

## Image-To-Image Translation for Animation Using Opencv and Gan

Karnati Sai Shashank, L. Sridhara Rao

Department of Information Technology, J. B. Institute of Engineering and Technology, Hyderabad.

**\*Corresponding Author:**

Karnati Sai Shashank. Department of Information Technology, J. B. Institute of Engineering and Technology, Hyderabad.

**Submitted:** 30 Dec 2022; **Accepted:** 20 Jan 2023; **Published:** 21 Jan 2023

**Citation:** Shashank, K. S., Sridhara, L. R. (2023). Image-To-Image Translation for Animation Using Opencv and Gan. *J Math Techniques Comput Math*, 2(1), 30-34.

### Abstract

The main goal is to produce an output to convert the global real image into supported impressions (Animated Image). The concept of the paper is based on one-of-a-kind photos that are turned into an art shape akin to oil. This project would use an OpenCV package in Python to create the layout and a Deep Convolutional Generative Adversarial Network (DCGAN) to generate realistic-looking photos. The system is made up of three components: (a) a generative adversarial network that has been trained to generate comic characters, (b) a cartoon model that will generate edge animated images using OpenCV, and (c) an edge detector that takes input from portraits and then generates comic characters based on the resulting edge images.

**Keywords:** Generative Adversarial Network, OpenCV, Cartoon Image, Image Processing, Transfer Learning, Efficient Net.

### Introduction

A cartoon is a basic sketch of things we see around us in an amusing fashion with plenty of vibrant colours in it, or it is a film, movie, or short video with animation in it. Cartoons are mostly intended for children. They are typically published in newspapers, comic books, and periodicals, or aired on television. Previously, cartoons were just intended for fun. However, cartoons are now widely utilised for a variety of different reasons. Cartoons are being utilised for teaching and increasing awareness among the public, as well as for pleasure and amusement.

Kids are always looking for ways to have fun. They want to be surrounded by joy and comedy at all times. And its human nature to associate cartoons with enjoyment and levity [1]. As a result, it can be stated that when students are looking for fun, the first entity that will catch their attention will be cartoons or comic strips. Not only children, but even adults, engage in this behaviour. When we open a book, we glance at the graphics first, followed by the content. Similarly, the kids will read the comic strip first, followed by the text.

The newest TikTok filter literally converts you into an animated character. When the fad originally appeared in December 2020, you had to leave the app to make your own Frozen-worthy avatar (which you can still do), but TikTok just introduced animation effects, giving you a plethora of new possibilities. If you're wondering how to make a TikTok cartoon face filter appear like a Disney

character, there are a few options. To begin, you may make your digital mini-me directly in the TikTok app, or you can go to Snapchat or the Voilà AI Artist app [2]. Despite the fact that none of these applications are officially affiliated with Disney, the cartoon vibes will remind you of Disney masterpieces. In this paper, an introductory programmed strategy for objects birth from vitality picture is proposed it's grounded on the dubitation that the demanded item is established inside or near the focal piece of photo.

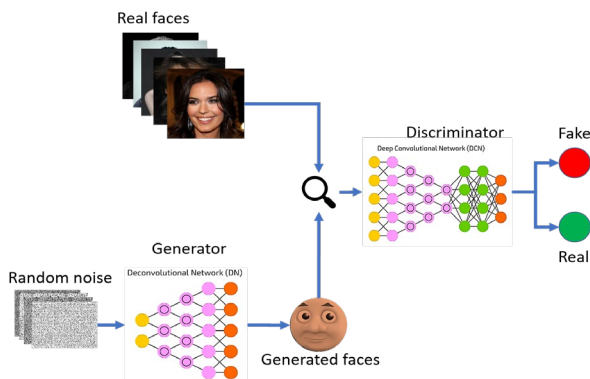
### Literature Review

GANs are a game changer in the field of Artificial Intelligence because they are extremely effective at generating realistic new data instances that closely mirror your training-data distribution. They're teaching robots to thrive in human-like activities like writing, painting, and music. The concept of GANs has completely transformed the generative modelling area. GANs are Neural Networks that accept random noise as input and produce outputs (e.g., an image of a human face) that appear to be a sampling from the training set's distribution (e.g., a set of other human faces) [3]. It is an artist who creates realistic graphics from scratch. And, like many great artists, it feels the need for a mentor in order to advance. As a result, a GAN consists of: a) An artist, i.e., the Generator b) a mentor, i.e., the Discriminator.

