Human Facial Generation Using Various Types Of GANS

Aamera Shaikh, Rahul Solanki and Ganesh Gude

Fall 2023 - INFO - H518 Faculty - Sunandan Chakraborty, *Ph.D.* SOIC, IUPUI - Indianapolis

Introduction

The human face, with its intricate details and nuanced expressions, remains one of the most challenging frontiers in artificial intelligence. Capturing its full complexity and generating realistic, high-fidelity faces has long been a tantalizing goal. Enter Generative Adversarial Networks (GANs), a revolutionary approach that has brought us closer than ever to achieving this dream.

This project delves into the fascinating world of GANs, specifically focusing on their application in generating realistic human faces. We will explore different GAN architectures, each with its unique strengths and weaknesses, and delve into the underlying mechanisms that drive their functionality. Through experimentation and comparison, we will gain a deeper understanding of how these models work and the factors that contribute to their success in producing lifelike faces.

Problem Description

In the captivating realm of Generative Adversarial Networks (GANs), our project embarks on an exploration of their application in crafting realistic human faces. Our focus lies in dissecting various GAN architectures, each endowed with distinctive strengths and weaknesses, while delving into the intricate mechanisms underpinning their functionality. Through a meticulous process of experimentation and comparison, our endeavor aims to unravel the mysteries of how these models operate and discern the pivotal factors influencing their prowess in generating lifelike faces. Despite the significant strides made in this domain, challenges persist, ranging from issues of diversity in facial features to ethical considerations surrounding deepfake technology. By immersing ourselves in this multifaceted investigation, we seek not only to enhance our understanding of GANs but also to contribute valuable insights that may pave the way for more robust and responsible applications of facial generation in diverse fields.

Data Description

Our primary data source for this project is the Celeb dataset, which we obtained from Kaggle, a platform renowned for hosting diverse datasets for machine learning and computer vision applications. This dataset comprises approximately 203,000 JPEG images, each featuring faces of celebrities from various backgrounds, professions, and ethnicities. The comprehensive nature of the Celeb dataset provides a rich repository for training and evaluating our Generative Adversarial Networks (GANs) in the task of human facial generation. The vast array of facial expressions, poses, and lighting conditions within the dataset enables us to explore the nuances of facial features and expressions, enhancing the diversity and realism of the generated faces.

In addition to the Celeb dataset, we have integrated two supplementary data sources to augment our training corpus. The first supporting dataset consists of images capturing landscapes and scenery, introducing an element of environmental context to our models. The second supporting dataset encompasses portraits from the medieval period, contributing historical diversity to our training data. These additional sources aim to enhance the versatility of our GANs by exposing them to a broader spectrum of visual information. The amalgamation of these datasets creates a holistic training environment, enabling our models to learn and synthesize human facial features in the context of diverse backgrounds, landscapes, and historical settings. Through this multi-faceted approach, we aspire to elevate the quality and creativity of the facial generation process.

Methodology

Basic GAN

In the Basic GAN, the generator and discriminator networks engage in a dynamic competition. The generator initiates the process by producing synthetic data from random noise, challenging the discriminator to distinguish between real and fake data. Both networks iteratively refine their capabilities through a feedback loop, leading to a generator that excels at crafting data indistinguishable from real samples.

Style GAN

Transitioning to the Style GAN framework, we leverage the baseline Progressive Growing GAN architecture. This entails a gradual increase in image resolution from 4×4 to 1024×1024, promoting stability by incorporating new blocks to support larger resolutions. Notably, bi-linear sampling replaces nearest neighbor up/down sampling in both the generator and discriminator, enhancing the overall quality of the generated images. The inclusion of a Mapping Network and Style Network further refines the process. The Mapping Network transforms the input latent vector into an intermediate vector, influencing distinct visual features through an 8-layer MLP. This intermediate vector undergoes a learned affine transformation before entering the synthesis network, which utilizes the Adaptive Instance Normalization (AdaIN) module to convert the encoded mapping into the final generated image. The combination of these techniques results in a Style GAN model capable of generating high-quality, diverse facial images with nuanced visual features.

