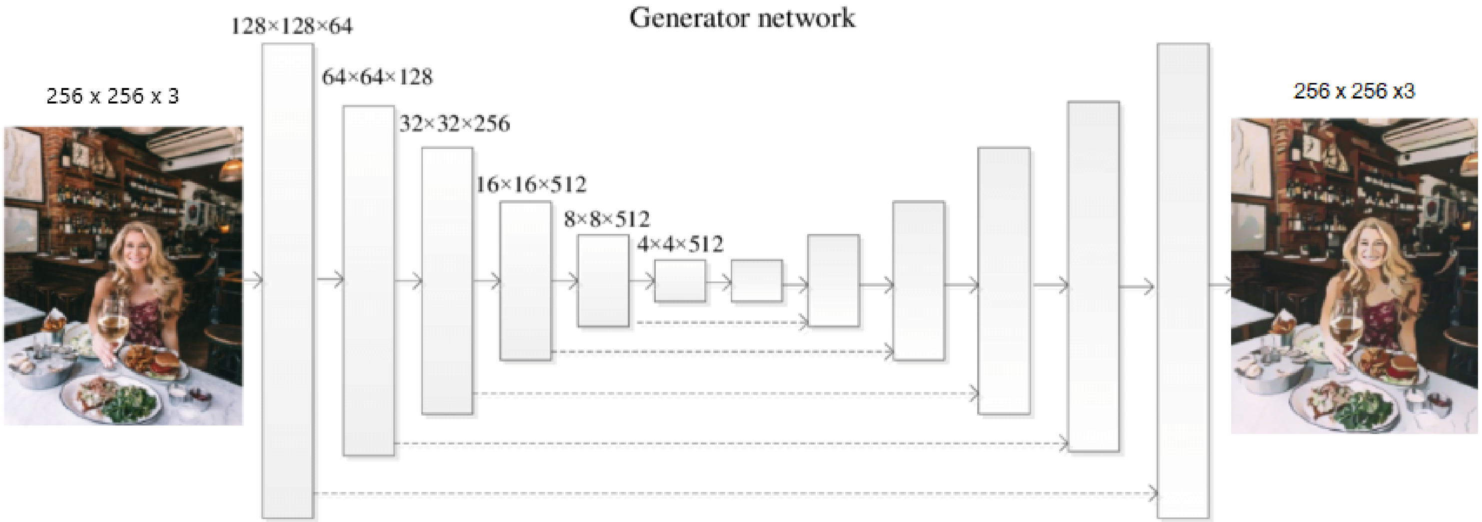


Performances of different models on the face2anime dataset

⇒ Many interesting and effective methods were proposed to solve various problems in style-transferring and image translation. However, some problems still require answers, such as unclean results caused by outliers, insufficient data, or poor style generalization caused by partial images segmentation of specific types.

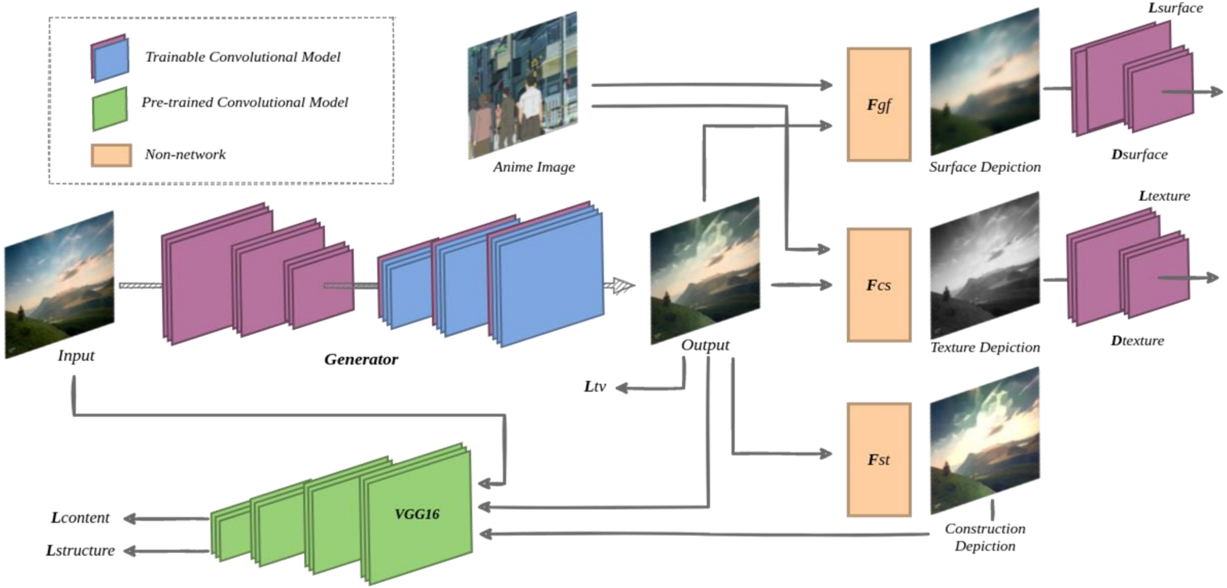
Model Architecture

Our Generator is a UNet-based generator capable of generating cartoon images in a short amount of time.