Ian Goodfellow et al. of Université de Montréal comms@path.cam.ac.uk, who first published a paper on Generative Adversarial Networks in 2014, at the NIPS conference [4]. He presented GAN

as a novel framework for estimating generative models using an adversarial process in which a generative model  $G$  records the data distribution and a discriminative model  $D$  predicts whether the sample originated from the training data or  $G$ . The Discriminator assists the Generator in producing realistic pictures from noise. GAN pits two adversaries against each other: the generator and the discriminator [5]. a) By using random noise as input and creating realistic-looking pictures, the generator attempts to understand the data distribution. b) On the other hand, the discriminator attempts to determine if the sample came from the original dataset or is a forgery (generated by the generator). GAN can represent more complicated distributions if they are enforced, but they suffer from mode collapse. However, we discovered that completely linked layers reduced the quality of produced pictures while implementing this vanilla GAN.

Convolutional layers learn hierarchical characteristics while keeping spatial patterns, but fully linked layers lose the intrinsic spatial structure found in pictures. DCGAN stands



**Figure 1:** Flow Diagram of the GAN

for Deep Convolutional Generative Adversarial Network. The use of convolutional layers in this new design dramatically increases the quality of GANs. In 2016, a group of authors led by Alec Radford [alec.radford@gmail.com](mailto:alec.radford@gmail.com), published a paper at the ICLR conference, named Unsupervised representation learning with DCGAN. DCGAN generated higher quality images by using stride convolutional layers in the discriminator to down sampling the images and using fractionally-stride convolutional layers to up sampling the images [6].

All of them had a generator and a discriminator that trained in tandem. In every GAN, the generator is fed a random-noise vector, while the discriminator is fed actual or false pictures that are either conditioned on class labels or are not [7]. In 2017, a group of scholars led by Phillip Isola [phillip.isola@gmail.com](mailto:phillip.isola@gmail.com), at Berkeley AI Research (BAIR) published the paper titled Image-to-Image Translation with Conditional Adversarial Networks. It departs from the notion of supplying a random-noise vector to the gen-

erator and adds numerous substantial architectural modifications although borrowing heavily from prior GAN algorithms [8]. The goal of Image-to-Image Translation is to develop a mapping between the input and output pictures using a training dataset of aligned or unaligned cross-domain image pairings.

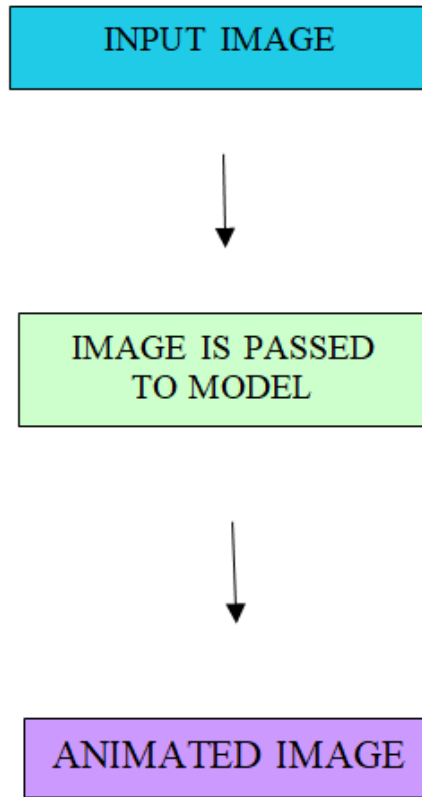
Later Style GAN proposed in 2019 by NV Labs [info@nvlabs.eu](mailto:info@nvlabs.eu) showed amazing performance in creating realistic images based on a style-based generator architecture by separating high-level attributes such as pose and facial expressions from stochastic variation in the images like hair and freckles [9]. Also, it aims to spread out the latent space by introducing an intermediate latent space which the generator infers from instead of the classic latent vector. Alias Freegan (stylegan3) proposed in 2022 aliasing is particularly noticeable when creating rotations of a given image [10]. When the pixels look like they were “glued” to some specific places of the image they do not rotate in a natural way [11]. In this we are generating faces new non-existing face in world but we are not styling our own face [12]. Style GAN provides numerous generator component modifications, allowing it to make lifelike high-quality images while also altering portions of the generator section. Style GAN is built on the baseline progressive GAN architecture, with many generator tweaks.

## Proposed Methodology

### Methodology

The GAN algorithm, which stands for "Generative Adversarial Networks," is employed for execution. GAN is a hybrid of a generative model and an AI model for discriminating. The generative model generates new data cases based on the training data. The discriminator is the model that is used to test the data that is being captured and compare it to the images generated by the Generator. The discriminator determines if the affair image is real or phoney. Both the generator and the discriminator are neural networks that compete with each other throughout the training phase. The way it happens several times so that the creator and discriminator acquire a better outcome once the way is repeated. In terms of design, around 276590 images from Kaggle were used to train the model (High-Resolution Anime Face Dataset). and a random image from FFHQ is chosen (Flickr-Faces-HQ Dataset). In comparison to the former style GAN, they have training that incorporates latent dim with random numbers, but in our design, we are training generator on real-world photos collected with a camera. In GAN, the generator is trained on random pictures from the FFHQ dataset and the discriminator is trained on the Anime Face dataset using the Efficient Net model with transfer learning to build a cartoon model. The latter discriminator is trained on actual photos as well as Edge animated images generated by OpenCV and recorded with the aid of a camera. The final classification of a cartoon model image determines whether the image is false or real.

