Project 3 Report Paper:

Impact of activation function when training a GAN for Monet-style image translation

Joseph Martínez

Old Dominion University, jmart130@odu.edu

*Abstract* - The *I’m Something of a Painter Myself* Kaggle competition serves as a platform to explore the capabilities of Generative Adversarial Networks (GANs) in replicating the artistic style of Claude Monet. With the objective of generating thousands of Monet-style images, participants are tasked with training a GAN, hyperparameter optimization, and data preprocessing to capture the essence of Monet's work. Beyond its artistic implications, the competition holds broader significance for computer vision and machine learning, with successful GAN models potentially finding applications in digital art synthesis, style transfer, and content generation.

In this study, we adapted the CycleGAN\_Monet framework by incorporating a change to the activation functions, transitioning from LeakyReLU and ReLU to Exponential Linear Unit (ELU). Our modification aimed to explore alternative nonlinearities and their impact on discriminator performance within the adversarial training paradigm. While ELU offers potential benefits such as improved gradient propagation and faster convergence during training, our experimentation revealed nuanced insights into its effects on GAN training dynamics.

Results demonstrated notable differences between the original and modified models, particularly in terms of training stability and convergence. Fluctuations and higher losses observed in the modified model underscored the intricate interplay between activation functions, optimization landscapes, and training dynamics in GANs. Despite the potential advantages of ELU, its introduction posed challenges such as increased model instability and oscillations, highlighting the importance of careful experimentation and tuning when modifying key architectural components.

*Index Terms* – GAN, Monet-style, MiFID, Image generation

Problem Definition

The "I'm Something of a Painter Myself" Kaggle competition [1] aims to explore the capabilities of Generative Adversarial Networks (GANs) in replicating the unique artistic style of Claude Monet. This effort stems from a desire to advance the field of computer vision while also delving into the nuances of artistic expression. With the specific goal of generating 7,000 to 10,000 Monet-style images, participants are challenged to develop GAN models capable of capturing the essence of Monet's work.

This competition presents a multifaceted challenge, requiring participants to navigate the complexities of GAN training, hyperparameter optimization, and data preprocessing. Additionally, participants must leverage their creative intuition to encapsulate the distinctive characteristics of Monet's artistic style within their generated images.

Beyond its artistic implications, the outcomes of this competition hold broader significance for the field of computer vision and machine learning. Successful GAN models capable of faithfully replicating Monet's style could have applications in various domains, including digital art synthesis, style transfer, and content generation. Thus, the competition serves as a platform for fostering innovation in both art generation and computational methodologies, with potential implications for real-world applications of deep learning techniques in creative contexts.

Methods

We adapted the code *CycleGan\_Monet[[1]](#footnote-2)* from Daisy Huang by incorporating changes to the activation function while preserving the overall structure, preprocessing steps, training regimen, and evaluation framework. Our modification diverges from the prior LeakyReLU activation to employ the Exponential Linear Unit (ELU) activation function. This alteration aims to explore alternative nonlinearities and their impact on the discriminator's performance within the adversarial training framework. Our adjustment seeks to enhance model adaptability and explore the implications of activation function selection on adversarial learning dynamics.

Our submission made this change to the original model since

ELU has been gaining popularity in the realm of deep learning [2] and switching from ReLU or Leaky ReLU to ELU can lead to improved performance and faster convergence during training.

This section outlines the methodology, including data preprocessing, model architecture, training process, and evaluation.

## Data Preprocessing:

The data processing for this project involves several steps to prepare the Monet paintings and photographs for training the GAN:

1. **Checking Image Dimensions:** The dimensions of all Monet paintings and photographs are checked to ensure consistency. For both sets of images, all images have dimensions of 256x256 pixels with 3 color channels (RGB).
2. **Defining Image Parameters:** Image parameters such as size, channels, and shape are defined for further processing. Each image has a size of (256, 256) pixels and consists of 3 color channels (RGB). The shape of each image is (256, 256, 3).
3. **Loading Images and Preprocessing**: A function is defined to load images from a specified path, convert them to arrays, and scale the pixel values to the range [-1, 1]. Monet paintings and photographs are loaded using this function, resulting in NumPy arrays. The Monet paintings array has a shape of (300, 256, 256, 3), indicating 300 images with a resolution of 256x256 pixels and 3 color channels. The photographs array has a shape of (7038, 256, 256, 3), indicating 7038 images with the same resolution and color channels.

