# IMAGE CARTOONIZATION USING GAN

Sai Meghana Kolla

CMPSC 445 – Machine Learning in Applied Data Science

**Abstract.** In the aim to reduce the effort to build cartoons from scratch, developed a Generative Adversarial Network which can transform a real image to a cartoon depending on the style chosen by the user. The result would be a cartoonized version of the input real image.

## 1 Introduction

Cartoons are a form of art that has various applications. From entertainment to learning cartoons are used in various aspects of an individual life. However it takes a lot of creativity and effort to come up with a cartoon character or image. To convert a real image into a cartoon the artists should create everything from the base. To reduce this effort and make it easy to create cartoons, our application built on GAN can be used.

A generative adversarial network (GAN) is a machine learning (ML) model, where two neural network models compete to improve their prediction accuracy. GANs are usually unsupervised and learn using a cooperative zero-sum game framework. The generator goal is to generate outputs that might easily be interpret as actual data. The discriminator's purpose is to figure out which O/P generated was intentional.[1]

We've come a long way due to the technology breakthroughs and approaches for cartoonizing photographs. To make the greatest possible cartoonified image, each algorithm/technique came up with its own modifications and variations. One of the major successes of the time was a technique called data augmentation in deep learning's approach in the computer vision area. Data augmentation improves the performance of our models by enhancing model skill and delivering a regularized effect that reduces generalization error. It works by using the most recent, contrived, but plausible examples from the input issue area to train the model [2].

Utilizing GAN's for image analysis brings them in the context of machine learning and allows for a quantitative evaluation of the features extracted in unsupervised learning. Image synthesis is still a key feature of GAN, and it's especially beneficial when the created image is subject to limitations. Super-resolution is an example of how an established approach can be enhanced by adding an adversarial loss component to improve the quality of the output. Finally, picture-to-image translation demonstrates how GANs may be used to solve

a variety of problems involving the conversion of an input image to an output image [3].

Machine learning has a huge difficulty in terms of accessibility (ML). Typical machine learning models are created by experts and validated using specific hardware and software as well as ML experience. This makes it difficult for non-technical collaborators and end-users to provide feedback on model building and establish confidence in machine learning. Collaboration is also made more difficult by the accessibility issue, which limits the ML researcher's exposure to realistic data and events that occur in the real world. To promote accessibility and cooperation, Gradio, an open-source Python program that allows researchers to quickly create a visual interface for their machine learning models. Gradio makes it as simple as providing a URL to access any machine learning model [4].

### 1.1 Contributions

- My learning curve for the project is exponential. I have researched on various techniques used for image cartoonization and compared then and finally settled on to GAN.
- Researched on why GAN is best and understood the architecture of GAN.
- It took a considerable amount of time to understand the flow of cartoonGAN.
- Built UI where the user can input the image and select style of cartoonization using Gradio.
- I have also tried to extend it for multiple images.

## 2 Material and Methods

In this project, we utilized the capabilities of pre-trained cartoonGAN for image transformation. cartoonGAN consists of a generator and a discriminator. The image that should be transformed to a cartoon must be uploaded by the user in the web application and after clicking on submit the image will be passed through generator and discriminator and the cartoonized version of the uploaded image is displayed on the UI.

