International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 05 | May 2022 www.irjet.net p-ISSN: 2395-0072

Photo Editing And Sharing Web Application With AI-Assisted Features

Kaustubh Jangde[1], Vishwajeet Ohal[2], Kaustubh Chaudhari[3], Anuja Aher[4]

[1-4]Student, Dept. of Computer Engineering, Sinhgad College of Engineering, Maharashtra, India

---***---

Abstract - *In the 21st century, images play a crucial role in the media. Applying filters and customizing images to one's desires is commonplace and the need for image manipulation tools has been on the rise for the last decade. We try to build an application using state-of-the-art Machine Learning technology of Generative Adversarial Networks (GANs). GANs have been proven to reduce human efforts for image manipulation and are one of the most suitable techniques for the task today. In addition, GANs provide image generation with unparalleled resolution and high fidelity. The proposed application makes use of this methodology to establish real-time image manipulation on a wide array of platforms using inputs from the user. Since most editing applications require some professional skills for editing an image, we try to make the editing process hassle-free with our AI-assisted features. Furthermore, the application allows users to share images on multiple platforms including our own.*

Key Words: **Generative adversarial network, StyleGAN, deep learning, image editing.**

1. INTRODUCTION

The system has been developed using Python-Django Web Framework. This makes the system highly scalable, and versatile in nature, which promotes time-efficient development and clean, practical design architecture. The system has been thoroughly tested which makes it durable and powerful enough to withstand the dynamic changes. The database used is SQLite, which, being an integral part of Django, provides high-end support for python. It is also secure, reliable, and powerful. The frontend is developed using HTML, JavaScript for making the UI interactive, and CSS to style the webpages. It also includes Bootstrap and jQuery third-party libraries to make the frontend responsive.

2. METHODOLOGY

2.1 AI-Assisted Editing

The Editing module of the application is divided into two main parts, viz. AI-Assisted Editing and Manual Editing, the former of which forms the crux of the application. To establish semantic editing, the recently developed deep learning algorithm of GANs has been used.

2.2 Image Transformation using GANs

The task of image generation is carried out best by using generative networks. Generative networks can be further classified into two types, viz. Auto Encoders and GANs. Recent literature has shown that GANs have an upper edge in performing image manipulation, especially when dealing with semantic feature editing in various scenarios.

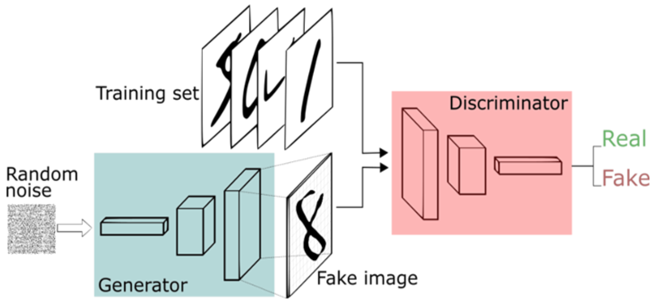

Fig -1: GAN Architecture

2.3 GAN Working

In GANs, the generator and discriminator are trained in an adversarial setting where each tries to perform better than the other. The discriminator is tasked with identifying whether a given image is real or fake. The generator attempts to make images such that the discriminator wouldn't be confident to differentiate between an artificial and real image.

Initially, the generator and discriminator are initialized with random weights and are in an unlearned state. To train the discriminator, a batch of real images and images generated by the generator is passed to it. Using these, the discriminator calculates a probability for each image provided to it belonging to either the real class or fake class during the forward propagation phase. During backward propagation, the discriminator network calculates loss by using the provided labels. The generator later updates its weights according to the magnitude and the sign of the loss.

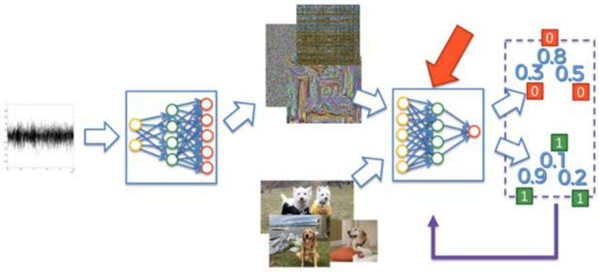

Fig -2: Discriminator training

To provide the generator with synthetic images, the generator samples input from a random distribution such as normal Gaussian distribution. At first, it produces images full of random noise. But as the generator learns by taking feedback from the output of the discriminator network, it starts generating lifelike images.