## Model Architecture:

The model architecture of Daisy Huang comprises two Generative Adversarial Networks (GANs), each consisting of a generator and a discriminator (see figure below).

A diagram of a garden

Description automatically generated

Figure I

Model’s framework

The generator network *Photo Generator* takes a painting as input and generates a corresponding photo. Similarly, *Monet Generator* takes a photo as input and produces a generated painting. Both generators utilize a CycleGAN architecture, which involves a series of downsampling convolutional blocks, upsampling convolutional blocks, and ResNet blocks. The ResNet blocks, consisting of two 3x3 convolutional layers, facilitate feature preservation and enable the generation of high-quality images.

The discriminator network is designed based on the PatchGAN approach, which evaluates the realism of generated images at the patch level. The discriminator takes either a real or a generated image as input and outputs a binary classification indicating whether the image is real or fake. The discriminator architecture consists of convolutional layers with increasing filter sizes, followed by leaky ReLU activation functions.

1. **Architecture of the generator**

The generator CNN takes a 256x256x3 image as input and produces a corresponding output image of the same size. It has 35,276,553 params, from which all are trainable. The architecture consists of several convolutional layers followed by instance normalization and activation functions. The network starts with a convolutional layer with 64 filters, followed by instance normalization and a ReLU activation function. Subsequently, there are additional convolutional layers with increasing numbers of filters (128, 256) along with instance normalization and activation functions.

The unique aspect of this architecture is the inclusion of ResNet blocks, which facilitate the preservation of image features across layers. Each ResNet block consists of two convolutional layers, and the output of each block is concatenated with the input to the block, channel-wise. This enables the network to capture and retain essential image details during the translation process.

After the ResNet blocks, the network continues with additional convolutional layers, each followed by instance normalization and activation functions. Finally, the output is obtained through a series of transposed convolutional layers, which upsample the feature maps to the desired output size of 256x256x3. Instance normalization and activation functions are applied to the output of each transposed convolutional layer to ensure the stability and quality of the generated image. Overall, this generator architecture is capable of translating paintings into photos and vice versa, capturing the distinctive characteristics of each style effectively.

1. **Architecture of the discriminator**

The discriminator architecture is a CNN that takes a 256x256x3 image as input and produces a single scalar value indicating the likelihood that the input image is real. The architecture comprises several convolutional layers followed by leaky ReLU activation functions and instance normalization.

The network begins with a convolutional layer with 64 filters and a kernel size of 4x4, followed by a leaky ReLU activation function with a negative slope of 0.2. Subsequently, additional convolutional layers with increasing numbers of filters (128, 256, 512) are employed, each followed by instance normalization and leaky ReLU activation functions.

Importantly, the architecture features two sets of convolutional layers with 512 filters each, both followed by instance normalization and leaky ReLU activation functions. These layers play a crucial role in capturing high-level features of the input image, aiding the discriminator in distinguishing between real and fake images effectively.

Finally, the output layer consists of a convolutional layer with a single filter, which produces a scalar value indicating the likelihood that the input image is real. No activation function is applied to the output layer, allowing it to output unbounded values.

## Training Process:

The training process involves composite models that combine the generators and discriminators. Two composite models, *Monet\_to\_Photo* and *Photo\_to\_Monet*, are defined to facilitate the training of GANs in both directions. These composite models incorporate adversarial loss, identity loss, forward cycle loss, and backward cycle loss, enabling the generators to learn the mapping between paintings and photos bidirectionally. Training was performed over 2 epochs with a batch size of 1 sample.

During training, real, and fake samples are generated for both paintings and photos. The generators are trained to minimize the discrepancy between generated and real images, while the discriminators are trained to distinguish between real and fake images accurately. Additionally, an image pool is employed to stabilize training by providing a diverse set of fake images for discriminator updates.

The architecture of each CNN for generators and discriminators are presented below.

## Evaluation:

The composite loss function guides the training of generator models in GANs. These components ensure fidelity, identity preservation, and consistency in image translation. It comprises:

1. adversarial loss (L2 or Mean Squared Error),
2. identity loss (L1 or Mean Absolute Error),
3. forward, and backward cycle losses (L1 or Mean Absolute Error).

