ChitraE

Contributors:Aneesh, Saloni, Bhuvan, Sachin, Chetan, Nihal, Rohith, Nandana, Himanshu, Buddha Teja, Nived

National Institute of Technology, Karnataka

Objectives

The project aimed to create a deep learning model that can apply a Monet-style filter to input images of various scenes. The model uses a style-transfer algorithm to produce images that resemble Claude Monet's signature style. Another objective was to develop a web application that integrates the model with a Flask web framework, allowing users to upload their images and view the Monet-style transformation in real-time.

Motivation

The primary motivation for undertaking this project was to explore the field of computer vision and generative models. Specifically, we were interested in the task of style transfer, which involves transforming an input image into the style of another image. We found the task to be challenging yet intriguing and wanted to experiment with different techniques and models to achieve it. Another motivation for this project was to create a tool that can generate monet-style paintings from real-life images. Such a tool can be useful for artists and designers who want to quickly visualize their ideas in a monet style without having to manually create the painting from scratch. Additionally, it can be an interesting application for art enthusiasts who want to explore Monet's unique style and see how different images can be transformed into it using computer algorithms. Overall, the combination of technical challenge and artistic creativity made this project an exciting endeavor for us. We hope that the tool we created can inspire others to explore the possibilities of generative models and computer vision in the context of art and design.

Technologies and Frameworks used

1. Python - Programming language used for development
2. TensorFlow - Open-source machine learning framework used for building and training deep learning models
3. Keras - High-level API built on top of TensorFlow used for building and training deep learning models
4. NumPy - Python library used for numerical computing
5. PIL - Python Imaging Library used for handling image data
6. Flask - Web framework used for building the web application
7. Flask-WTF: It is an integration of the WTForms library with Flask. It is used for form handling in the web application.
8. WTForms: It is a flexible forms validation and rendering library for Python. It is used in conjunction with Flask-WTF for form handling in the web application.

Dataset Used :

The 'I am something of a painter myself' dataset is a collection of images that can be used for training and testing generative adversarial networks (GANs) for image-to-image translation tasks. The dataset consists of two domains: photos and Monet paintings. The photos domain contains 7,038 high-quality landscape photos from various sources, such as Flickr and Unsplash. The Monet paintings domain contains 300 artworks by the famous impressionist painter Claude Monet, curated from WikiArt. The images in both domains have a resolution of 256x256 pixels and are stored in TFRecord format.