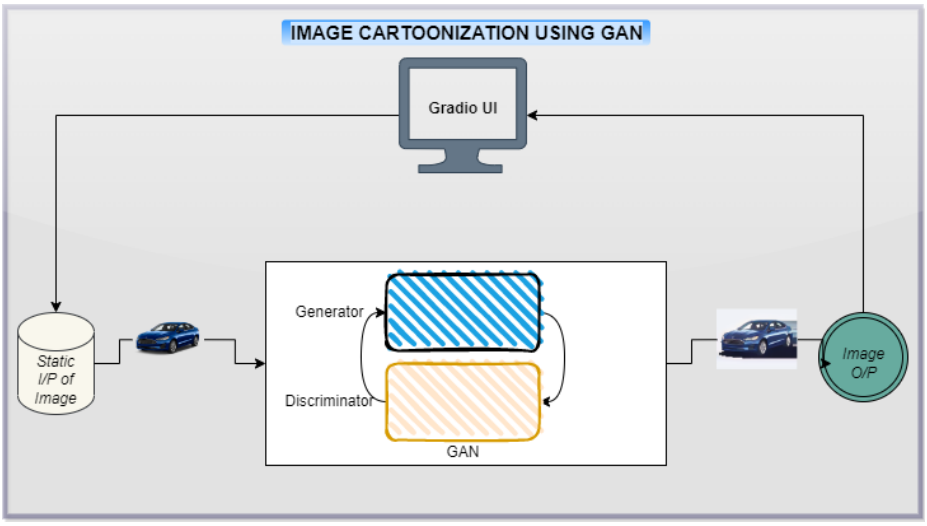


**Fig. 1.** Architecture of the application

### 2.1 Dataset Description

To train the GAN, image data set of real images and cartoon images for each style are used. 6,153 real images in which 5,402 are used for training and the remaining are used for testing. Cartoon images of different styles from different artists are used. 4,573 images for Shinkai style, 4,212 images for Hosoda style, 3617 images for Hayao style and 2,302 images for paprika style are used.[4]

### 2.2 Methodology

CartoonGAN consists of two convolutional neural networks, generator and discriminator.

Generator is responsible for transforming a real image uploaded by the user into a cartoon. After the generator is trained when an image is uploaded by the user the image is first down convoluted and then up convoluted. This process is used so that no information is lost while processing the image.

Discriminator accepts the image generated by the generator network and verify if the image is in the expected cartoon level. Since this is not a typical task, we use a regular patch - level discriminator.

### 2.3 Performance Evaluation

This performance evaluation is based on 900 test images, important metric for any ML application would be the inference time and average inference time for the GAN cartoonization application is less than 4 seconds.

## 3 Experimental analysis

Figure 2 displays the UI of the application where the user can input any image they want to transform and pick a style of cartoonization.

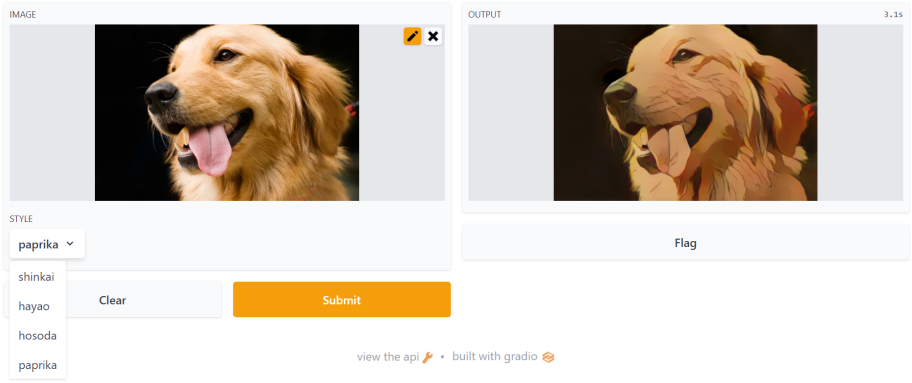


**Fig. 2.** Image transformation using Shinkai style

Below figures illustrates each style of cartoonization on a given real image. The picture on the left is the real image and the picture on the right is the cartoon version of the real image for the respective styles.

**Fig. 3.** Image transformation using Shinkai style

**Fig. 4.** Image transformation using Hayao style

**Fig. 5.** Image transformation using Hosoda style

**Fig. 6.** Image transformation using Paprika style

## 4 Conclusion

We implemented a Cartoon GAN, a Generative Adversarial Network, in this study to convert real-world pictures into high-quality cartoon style graphics. The goal of restoring realistic qualities of cartoon images while boosting adversarial loss for clear edges gives enough flexibility for smooth shading to be reproduced. To aid with convergence, we also offer a simple yet effective initialization phase. The results reveal that Cartoon GAN can learn a model that converts photos of real-world scenes into cartoon-style images with great quality and efficiency, exceeding state-of-the-art stylization approaches by a wide margin. In the future work, due to the importance of social-media short videos, would like to implement GAN cartoonization application to various video formats.

## References

1. Brownlee, Jason. "A Gentle Introduction to Generative Adversarial Networks (Gans)." Machine Learning Mastery, 19 July 2019, https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/.
2. Advancement of Techniques for Image Cartoonization, Aditi Lohomi.
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., amp; Zou, J. (2019, June 6). Gradio: Hassle-free sharing and testing of ML models in the wild. arXiv.org. Retrieved from https://arxiv.org/abs/1906.02569.