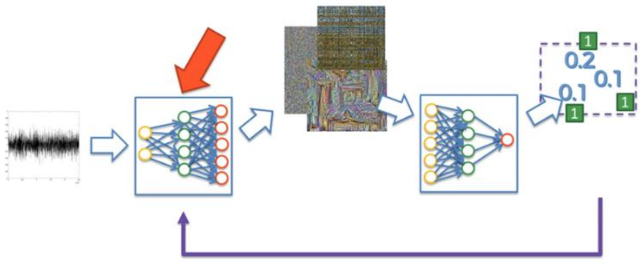

Fig -3: Generator training

2.4 StyleGAN

StyleGAN is a revolutionary approach used in GANs proposed by the researchers at Nvidia. It takes inspiration from and leverages the power of transfer learning in GAN. Its objective is to learn the style characteristics from an input image and apply the same styles to another image.

In StyleGAN, the dataset is firstly normalized in order to eliminate the problem of non-uniformity in data which can cause the model to generate a distorted latent space. While querying the generator, we sample from a random latent distribution z and normalize it.

Then, a mapping network of eight fully connected layers maps the input into a latent space w. A synthesis network *g* uses this latent space to upscale the output vector. This synthesis network is built up of multiple blocks of increasing size Each block of the synthesis network takes input from the latent space w and adds some noise to avoid overfitting. At the end of the 8th block, a vector of size 1024 x 1024 is generated.

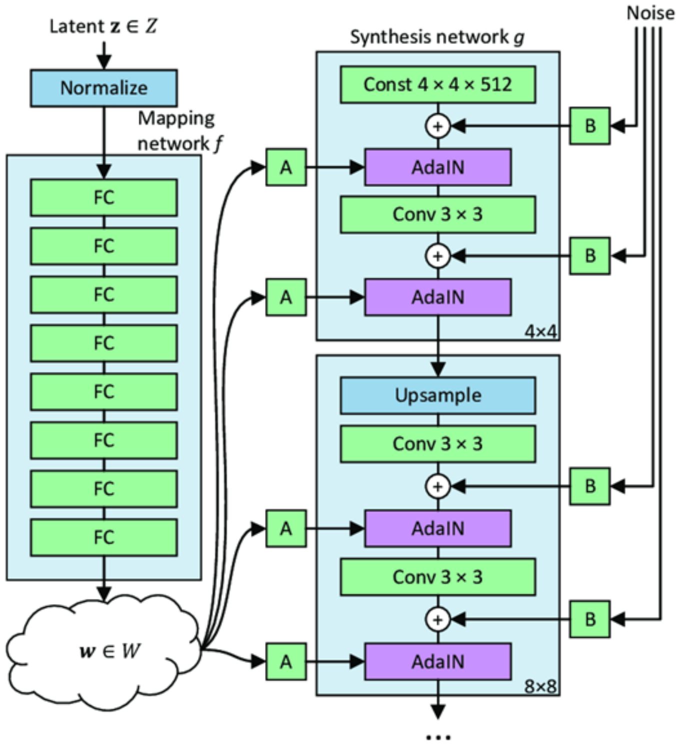

Fig -3: StyleGAN Architecture

2.5 StyleGAN Encoding

At the end of the generator's training process, the latent space has learned the probability distribution of training data and the generator can replicate the distribution effectively. In an ideally disentangled latent space, the feature correlation is minimal, and moving in a certain direction in latent space affects only a single attribute. But in practice, having such a one-to-one correspondence is unlikely.

In our application, we try to encode the target image (i.e image under consideration for editing) into the latent space as close as possible. Based on the encodings, we use InterfaceGAN to walk in different latent directions to gain the desired edit. This encoding is done using a VGG network, which is used to compare the image generated by the generator and the target image on the feature level. The difference is calculated and backpropagated to modify the initial latent vector used to generate the image. After multiple iterations, we are able to achieve a latent vector that generates an image resembling closely to our target image.

To speed up the encoding process, we use a ResNet trained along with the StyleGAN to output a latent vector given an image. This vector gives us a starting point to start optimizing the latent code for our target image.

