Adapting GANs: Can the Generator act as a Classifier?

Vaishali Singh
Student Number 230710935
MSc. Artificial Intelligence, QMUL
v.singh@se23.qmul.ac.uk
Project Supervisor: Dimitrios Kollias
d.kollias@qmul.ac.uk

***Abstract*—This research examines the application of Generative Adversarial Networks (GANs), with a particular focus on the StarGAN architecture, to perform image classification across different datasets. Traditionally, GANs are utilized for generating synthetic data; however, this study investigates the potential of repurposing the StarGAN generator for classification tasks. The adaptability of StarGAN, which facilitates attribute manipulation across multiple domains, is employed to classify images in the CelebA dataset, containing annotated images of celebrities, and the MNIST dataset which consists of images of handwritten digits. Through this approach, the study aims to evaluate the effectiveness of using a generative model in a discriminative capacity. The results suggest that the generator can be successfully adapted for image classification, providing insights into the broader applicability of generative models for discriminative tasks. This research contributes to the ongoing exploration of novel uses for generative models, potentially influencing future directions in deep learning.**

***Index Terms*—GANs, StarGAN, Deep Learning, MNIST, CelebA**

I. Introduction

A. Research Background

Generative Adversarial Networks (GANs), introduced by Ian Goodfellow in 2014 [17], have significantly impacted the field of artificial intelligence, particularly in image synthesis, data augmentation, and unsupervised learning [1]. GANs are made up of two neural networks, a generator and a discriminator, which are trained together using an adversarial learning process. The generator produces images that try to replicate real data, while the discriminator assesses these images to differentiate between authentic and generated ones. This process results in the generator gradually improving its image synthesis capabilities over time.

StarGAN, a notable variant of GANs, extends the basic framework to multi-domain image translation. Proposed by Y. Choi [6], StarGAN can convert images across multiple domains using a single generator-discriminator setup, making it particularly efficient for tasks that involve manipulating images in various styles or categories. This capability is particularly advantageous in scenarios requiring domain adaptation and image-to-image translation, such as facial attribute editing, style transfer, and multi-domain image synthesis.

In recent years, there has been growing interest in leveraging GANs for tasks beyond image synthesis, including classification. Traditional classification methods often rely on supervised learning and require large, labeled datasets, which can be resource-intensive to obtain. GANs, particularly models like StarGAN, offer a promising alternative by using the learned features from the generative process for classification tasks, potentially reducing the need for extensive labeled data. [14]

B. Domestic and International Research Status

Research on GANs and their applications in classification tasks has seen significant developments worldwide. In the domestic context, numerous studies have investigated the application of GANs for data augmentation and the enhancement of classification models. For instance, researchers in China have applied GANs to medical imaging, enhancing the accuracy of disease detection by generating additional training samples from limited datasets. [5]

Internationally, the focus has been on improving GAN architectures and exploring their potential in various domains. In the United States, extensive research has been conducted on using GANs for cybersecurity, where they generate adversarial examples to train more robust classifiers [7]. European researchers have investigated the application of GANs in autonomous driving, where generative models create diverse driving scenarios to improve the training of vehicle perception systems. [8]

In the context of using GANs for classification, several notable studies have demonstrated the potential of adapting GAN generators for this purpose. For example, recent work has shown that generators pre-trained on large datasets can be fine-tuned for classification tasks, leveraging the rich feature representations learned during the generative process [9].