Model Architecture

After going through the Generator, images are decomposed into the surface depiction, the construction depiction, and the texture depiction. Three independent discriminators and losses are also proposed to extract information and guild the model to learn.



Structure Loss

The structure loss aims to imitate the animated style of clear edge, high-level simplification and abstraction, and sparse color blocks:

$$L_{structure} = \left\| VGG(G(I_x)) - VGG\left(F_{sr}(G(I_x))\right) \right\|$$

where

- A pre-trained *VGG-16* feature extractor as a structure discriminator
- F_{sr} be the extracted structure representation using Felzenszwalb segmentation and hierarchical grouping

Surface Loss

The surface loss will try to force the model to learn the cartoon painting style where artists usually draw drafts with coarse brushes and have smooth surfaces similar to cartoon images.

$$L_{surface} = \log D_s\left(F_{gf}(I_y, I_y)\right) + \log\left(1 - D_s\left(F_{gf}(G(I_x), G(I_x))\right)\right)$$

where

- F_{gf} is differentiable guided filter for edge-preserving filtering which will take an image as input and return a smooth, blur version
- A simple discriminator D_s is used to decide if the generated output has the same surface as the animated picture

Texture Loss

Help the model to re-create the unique characteristics with high-level simplification and high-frequency features of animated frames:

$$I_{grayscale} = \frac{\beta_1 \cdot I_r + \beta_2 \cdot I_b + \beta_3 \cdot I_g}{\beta_1 + \beta_2 + \beta_3}$$

where F_{cs} is a simple random color shift algorithm used to convert the image to a grayscale feature map that still contains information about all the high-frequency textures.

$$L_{texture} = \log D_t\left(F_{cs}(I_y)\right) + \log\left(1 - D_t\left(F_{cs}(G(I_x))\right)\right)$$

where D_t discriminator separates the grayscale feature map extracted from the generated and cartoon images.

Total-Variant Loss and Superpixel Loss

The total-variation loss L_{tv} is used to impose spatial smoothness on generated images and reduce high-frequency noises such as salt-and-pepper noise.

$$L_{tv} = \frac{1}{H \cdot W \cdot C} \left\| \nabla_x\left(G(I_x)\right) + \nabla_y\left(G(I_x)\right) \right\|$$

Superpixel loss L_{sp} to maintain important content from the input photo, which ensures that the cartoonized results and input photos are semantically unchanged. We also use a pre-trained VGG16 model to calculate it, similar to the structure loss:

$$L_{sp} = \left\| VGG\left(G(I_x)\right) - VGG(I_x) \right\|$$

Final Generator Loss

With all of the losses mentioned above, we can write our final loss function as:

$$L_{generator} = \beta_1 \cdot L_{tv} + \beta_2 \cdot L_{surface} + \beta_3 \cdot L_{structure} + \beta_4 \cdot L_{texture} + \beta_5 \cdot L_{sp}$$

where the parameter $\beta_1, \beta_2, \beta_3 \ldots$ can be changed for separate uses.

Training Experiments

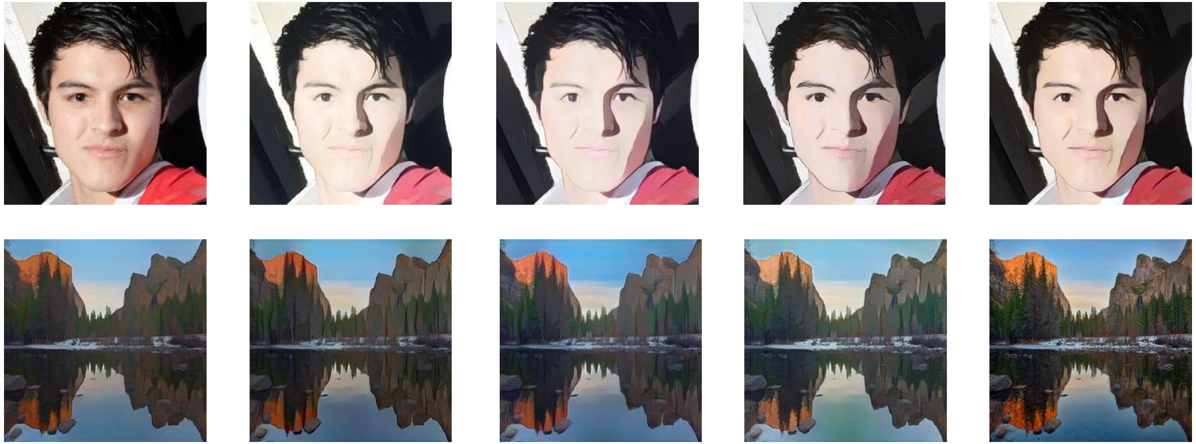
- GPU: NVIDIA 1060Ti GPU
- Data:
  - For the animation data, the study uses 10000 for scenery and 5000 for human faces
  - Also for real-word data, the study uses 10000 for scenery and 5000 for human faces