Super Resolution GAN

Our approach centers on the Super Resolution GAN (SR GAN) architecture, designed with the primary objective of preserving and enhancing finer textures during image upscaling without compromising overall quality. The architecture comprises two key components: the Generator and the Discriminator. The Generator, responsible for producing data based on a probability distribution, utilizes a residual network instead of deep convolution networks. This choice is informed by the ease of training and the ability to achieve greater depth, facilitated by skip connections within the residual network. These skip connections enable effective information flow through the network, contributing to the generation of high-quality images. Meanwhile, the Discriminator assesses whether the input data comes from the original dataset or the Generator, engaging in a competitive process that encourages the Generator to optimize its output and deceive the Discriminator. This methodology ensures the recovery of intricate details and the preservation of image quality during the upscaling process.

Cycle GAN

Our study employs the Cycle GAN architecture, a powerful framework designed for unpaired image-to-image translation tasks. Comprising two generators and two discriminators, the Cycle GAN operates on the principle of cyclic consistency, ensuring the translated images can be reverted to their original domains. The generators, denoted as A and B, are responsible for mapping images from one domain to another and vice versa, while discriminators assess the authenticity of the translated images. The adversarial training process involves both generators attempting to fool their corresponding discriminators, while the cycle consistency loss enforces the reconstruction of the original input from the translated output. This dual-generator approach enables effective style transfer between domains without the need for paired data, making Cycle GAN a versatile tool for tasks such as image style transfer, object transfiguration, and domain adaptation**.**

Results

Fig 1: BASIC GAN

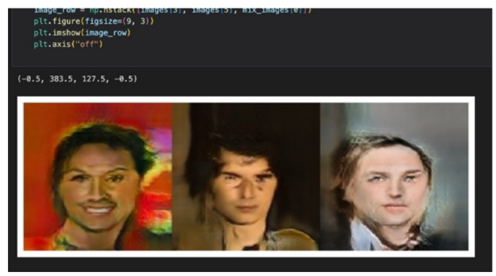

Fig 2: STYLE GAN

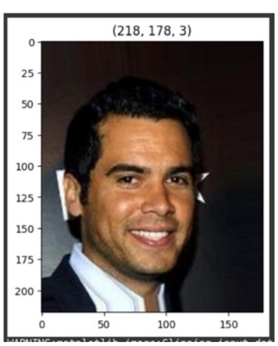

Fig 3: SR GAN (INPUT IMAGE)

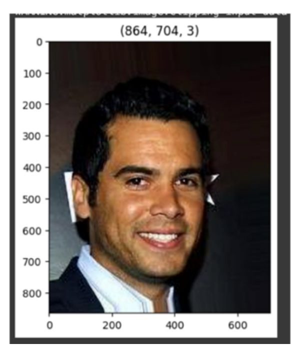

Fig 4: SR GAN (OUTPUT IMAGE)

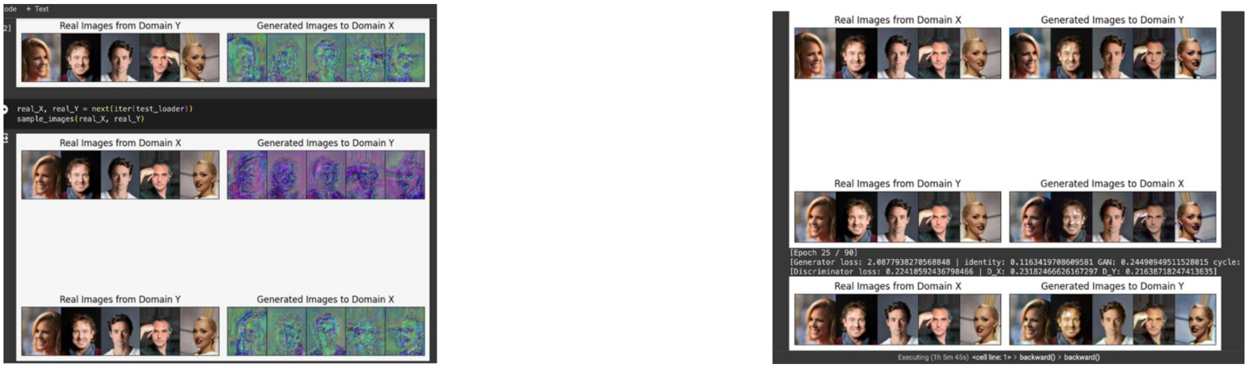

Fig 5: DIFFERENT STAGES OF CYCLE GAN

Fig 1 represents generation of images from Celeb Dataset. Moving to Fig 2, images are generated through the utilization of Style GAN. Fig 3 and 4 highlight the application of SR GAN, demonstrating the enhancement in image quality achieved by this architecture. Finally, in Fig 5, the implementation of Cycle GAN is presented at various epochs, revealing the evolution of image translation over the training process.