II. Background and Related Work

A. Generative Adversarial Networks (GANs)

These frameworks involve a pair of neural networks that engage in a competitive process. The generator network is

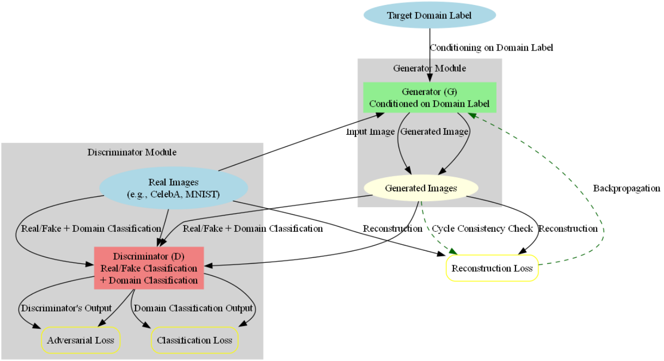

Fig. 1. Overview of the StarGAN Architecture. This diagram illustrates the interaction between the Generator (G) and Discriminator (D) modules within the StarGAN model. The generator takes an input image and a target domain label, producing a generated image conditioned on the domain label. The discriminator evaluates both real and generated images for real/fake classification and domain classification. The architecture incorporates adversarial loss, classification loss, and cycle-consistency loss to ensure that the generated images are realistic, correctly classified, and maintain consistency during reconstruction

responsible for producing new data samples, while the discriminator network critically examines these samples, comparing them against real data to determine their authenticity. This adversarial interaction drives both networks to improve over time, with the generator striving to produce increasingly realistic outputs and the discriminator refining its ability to distinguish between genuine and generated data.The main goal of GANs is to create data that cannot be distinguished from real data.

B. StarGAN

StarGAN expands the GAN framework to manage multi-domain image-to-image translation with just one generator-discriminator pair. It is particularly effective in tasks requiring the transformation of images across different styles or attributes. StarGAN's ability to perform multiple image translation tasks with a unified model makes it a versatile tool in the field of image synthesis.

C. Previous Work on GANs as Classifiers

Several studies have explored the potential of GANs beyond generation, including their use in classification. Pathak and Dufour (2023) [2] demonstrated the feasibility of GAN-classifiers, revealing knowledge gaps in GAN training. Miranda (2022) [4] proposed a method to directly use trained classifiers for image generation, indicating a possible bidirectional relationship. This body of work highlights the potential for repurposing GAN components for tasks beyond their original scope.

D. StarGAN Overview

StarGAN is designed to enable image-to-image translation across multiple domains using a unified model. The architecture is composed of a single generator G and a discriminator D, which work together to translate images from one domain to another while ensuring the translated images are realistic and consistent with the target domain.

Generator G: The generator takes an input image x from a source domain and a target domain label c. It then produces a translated image $G(x, c)$ that reflects the characteristics of the target domain while retaining the essential content of the original image.

Discriminator D: The discriminator's role is twofold. It evaluates the authenticity of the input image, determining whether it is real (from the training dataset) or fake (generated by G). Additionally, it predicts the domain label of the image, ensuring that the generated image matches the intended target domain.

E. Loss Functions

StarGAN employs several loss functions to ensure the quality of generated images and the effectiveness of domain translation:

Adversarial Loss: This loss encourages the generator to produce realistic images that can fool the discriminator.

$$L^D_{\text{adv}} = E_x[\log D(x)] + E_{x,c}[\log(1 - D(G(x, c)))] \tag{1}$$

$$L^G_{\text{adv}} = E_{x,c}[\log D(G(x, c))] \tag{2}$$

Domain Classification Loss: Ensures that the generated images belong to the target domain. The generator modifies the attributes based on the target domain, while the discriminator classifies the domain of both real and generated images.

$$L^D_{\text{cls}} = E_{x,c}[-\log D_{\text{cls}}(c \mid x)] \tag{3}$$

$$L^G_{\text{cls}} = E_{x,c}[-\log D_{\text{cls}}(c \mid G(x, c))] \tag{4}$$

Reconstruction Loss: Encourages the generator to preserve the core content of the original image while modifying its attributes.

$$L_{\text{rec}} = E_{x,c,c'}[\|x - G(G(x, c), c')\|_1] \tag{5}$$

F. Loss Functions and Training Strategy

In this project, while we do not directly modify the original StarGAN generator to act as a classifier, we leverage the existing architecture and loss functions for an indirect evaluation. Specifically, we analyze whether the generator's ability to manipulate image attributes can lead to high classification performance when evaluated by the discriminator. We rely on the adversarial, classification, and reconstruction losses to guide the generator towards producing realistic and accurately classified images.