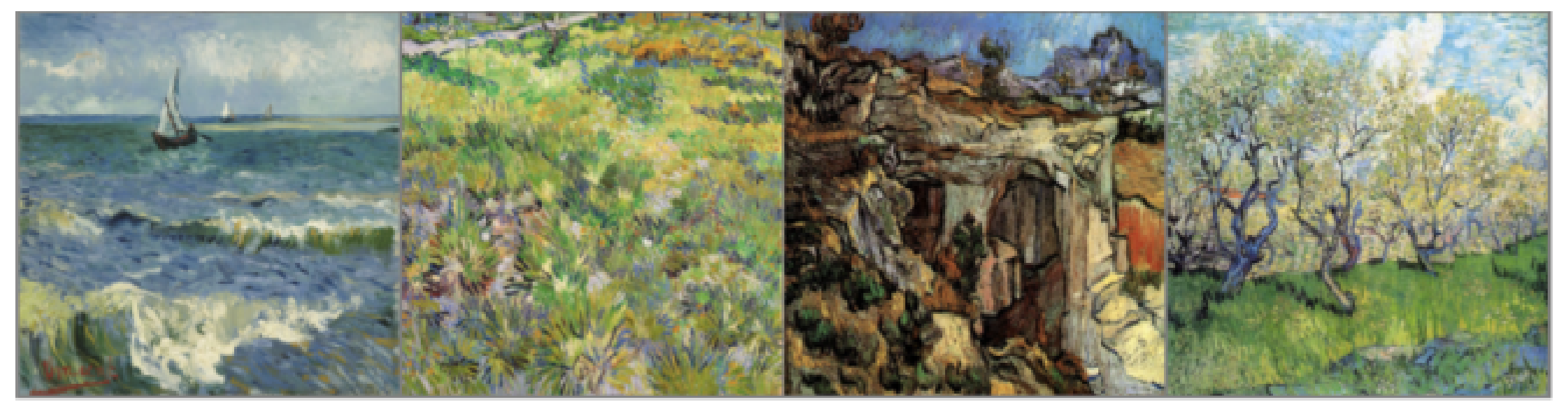

Figure 1:Style input [Vincent Van Gogh's Paintings]

Figure 2:Content input [Landscape Images]

Cyclic GAN

The Cyclic GAN consists of two pairs of generators and discriminators: a Monet generator (which generates Monet-style images from real photos) and discriminator, and a photo generator (which generates real photos from Monet-style images) and discriminator. The two pairs are trained simultaneously, with the goal of the Monet generator producing Monet-style images that are indistinguishable from actual Monet paintings, and the photo generator producing real photos that are indistinguishable from those taken by a camera. A generator function, Generator() defines the architecture of the generators, which consist of downsampling layers to extract low-level features from the input image, followed by upsampling layers to generate the output image.