Discussions

- In image generation from the Celeb Dataset using a Basic GAN, the diversity and realism in the generated faces are evident, highlighting the foundational principles of adversarial training.
- The use of Style GAN for image generation yields images with a heightened focus on realistic and high-quality features. The incorporation of a Style GAN framework emphasizes the importance of a large, high-quality dataset, reflected in the refined visual outputs.
- The utilization of perceptual loss in SR GAN contributes to the improved super-resolution, resulting in crisper and more detailed images. This underscores the effectiveness of SR GAN in upscaling low-resolution images while preserving finer textures.
- The cycle consistency loss in Cycle GAN ensures that the translated images maintain consistency with the original input, providing a glimpse into how the model refines its ability to perform unpaired image-to-image translation over successive training iterations.

Key Takeaways

- Our novel contribution lies in our holistic perspective, demonstrating how these architectures can synergize to address various challenges in creative image synthesis, image-to-image translation without paired data, realistic image generation, and super-resolution.
- This integrative framework encourages a more comprehensive understanding of generative modeling, offering a unique synthesis of foundational concepts and advanced applications, thereby contributing to the broader landscape of artificial intelligence and computer vision research.
- We recognize the societal impact of generative modeling in image synthesis and manipulation and advocate for the responsible development and deployment of such technologies. By addressing issues related to bias, fairness, and interpretability within the context of creative image synthesis,

image-to-image translation, and super-resolution, our framework contributes to a more ethically grounded artificial intelligence landscape.

Limitations

- Training instability: GANs are notorious for their training instability. The process involves two networks (generator and discriminator) trying to outperform each other, and this competition can lead to oscillations or mode collapse. Achieving balance during training can be tricky.
- Sensitive to Hyperparameters: GANs are sensitive to hyper parameter choices, making finding the right configuration crucial for effective training.
- Computational resources: Training GANs can be computationally expensive, especially for large datasets and complex models. GPUs or TPUs are often necessary to accelerate training, and even with these resources, training can take a significant amount of time.

Conclusion

In conclusion, implementation of GANS serves as a foundational exploration, providing insights and methodologies essential for diverse generative modeling tasks, from fundamental adversarial training to specialized applications like image-to-image translation, realistic image synthesis, and super-resolution.

References

1. "A Style-Based Generator Architecture for GANs - Generating and Tuning Realistic Artificial Faces" https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431
2. "An End-to-End Introduction to Generative Adversarial Networks (GANs)" https://www.analyticsvidhya.com/blog/2021/10/an-end-to-end-introduction-to-generative-adversarial-networksgans/
3. "How Cycle GAN Works?" https://developers.arcgis.com/python/guide/how-cyclegan-works/
4. "Super Resolution GAN (SRGAN)" https://www.geeksforgeeks.org/super-resolution-gan-srgan/

Data sources:

1. https://www.kaggle.com/datasets/jessicali9530/celeba-dataset/
2. https://drive.google.com/drive/folders/18DpJPC9osdmGuSPbbLNZnR_Sk38d7gtF?usp=share_link
3. https://drive.google.com/drive/folders/1DwH8QqyvHAPik2QwMAW06P9fRQNldYk1?usp=share_link

Team member contributions

Task	Team member
Background and topic research	Aamera Shaikh, Ganesh Gude, Rahul Solanki
Dataset collection	Aamera Shaikh
Model Research	Ganesh Gude
Basic GAN & Style GAN Implementation	Ganesh Gude
Cycle GAN Implementation	Aamera Shaikh
SR GAN Implementation	Rahul Solanki
Result evaluation	Aamera Shaikh, Ganesh Gude, Rahul Solanki
Presentation & Report	Aamera Shaikh, Ganesh Gude, Rahul Solanki

Link to GitHub repo with project code: https://github.com/ganeshg0722/DeepLearningProject