Changes to Original Architecture

## Activation function:

In contrast to the original implementation utilizing the activation functions LeakyReLU and Rectified Linear Unit (ReLU), we decided to utilize the Exponential Linear Unit (ELU) activation function (See figure below for visualization of activation functions). ELU has been gaining popularity in the realm of deep learning due to its unique properties [2]. Unlike ReLU and Leaky ReLU, ELU introduces negative values, which allows the model to learn better representations. This is because ELU can push mean unit activations closer to zero, similar to batch normalization, but with lower computational complexity [3]. Furthermore, ELU handles the vanishing gradient problem better than ReLU by smoothing the function, preventing sudden jumps, and allowing for faster convergence during training. Therefore, switching from ReLU or Leaky ReLU to ELU can lead to improved performance and faster convergence during training.

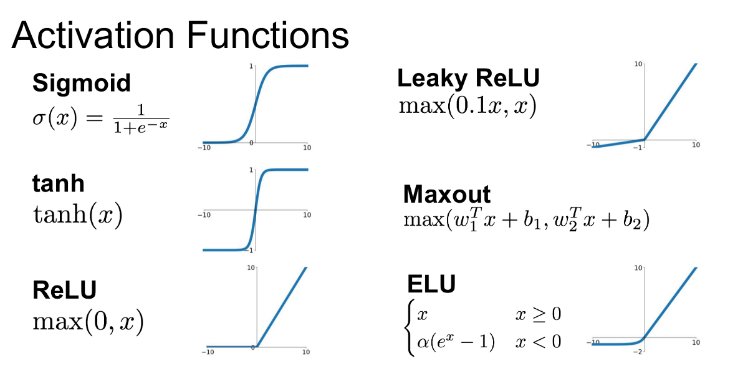


Figure Ii

Well-known activation functions[[2]](#footnote-3)

Results

The performance of the model trained using LeakyReLU and ReLU is shown below. Followed by the model using ELU.