2.6 Editing the Latent Vector

As a result of the encoding process, we get a latent vector that outputs a closely resembling target image. By operating on this vector, we intend to achieve the desired edit.

We make use of the fact that, in a disentangled latent space, there exists a hyperplane such that, moving along the normal to the hyperplane in one direction affects a feature of the image in one manner. In addition, moving in the opposite direction will bring about a change that is exactly in contrast with the prior feature modification.

2.7 General Applications

Generative Adversarial Networks are popular for being applicable to a wide range of problems. Its characteristics of being able to generate a particular kind of image and enabling modifications make it particularly useful in the following application

(1) Text-to-Image Translation
(2) Image-to-Image Translation
(3) Image Quality Enhancement
(4) Human Pose Generation
(5) Image Inpainting.

3. RESULTS

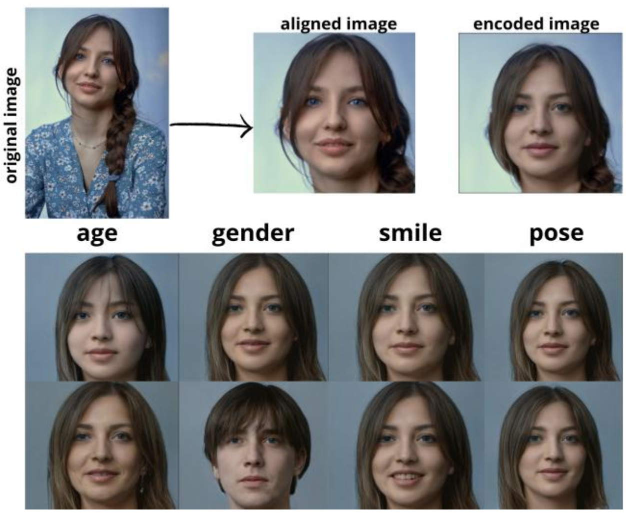

Fig -4: AI-Assisted Editing.

The original image is aligned and encoded in latent space. Corresponding changes can be seen in Fig-4. Age, gender, smile, and pose edits are made with great fidelity. These images produced are of high resolution and high fidelity.

4. FUTURE SCOPE

4.1 Image Fidelity

The used machine learning models currently fail to give an accurate output when presented with extreme parameters. Therefore, further research is needed to gain precise control over image feature manipulation without sacrificing image quality

4.2 Use of DevOps to automate software development pipeline

DevOps is a methodology used for automation of the software development lifecycle. By adopting this method, we can reduce the software delivery time so that new features can be deployed quickly in the market. MLOps, which is a sub-branch of DevOps, can be used to automate machine learning processes. After adopting MLOps, we would be able to train our models quickly and in a time-efficient manner, while avoiding human errors.

4.3 Use of Graph Databases

Graph databases are very commonly used in multi-user environments and as the system has a multi-user environment for the community to share their creations with each other, we can replace SQLite databases with Graph databases. There are many graph databases available that can be used in the system. For example, Neo4j, or Nebula Graph.

4.4 Use of Microservices Architecture

Microservices can be used to isolate each component of the application and deploy them separately while integrating them on the same webpage. This would offer fault tolerance in such a way that even if one particular component fails, the operation of the rest of the application would not be affected.

3. CONCLUSIONS

A robust and versatile system for image editing with not only basic but also advanced photo editing options that work semantically rather than just on pixel values has been developed. Specifically, the user is allowed to manipulate features like age, pose, smile, etc. semantically without needing to edit pixels manually. The system makes use of the latest architectures of GANs and provides a one-stop-shop solution for a user's editing needs.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Mahesh P. Wankhade, Head of Department, Department of Computer Engineering, SCOE, for his continuous support and guidance throughout the development and research process.

REFERENCES

[1] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in CVPR, 2019, pp. 4401–4410.

[2] Tero Karras, , Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and Improving the Image Quality of StyleGAN." in CVPR, 2020

[3] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans, 2020

[4] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "Gan inversion: A survey," arXiv preprint arXiv: 2101.05278, 2021.

[5] Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., & Cohen-Or, D. (2021). Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

BIOGRAPHIES

Kaustubh Jangde,
Sinhgad College of Engineering

Vishwajeet Ohal,
Sinhgad College of Engineering

Kaustubh Chaudhari,
Sinhgad College of Engineering

Anuja Aher,
Sinhgad College of Engineering