The generator's overall loss function is:

$$L_G = L^G_{\text{adv}} + \lambda_{\text{cls}} L^G_{\text{cls}} + \lambda_{\text{rec}} L_{\text{rec}} \tag{6}$$

where:

- L^G_{adv} is the adversarial loss, encouraging the generator to produce images indistinguishable from real ones.
- L^G_{cls} is the classification loss that guides the generator to modify the image attributes correctly, making the

discriminator's classification of generated images crucial in evaluating this.
- L_{rec} is the reconstruction loss, ensuring that the generator can preserve the image's core content while manipulating its attributes.

G. Evaluating the Generator as a Classifier

To assess whether the generator can act as a classifier, we rely on the discriminator's performance as it classifies the generated images. Specifically, after the generator modifies the input images based on the target attributes, we measure how accurately the discriminator can identify these modified attributes. This evaluation is done using F1 scores, precision, recall, and accuracy metrics derived from the discriminator's classification of the generated images.

By training the generator to produce images with specified attributes and evaluating the classification performance of the discriminator, we indirectly test the generator's capacity to act as a classifier. The generator is trained to ensure that the images it generates not only look realistic but also belong to the correct domain (as classified by the discriminator). This allows us to measure the classification performance of the generated images, thereby testing the hypothesis of whether the generator can be used as a classifier.

H. Training Dynamics

The training process alternates between updating the generator and the discriminator:

Discriminator Update: The discriminator's dual role involves distinguishing between real and generated images and classifying the domain of both real and generated images. The discriminator is updated using the following loss:

$$\theta_D \leftarrow \theta_D - \eta \nabla_{\theta_D} (L^D_{\text{adv}} + \lambda_{\text{cls}} L^D_{\text{cls}}) \tag{7}$$

where:

- L^D_{adv}: Adversarial loss, which helps the discriminator differentiate between real and generated images.
- L^D_{cls}: Classification loss, which ensures the discriminator correctly classifies the domain of the images.

This allows the discriminator to accurately classify the domains of both real and generated images.

Generator Update: The generator focuses on producing images that fool the discriminator while ensuring correct classification of attributes. Its parameters are updated using:

$$\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} (L_G) \tag{8}$$

This process improves the generator's ability to manipulate image attributes correctly while maintaining the realism of the generated images.

I. Final Evaluation and Results

The discriminator's classification performance on the generated images provides an indirect measure of the generator's ability to serve as a classifier. We evaluate this performance using traditional classification metrics such as F1 score, precision, recall, and accuracy. These metrics give insight into how well the generator can modify the attributes of images, making it a potential classifier for multi-domain tasks.

By leveraging the inherent structure of StarGAN, we hypothesize that the generator can learn to produce images that are both realistic and semantically accurate according to the target attributes, as verified by the discriminator's classification performance.

III. Research content

A. Evaluating StarGAN's Generator for Classification

This research aims to explore whether the generator in StarGAN can be indirectly evaluated as a classifier by examining the discriminator's performance. The key objectives are as follows:

Leveraging StarGAN's Generator for Classification: We focus on utilizing StarGAN's generator to generate attribute-specific image modifications, which are then classified by the discriminator. The generator's role in this context is to create images with modified attributes (e.g., Eyeglasses, Bald, Smiling) that can be correctly classified by the discriminator. No modifications were made to the generator's core structure, and the standard StarGAN architecture was preserved.

Performance Evaluation: The performance of the model is evaluated by analyzing the discriminator's ability to classify generated images. We measure key classification metrics such as accuracy, precision, recall, and F1-score, focusing on whether the discriminator can successfully classify the attributes altered by the generator.