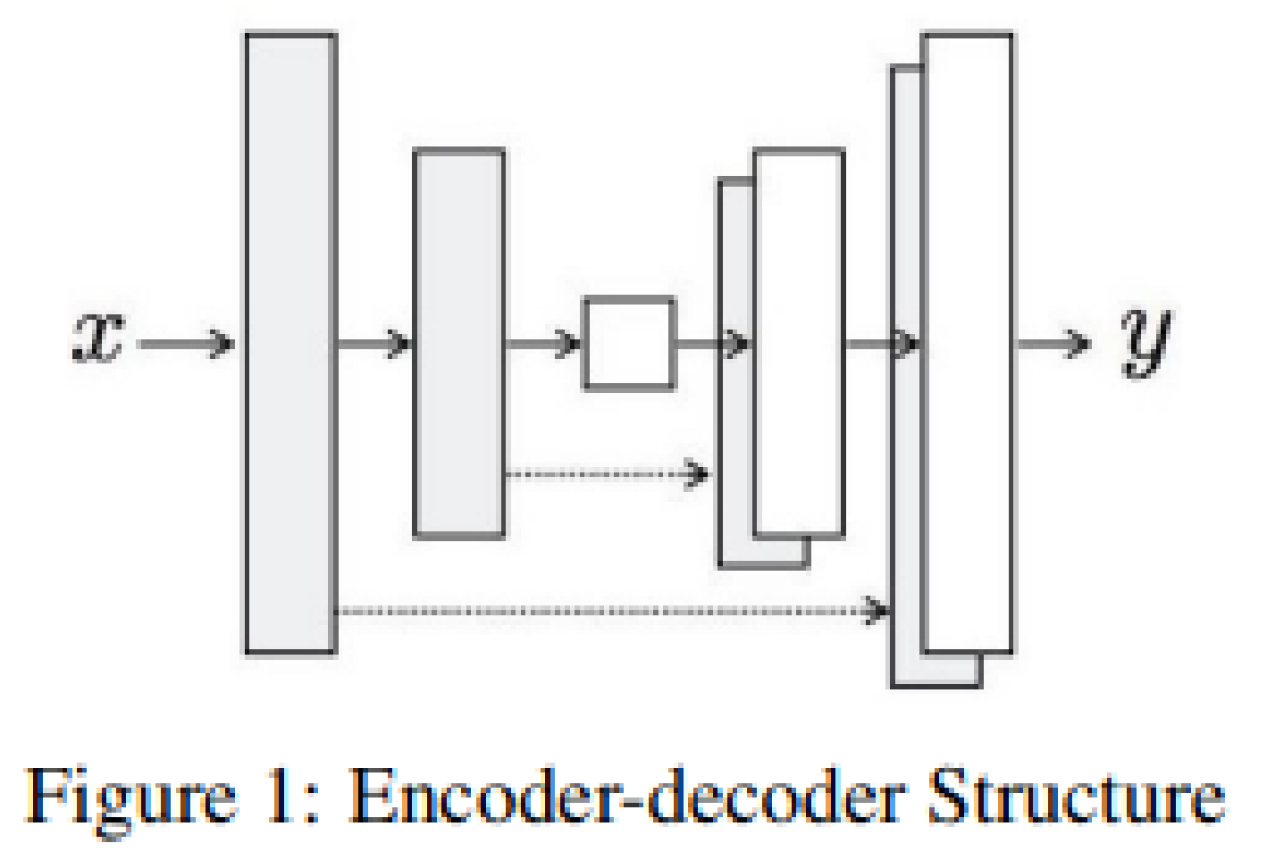

Figure 3:cGAN Architecture

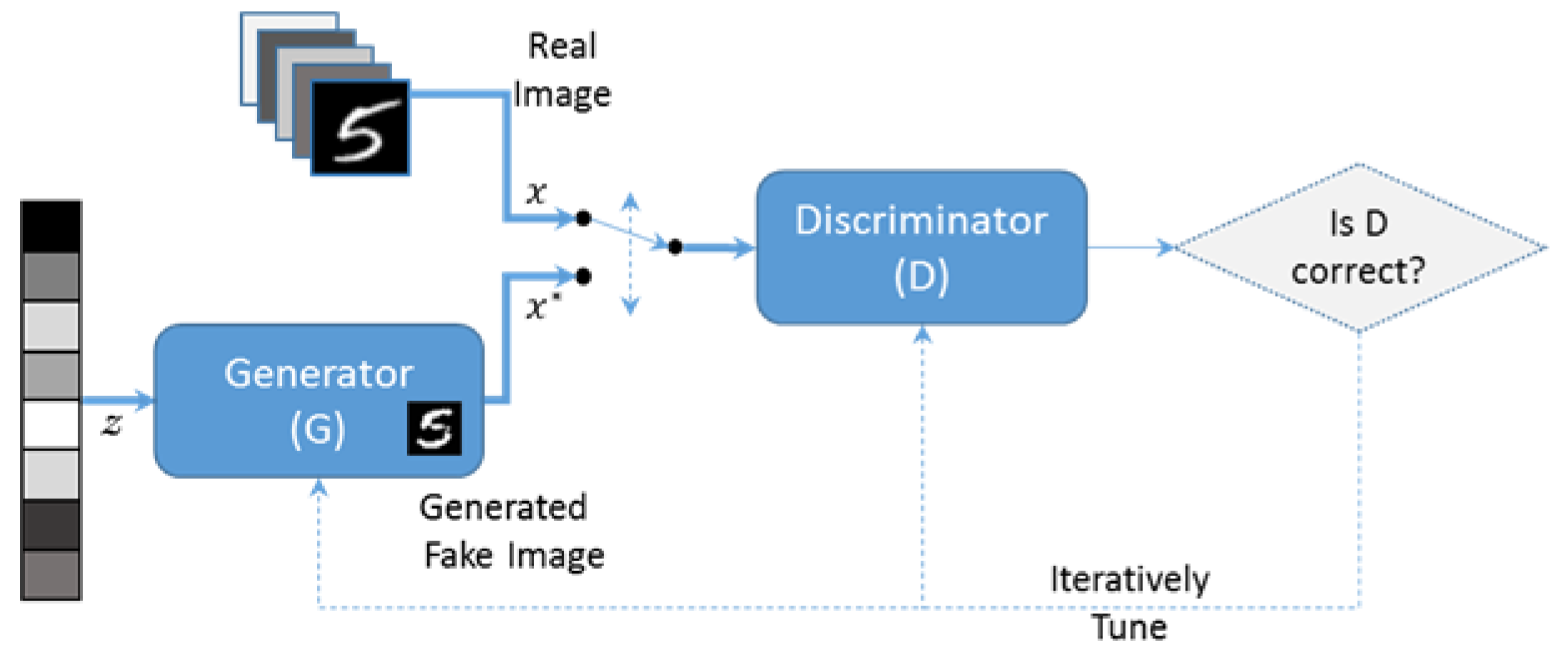

Figure 4:Typical cGAN

Results and observations

Our proposed model, which uses a CycleGAN architecture to transform input images into Monet-style paintings, achieved impressive results. We evaluated the model on a test set of diverse images and obtained visually appealing outputs that successfully mimicked Monet's style of painting.