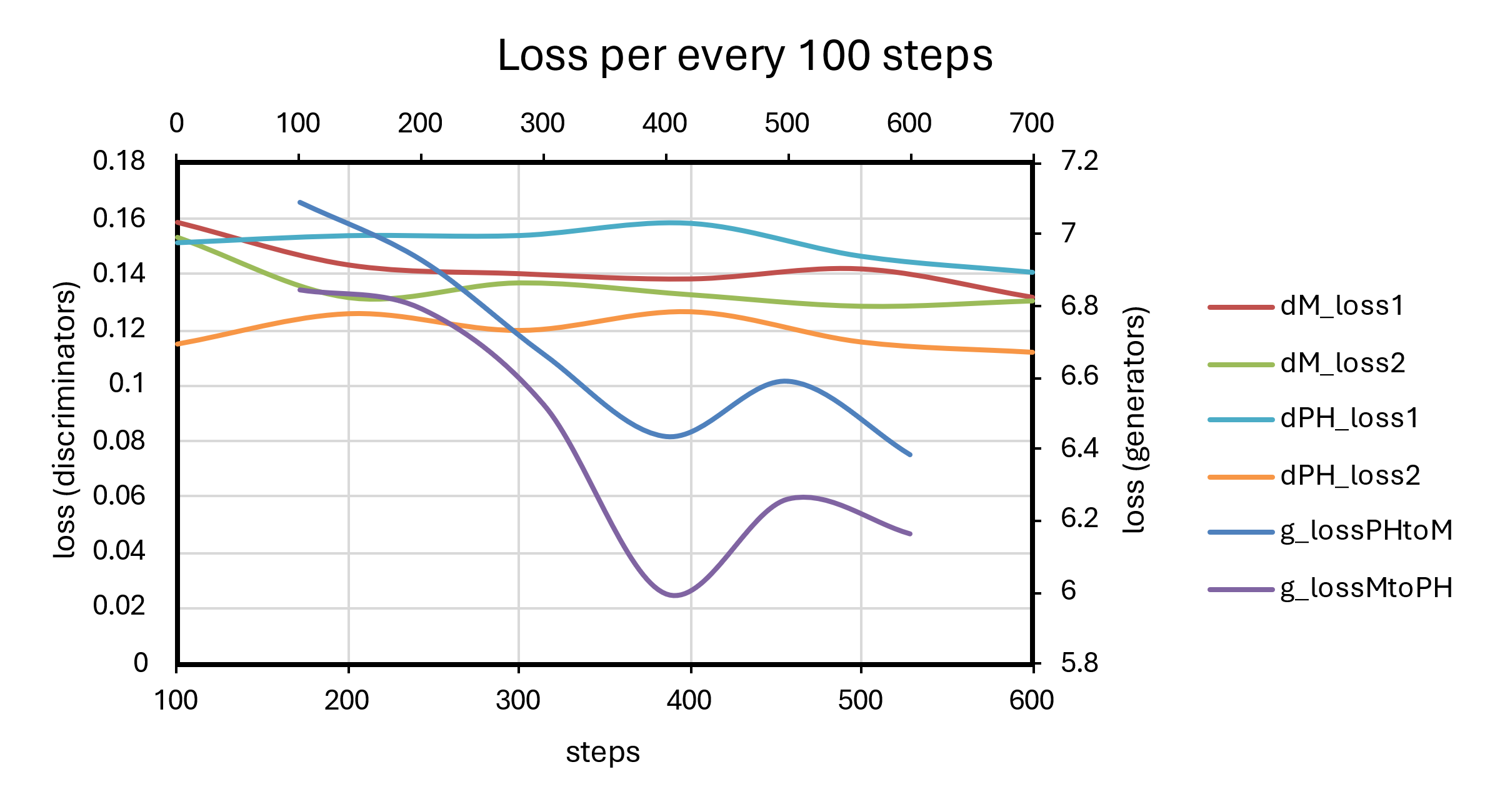


Figure III

Loss vs step plot of the original model

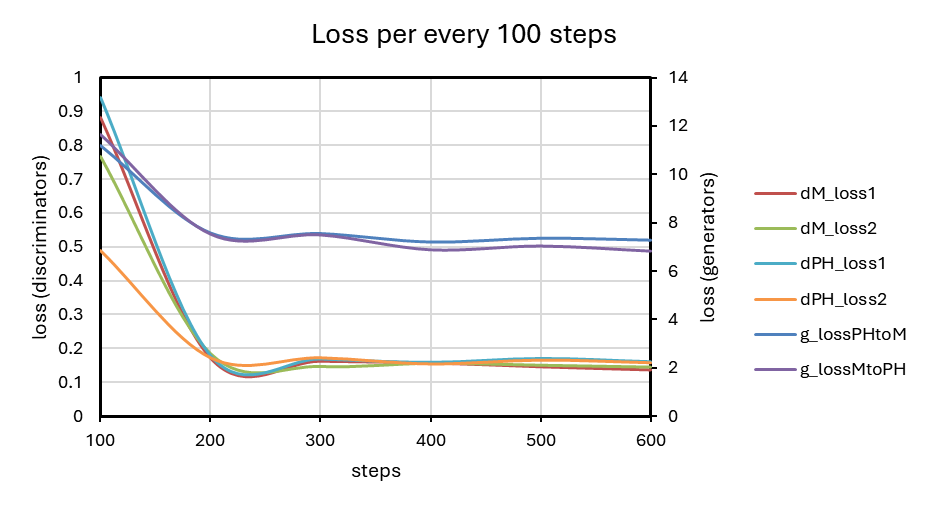


Figure IV

Loss vs step plot of the modified model

The results from the original and modified models provide insights into the training dynamics and performance changes. In the original model, the generator and discriminator losses for both the *Photo to Monet* and *Monet to Photo* translations appear to stabilize over time. Specifically, the generator losses for both translations (*g\_lossPHtoM* and *g\_lossMtoPH*) exhibit relatively consistent values, indicating that the generator successfully minimizes its loss during training. Similarly, the discriminator losses (*dM\_loss1, dM\_loss2, dPH\_loss1*, and *dPH\_loss2*) demonstrate stability, suggesting effective adversarial training.

In contrast, the modified model yields fluctuations in the generator and discriminator losses across steps. Notably, the generator losses for both translations fluctuate more prominently compared to the original model, indicating potential challenges in convergence. Additionally, the discriminator losses exhibit variations, which may suggest difficulties in effectively discriminating between real and generated samples.