Multi-Domain Classification: We assess StarGAN's generator's ability to handle classification across multiple domains using the CelebA dataset. The model was tested on five key attributes—Eyeglasses, Mustache, Bald, Smiling, and Wearing Lipstick—without requiring any structural changes to the generator.

IV. Methodology

To evaluate the potential of StarGAN's generator for multi-domain image classification, a detailed experimental framework was implemented using two datasets: MNIST and CelebA. Experiment A focused on the MNIST dataset, whereas Experiment B utilized the CelebA dataset. This setup aimed to thoroughly examine the effectiveness and versatility of StarGAN's generator across different image domains, providing a robust analysis of its capability in multi-domain classification tasks.

A. Experimental A

MNIST: Contains 70,000 grayscale images of handwritten digits in 10 classes (0-9). The dataset is split into 60,000 training images and 10,000 test images, each of size 28x28 pixels [10]

Preprocessing:

- Normalization: Each image is normalised to have a zero mean and a unit standard deviation, which standardises the pixel values and aids in faster convergence and more stable training of the model.

- Resizing: MNIST images, originally 28x28 pixels, are resized to 32x32 pixels to meet the input size requirements of the model architecture. This resizing ensures compatibility across layers designed for 32x32 input dimensions, preserving the spatial structure necessary for effective feature extraction.

Adapting StarGAN's Generator:In this study, the original StarGAN architecture is applied to the MNIST dataset to enable both the generation of handwritten digit images and their classification into the correct digit classes. The generator in StarGAN synthesizes images conditioned on class labels (0-9), generating new images based on the target digit class. No modifications were made to the generator's architecture; instead, the original model's capability of generating images conditioned on domain labels is directly used for digit classification.

The discriminator in StarGAN performs two tasks: differentiating between real and generated images and predicting the corresponding digit class labels. The discriminator contains a classification head, which processes the features extracted from the input images through fully connected layers and outputs class probabilities. This classification head enables the discriminator to classify both real and generated images into the correct digit classes, ensuring that the model accurately recognizes and generates images that align with the specified class labels.

The training process employs a combination of loss functions that are essential for balancing both image synthesis and classification tasks. The **adversarial loss**, computed by the discriminator, plays a key role in encouraging the generator to produce images that are indistinguishable from real images. This adversarial process drives the generator to improve the quality and realism of the generated images over successive iterations by learning to ”fool” the discriminator.

In parallel, a **cross-entropy loss** is applied to the outputs of the discriminator's classification head. This loss ensures that the discriminator accurately classifies both real and generated images into the correct digit classes. The cross-entropy loss minimizes the difference between the predicted and actual class labels, helping the model to generate images that are not only visually realistic but also aligned with the correct digit class.

Additionally, a **reconstruction loss**, also known as cycle-consistency loss, is used to ensure that the generated images remain faithful to their target class throughout the synthesis process. This loss acts as a safeguard, ensuring that the generator maintains consistency between the target class and the generated images. Together, these loss functions—adversarial loss, cross-entropy classification loss, and reconstruction loss—help the generator to produce images that are both visually convincing and representative of the intended digit class, thereby achieving realism and class-consistency.

The model is optimized using the **Adam optimizer** [13], known for its efficiency in handling non-convex optimization problems, such as those typically encountered in GAN training. The learning rates are carefully tuned to balance convergence speed with training stability, and dynamic adjustments are made based on the model's performance across epochs. This approach ensures that the model effectively learns to generate realistic images while maintaining high classification accuracy.