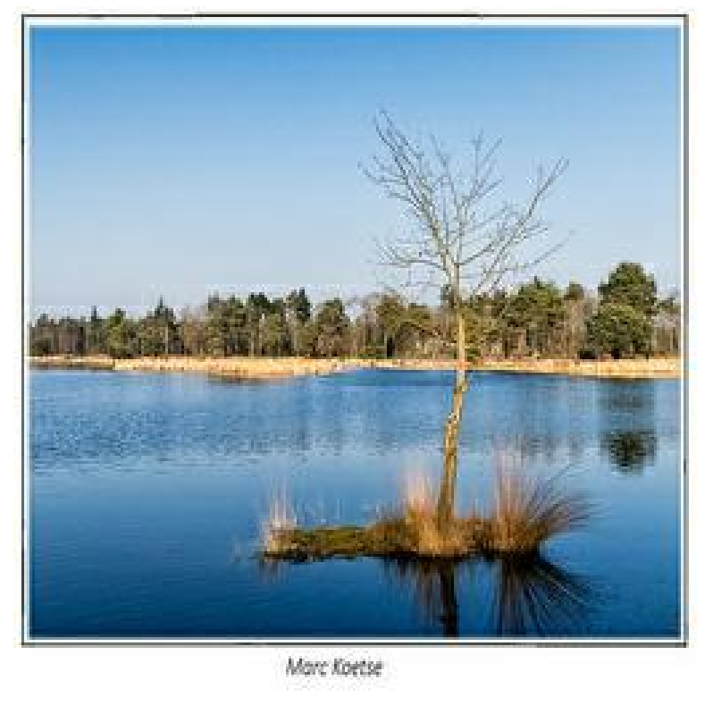

Figure 5:Input Image

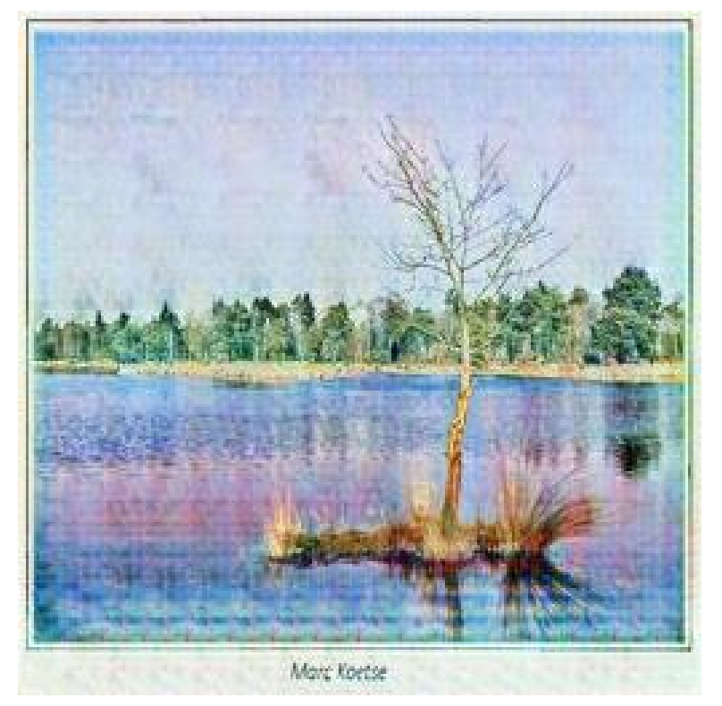

Figure 6:Output Image

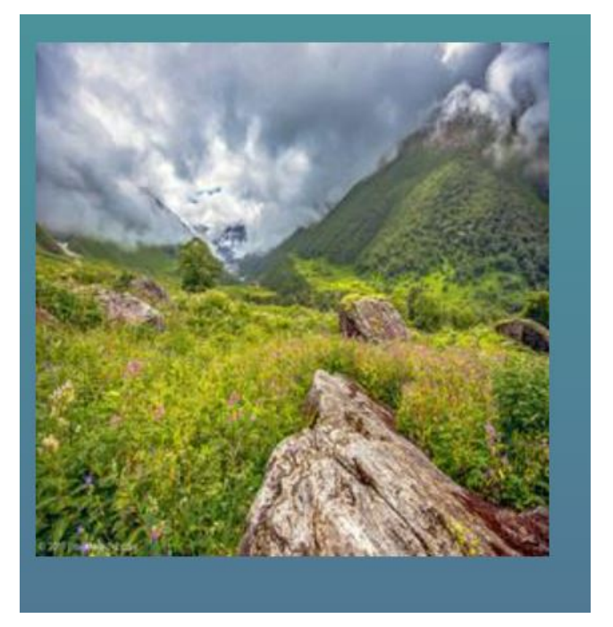

Figure 7:Input Image

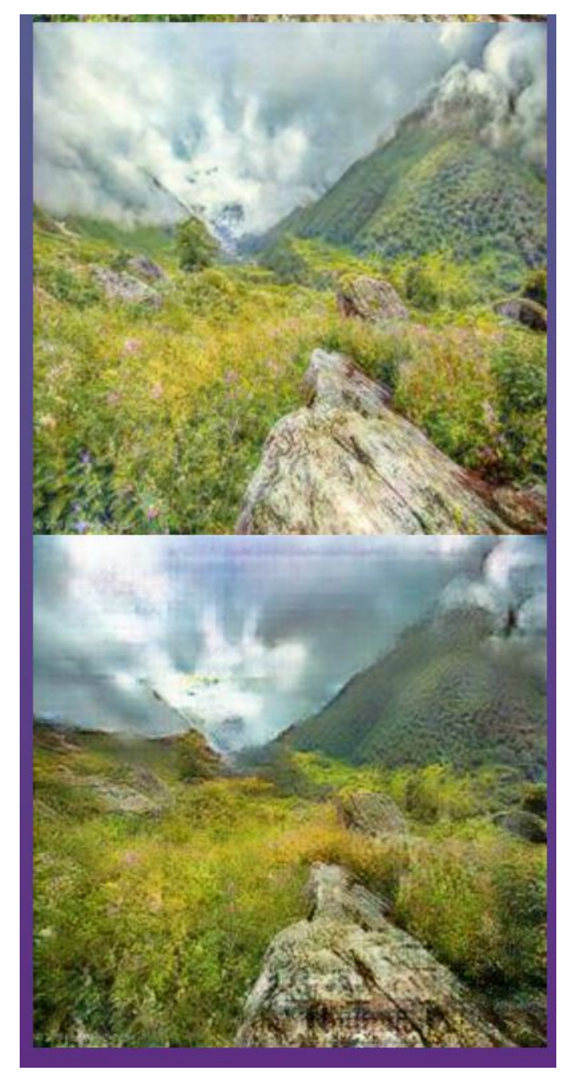

Figure 8:Output Image