Dataset includes: scenery and face real images, scenery and face animation images

Training Experiments

Origin 5k 10k 15k 20k



- This GAN model is implemented in Tensorflow
- We use Adam algorithms with a learning rate of $1.5*10^{-4}$ and train the model with batch size 16 for 20000 iterations

Time Performance and Model Size

Table 1. Parameters and Time Performance comparison

Methods	AnimeGAN	CartoonGAN	CycleGAN	Ours
HR, GPU(ms)	45.53	148.02	106.82	15.23
Parameter(m)	3.96	11.38	11.13	1.48

Our method has a relatively low number of parameters and running time. On our GPU, we could reach the time of 17ms to process a 720*1280 image, which is much faster than other related works and can be totally capable of real-time high-resolution video processing tasks. Our model only has about 1.5 million parameters with the size of 5.6MB, which can be used to deploy on mobile apps.

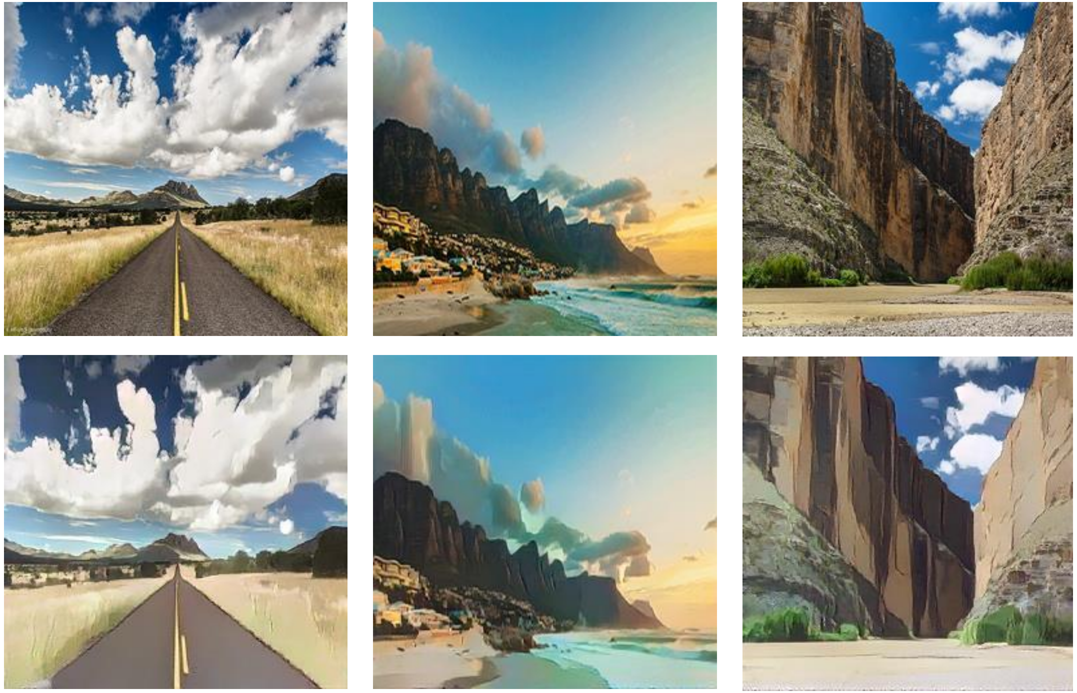
Results

Human and animal translated images

Results

Food and street images

Results



Outdoor scenery images

Quantitative Comparison

For qualitative evaluation, this paper use Frechet Inception Distance (FID) is proposed for quantitative evaluation to compare the generated images with the target images

Table 2: Performance evaluation based on FID

Method	**Real Photo**	**CartoonGAN**	**AnimeGan**	**WhiteBox**	**Ours**
FID to Cartoon	160	125	130	118	110

Qualitative Comparison

Origin *CartoonGAN* *AnimeGan* *WhiteBox* *Ours*

Illustration of Controllability

Input photo | More Texture | More Structure | More Surface

Analysis of Each Component

Original photo (a) W/O Texture Loss (b) W/O Structure Loss (c) W/O Surface Loss (d) Full

Conclusion & Future Works

This thesis proposes a lightweight and controllable approach for image cartoonization by translating actual footage into animation. We use GANs as our translating network, then pay close attention to the animation painting process and extract separate feature maps from generated pictures, and finally use different discriminators and losses to control the learning process.

In the future, we would like to extend the application of this method on real-time rendering to generate smooth, anime-like cuts. Details on portrait and facial expression also needs improving so that the character's emotion and sentiment would be more well-described.

Thank you for listening!

We would like to send our sincere gratitude to our supervisor, Dr. Phan Duy Hung for helping us with this thesis; our teachers, friends, and family who always support us; also the inspirational fellow researchers and artists who create those animation artworks.