B. Experimental B

CelebA: A large-scale dataset with over 200,000 celebrity images, annotated with 40 attribute labels. The images are carefully cropped and aligned to a uniform size of 128x128 pixels, ensuring the focus remains on the facial features. This alignment process also normalises the face orientation, which is crucial for consistent facial attribute manipulation and recognition tasks. The dataset is highly diverse, capturing various poses, expressions, and lighting conditions, making it a robust benchmark for evaluating facial attribute models. [11]

Preprocessing:

- **Image Resizing**: All images are resized to a fixed size of 128x128 pixels. This ensures that the input images have consistent dimensions, which is essential for the model to process them efficiently during training. [11]
- **Normalization**: The pixel values of the images are scaled down to the range [-1, 1]. This step involves taking the original pixel values (which range from 0 to 255) and normalizing them. Normalization helps to stabilize the training process and improve the model's performance. [11]

Data Augmentation:

- **Random Horizontal Flipping**: During training, images are flipped horizontally with a 50% chance. This technique is used to introduce more variability in the dataset, helping the model learn to recognize facial features from different orientations. [11]
- **Attribute Selection**: From the 40 available attributes in the CelebA dataset, specific attributes are selected for the training process. These attributes serve as target labels, guiding the model to focus on meaningful facial features and transformations. [11]

Adapting StarGAN's Generator: In this project,Instead of directly adapting the generator for classification, we rely on the discriminator to assess the generator's ability to classify images based on target attributes, such as Eyeglasses, Mustache, and Wearing Lipstick.

The generator's role in this experiment remains focused on producing high-quality images across different domains, as per its original design. A classification task is integrated into the training process, but it is the discriminator that plays a pivotal role in classification. The discriminator not only distinguishes between real and generated images but also predicts class labels using a classification head consisting of fully connected layers. This classification head outputs probabilities for selected attributes, ensuring the discriminator provides feedback on how well the generator captures the attribute-specific features in the generated images.

The training process involves several loss functions critical for maintaining a balance between the generator's image synthesis capability and the classification objectives:

Adversarial Loss: Encourages the generator to create images that can fool the discriminator into believing the generated images are real.

Classification Loss: Applied to the discriminator's classification head, using a cross-entropy loss to minimize the difference between the predicted and true class labels, guiding the generator to produce attribute-consistent images.

Reconstruction Loss: Ensures that the generated images remain faithful to the original input, while still reflecting the desired target attributes. By combining adversarial loss, classification loss, and reconstruction loss, we ensure that the generator synthesizes images that are both visually realistic and semantically accurate with respect to the desired attributes. This approach enables us to evaluate the generator's effectiveness in producing images that contain the specific characteristics required for classification, without directly modifying its architecture.

Optimization of the model is performed using the Adam optimizer, which is well-suited for handling the complex dynamics of GAN training [13]. The learning rate is initially set to 0.0001 for both the generator and discriminator, providing a balance between stable learning and effective convergence. After the first 100,000 iterations, the learning rate is gradually reduced in a linear fashion, allowing for fine-tuning as the model approaches convergence. Training continues for a total of 200,000 iterations, with the learning rate decay helping to prevent overfitting and ensuring that the generator adapts effectively to the new task while maintaining the ability to generate high-quality images.

V. Experimental Results

1) Performance Metrics:The adapted StarGAN generator was evaluated using key performance metrics to assess its effectiveness in classifying facial attributes.:

2) Accuracy:Accuracy is the ratio of correctly classified images to the total number of images. It provides a broad measure of model performance.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Number\ of\ Samples} \tag{9}$$

:

3) Recall:Recall, or sensitivity, is the proportion of true positive predictions out of all actual positives. It indicates how well the model captures relevant instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{10}$$

:

4) Precision:Precision is the ratio of true positive predictions to the total predicted positives. It reflects the accuracy of the model's positive predictions.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{11}$$

:

5) F1-Score:The F1-Score is the harmonic mean of precision and recall, balancing the two to provide a single metric for performance evaluation.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

: [19]

6) Confusion Matrix:The Confusion Matrix contrasts actual versus predicted classifications, providing a detailed summary of the model's performance.

$$Confusion\ Matrix = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{13}$$

:

Performance on MNIST Dataset

The adapted StarGAN generator exhibits strong performance on the MNIST dataset, achieving high classification accuracy after extensive training. After 155 epochs, visual inspection confirmed that the generator was particularly effective at producing digit images that were consistently and accurately classified by the discriminator. At this stage, the model attained a test classification accuracy of 99.97%, indicating that the generated images were nearly indistinguishable from real MNIST digits when evaluated by the discriminator. This high level of accuracy reflects the model's ability to generate visually convincing and class-consistent images, demonstrating the effectiveness of the adversarial training process in aligning the generated images with the target digit classes.

a) F1 Score, Precision, and Recall: The model's proficiency is further validated by the evaluation metrics, where precision, recall, and F1 score all peaked at 99.97% after 155 epochs. These metrics highlight the model's ability to maintain high accuracy, minimize false positives, and consistently generate images that are class-consistent. The balanced F1 scores between the generator and discriminator, as shown in the first figure, suggest that both components are well-aligned, contributing equally to the model's performance. This balance is crucial for maintaining the high classification accuracy observed.

b) Loss Analysis: The loss metrics over 200 epochs, illustrated in the fig 2, shows the training process's stability and convergence. The generator's GAN loss exhibited significant fluctuations in the initial epochs but gradually stabilized, which is typical in GAN training as the generator improves its ability to produce realistic images over time. The classification losses for both the generator and discriminator, as well as the reconstruction loss, remained relatively stable throughout the training, indicating effective learning without signs of overfitting.

c) Confusion Matrix: The confusion matrix at epoch 160, displayed in the third figure, provides a detailed view of the model's classification performance across all digit classes. The matrix shows strong diagonal dominance, indicating accurate predictions for most classes. The few misclassifications that do occur are minimal and primarily involve visually similar digits. This detailed analysis confirms the high precision and

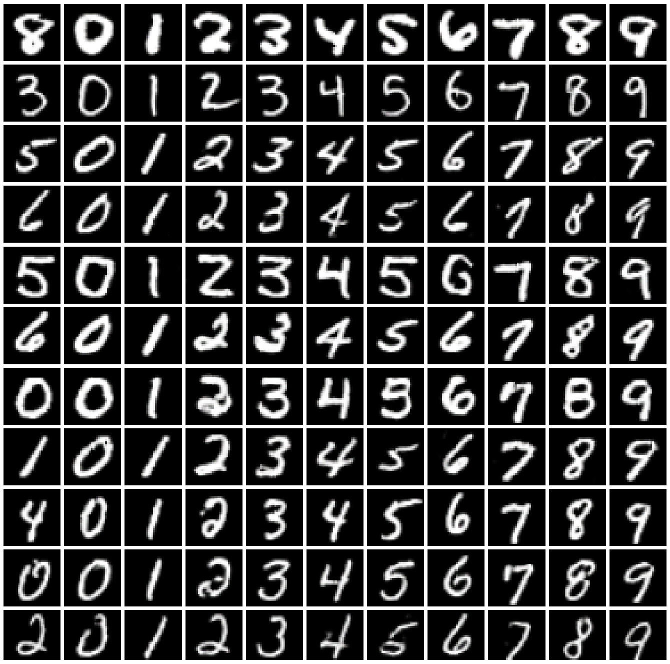

Fig. 2. StarGAN Output on MNIST Dataset. The first column contains the real images from the MNIST dataset, while the subsequent columns display the generated (fake) images conditioned on different target labels. Each row shows how the StarGAN model translates a single real image into various digits, demonstrating the generator's ability to adapt and generate realistic digits corresponding to different classes.