Image conversion is mostly portrait to:

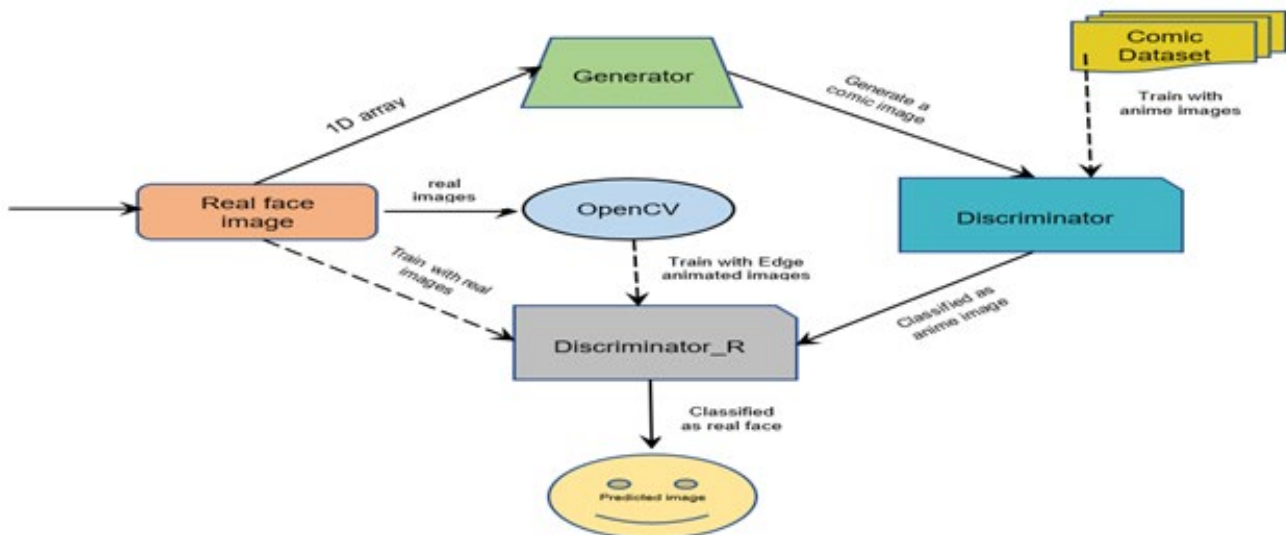


**Figure 2:** Flow Diagram of the proposed model

Creating an animated picture from regular bone. The videotape is first broken into frames and saved in an array containing photographs of the user, which is then handled by a discriminator to train and a generator to produce images using OpenCV.

**Basic Workflow**

The workflow of our entire model is briefly described, as are some of the primary tools and technologies used in our model, are briefly outlined in fig 3.



**Figure 3:** Data Flow Diagram of the proposed model

**Generator**

GAN's generator is a neural network that, given a random collection of variables, performs a sequence of non-linear calculations

to generate realistic-looking pictures. When fed a random vector  $Z$  taken from a genuine picture, the generator generates phoney images  $X_{fake}$ . We will be up-sampling the picture using Con-

v2DTranspose, followed by a Tanh function. In this stage of the algorithm, we used batch normalisation and dropout. We were able to greatly speed our model by adding several more layers and employing batch normalisation. In this example, we used Leaky Relu as the activation function. The generator model is used to update the discriminator model, and vice versa. As our performance optimizer, we used the Adam optimizer.

### Discriminator

In GAN, a discriminator is a neural network that classifies various classes in a dataset using class-specific labels. Thus, it is analogous to a supervised-classification issue in essence. As a classifier, we are employing the EfficientNet-B0 model with transfer learning. The discriminator will be given a single input to compare. The discriminator will be updated instantly. In order for the classification to be effective, both the input and target images must be classed as real. We used the Adam optimizer to enhance our system.

### OpenCV

OpenCV is a large open-source library for computer vision, machine learning, and image processing that is currently used in real-time operations. It can analyse photos and movies to recognise items, people, and even human handwriting. Python can process the OpenCV array structure for analysis when combined with other libraries such as NumPy. We are animating a real image with various CV2 functions and feeding it into a discriminator to learn animated faces to categorise. Every time, I get a different set of results.