Furthermore, when comparing the magnitude of losses between the original and modified models it appears that the modified model tends to have higher losses, particularly in the initial steps. This increase in losses could imply either a more complex optimization landscape or a divergence in training dynamics due to the introduction of the ELU activation function.

Activation functions play a crucial role in shaping the nonlinearity of neural networks and can influence the optimization landscape during training. ELU activation function, while offering benefits such as improved gradient propagation and reduced vanishing gradient problem, may introduce new challenges during optimization due to its different mathematical properties compared to LeakyReLU. Specifically, ELU introduces exponential behavior for negative inputs, potentially leading to more pronounced activations and gradients, which could contribute to increased model instability and oscillations during training. Moreover, the altered activation function might disrupt the balance between the generator and discriminator networks, affecting their ability to learn meaningful representations and adversarial dynamics. Additionally, the change in activation function could require adjustments in hyperparameters or training strategies to ensure effective convergence, highlighting the importance of careful experimentation and tuning when modifying key architectural components of neural networks.

Overall, these results suggest that the modification to the activation function may have influenced the training dynamics of the model, potentially affecting its convergence and performance.

## Lessons Learned

The experiment highlights several crucial lessons regarding the impact of activation function selection on the training dynamics and performance of GANs.

1. The switch from LeakyReLU to Exponential Linear Unit (ELU) activation function introduced notable changes in the model's behavior, as evidenced by fluctuations and higher losses observed during training. This underscores the importance of understanding the mathematical properties and implications of different activation functions when designing neural network architectures.
2. While ELU offers advantages such as improved gradient propagation and mitigation of the vanishing gradient problem, its introduction can also introduce new challenges, including increased model instability and oscillations. This emphasizes the need for careful consideration and experimentation when introducing architectural modifications to neural networks.
3. The observed discrepancies between the original and modified models highlight the complexity of GAN training dynamics and the intricate interplay between components such as generators and discriminators.
4. The findings underscore the necessity of fine-tuning hyperparameters and training strategies to ensure effective convergence and optimal performance when implementing architectural changes in deep learning models.

Conclusion

The *I’m Something of a Painter Myself* Kaggle competition presents a compelling challenge to explore the potential of GANs in replicating the distinctive artistic style of Claude Monet. With the goal of generating thousands of Monet-style images, participants navigated the complexities of GAN training, hyperparameter optimization, and data preprocessing while striving to encapsulate Monet's artistic essence. Beyond its artistic significance, the competition holds broader implications for computer vision and machine learning, with successful GAN models potentially finding applications in digital art synthesis, style transfer, and content generation.

In this study, we adapted the *CycleGAN\_Monet* (Daisy Huang) framework by introducing a change to the activation functions, shifting from LeakyReLU and ReLU to ELU. This alteration aimed to explore alternative nonlinearities and their impact on discriminator performance within the adversarial training paradigm. While ELU has garnered attention in deep learning for its unique properties and potential performance improvements, our experimentation revealed nuanced insights into its effects on GAN training dynamics.

Our results demonstrated notable differences between the original and modified models, particularly in terms of training stability and convergence. Fluctuations and higher losses observed in the modified model underscored the intricate interplay between activation functions, optimization landscapes, and training dynamics in GANs. Despite the potential benefits of ELU, its introduction posed challenges such as increased model instability and oscillations, highlighting the importance of careful experimentation and tuning when modifying key architectural components.

Overall, this study provides valuable lessons for researchers and practitioners in the field of deep learning. By integrating these insights into future research and development efforts, we can advance the capabilities of GANs and unlock their potential for creative expression and computational innovation.

Author Information

**Joseph Martínez,** Modeling and Simulation Engineering Master’s student at Old Dominion University (ODU).

References

1. Amy Jang, P.C.A.S.U., *I’m Something of a Painter Myself*. <https://www.kaggle.com/competitions/gan-getting-started/overview>: Kaggle.

2. Dubey, S.R., S.K. Singh, and B.B. Chaudhuri, *Activation functions in deep learning: A comprehensive survey and benchmark.* Neurocomputing, 2022. **503**: p. 92-108.

3. Clevert, D.-A., T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (elus). arXiv 2015.* arXiv preprint arXiv:1511.07289, 2020.

1. <https://www.kaggle.com/code/daisy1020/cyclegan-monet> [↑](#footnote-ref-2)
2. Image taken from: <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092> [↑](#footnote-ref-3)