### Experimental Analysis

As a result of the experiments performed on the FFHQ dataset, which comprises numerous facial photos. We modified the amount of the data numerous times to evaluate how the model works under different scenarios.

### Dataset

In this research, we will be utilising two datasets. HRAF is a dataset from Kaggle (High-Resolution Anime Face Dataset) and FFHQ (Flickr-Faces-HQ Dataset). We are training a discriminator to classify anime images using the HRAF dataset, and we are picking random images from the FFHQ dataset for testing.

### Tools and Environment for Experimentation

Our model has been tested in a variety of scenarios and with a variety of tools to guarantee that we obtain exceptional results from our data and model.

### Programming Language

It was created with the aid of the TensorFlow and Keras libraries from the Python programming language, both of which were used in the creation of our model. Python is now the most popular programming language for implementing Machine Learning and Deep Learning models, owing to the enormous number of packages available for it, which are particularly beneficial for implementing the numerous accessible Machine Learning and Deep Learning models. Python is also the most extensively used programming language since it is simple to learn and apply. In the Python programming language, there are several free and open-source libraries for Machine Learning and Deep Learning models. One of these technologies is TensorFlow. Keras, a free open-source software application, provides a Python interface for artificial neural networks. Keras acts as a bridge between the TensorFlow library and the outside world.

### Result

When we originally started, we ran our model across the whole dataset. To acquire a good outcome, we had to run the model roughly 15 times. Exercising the model for a larger number of steps yields a suitable result. This requires the fulfilment of a large number of phases. Because generative adversarial models generate new images dependent on the task at hand, they are referred to as adversarial models.



Figure 4: Predicted Images

### EfficientNet

The EfficientNet models outperform previous CNNs in terms of accuracy and efficiency, lowering parameter size and FLOPS by an order of magnitude.

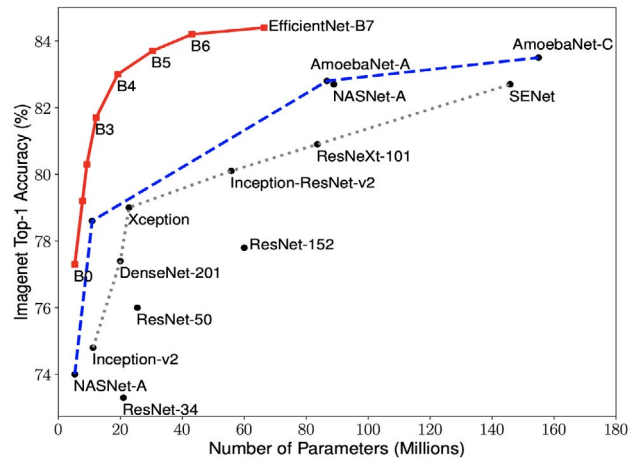


Figure 5: EfficientNet Performance

### Conclusion

Our research aims to enhance image-to-image translation by eliminating random values as generator input. In addition, with the help of CV2, also known as Computer inventive and prescient, the project transformed videotape into electricity (cartooned image). This project demonstrated the conversion of a picture to a cartoon image using GAN and the Python library OpenCV. It was forecasting an anime image of a real face with excellent quality. This is the magic of OpenCV creating edge animated pictures and assisting in training discriminators to create correctly classified human animated images.

### References

1. Kumar, S., Bhardwaj, U., & Poongodi, T. (2022, April). Cartoonify an Image using Open CV in Python. In 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM) (pp. 952-955). IEEE.
2. de Lima, C. R., & Beltes, J. (2021, September). Deep Learning Image-To-Image Translation for Animation. In 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) (pp. 1-2). IEEE.
3. Plabon, S. S., Khan, M. S., Khaliluzzaman, M., & Islam, M. R. (2022, February). Image Translator: An Unsupervised Image-to-Image Translation Approach using GAN. In 2022 International Conference on Innovations in Science, Engineering and Technology (ICISSET) (pp. 338-343). IEEE.

4. Aditya Sharma (2021) Introduction to GANs.
5. Altakrouri, S., Usman, S. B., Ahmad, N. B., Justinia, T., & Noor, N. M. (2021, September). Image to Image Translation Networks using Perceptual Adversarial Loss Function. In 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA) (pp. 89-94). IEEE.
6. Aditya Sharma (2021) Introduction to DGANs.
7. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).
8. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
9. Nv Labs (2021) Style GAN TensorFlow implementation.
10. Baggio, D. L. (2012). Mastering OpenCV with practical computer vision projects. Packt Publishing Ltd.
11. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
12. Maier, A., Syben, C., Lasser, T., & Riess, C. (2019). A gentle introduction to deep learning in medical image processing. Zeitschrift für Medizinische Physik, 29(2), 86-101.

**Copyright:** ©2023: Karnati Sai Shashank. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.