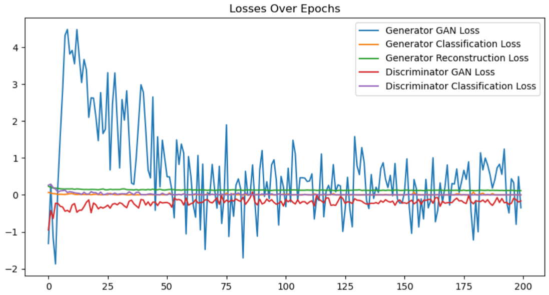

Fig. 3. Losses Over Epochs During StarGAN Training on the MNIST Dataset. The graph illustrates the behavior of different loss components over 200 epochs, including Generator GAN Loss, Generator Classification Loss, Generator Reconstruction Loss, Discriminator GAN Loss, and Discriminator Classification Loss. The fluctuations in the Generator GAN Loss decrease over time, indicating improved stability in generating realistic images, while the other losses remain relatively stable, demonstrating effective learning and convergence.

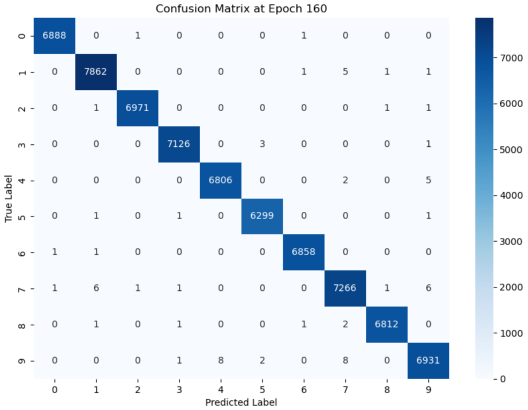

Fig. 4. Confusion Matrix at Epoch 160 for MNIST Dataset. The matrix illustrates the accuracy of predictions across different digit classes, showing strong diagonal dominance, which indicates high classification accuracy.

recall scores, further emphasizing the model's robustness in distinguishing between different classes within the MNIST dataset.

Performance on CelebA Dataset

The adapted StarGAN generator was trained extensively on the CelebA dataset, focusing on five key facial attributes: *Eyeglasses*, *Wearing Lipstick*, *Bald*, *Smiling*, and *Mustache*. The best model was achieved at iteration 280,000, after which the performance was evaluated using multiple metrics, including F1 Score, Precision, Recall, and overall Accuracy.

a) Classification Metrics

The model demonstrated robust classification performance across the selected attributes. The overall F1 Score, a measure that balances Precision and Recall, was recorded at **0.9683**, indicating strong consistency in classification. Precision, which measures the accuracy of positive predictions, was notably high at **0.9975**, suggesting that the model is highly reliable in identifying the specified attributes without producing many false positives. Recall, which indicates the model's ability to capture all relevant instances of an attribute, was slightly lower at **0.9432**, highlighting some areas where the model might miss certain features. The overall Accuracy of the model was recorded at **0.9410**, reflecting its strong ability to correctly classify images according to the targeted attributes.

b) Confusion Matrix Analysis

The confusion matrices generated for each attribute provide further insight into the model's performance. As shown in the attached figure, the matrices indicate strong diagonal dominance, meaning that the majority of predictions were correct. However, there were some misclassifications, particularly for attributes like *Bald* and *Smiling*, where visually subtle differences might have led to some confusion. This can be attributed to the inherent complexity and overlap between certain attributes, which can cause the model to occasionally

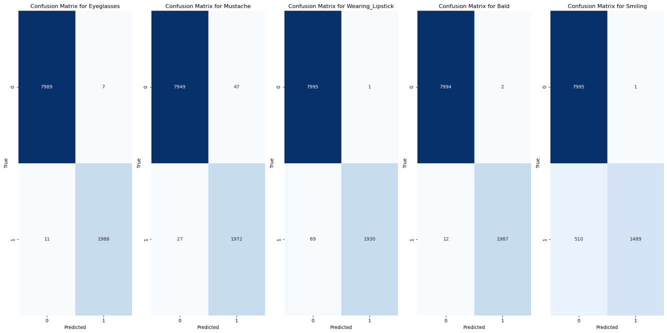

Fig. 5. Confusion Matrices for CelebA Dataset. The matrices display the model's classification performance across five facial attributes: Eyeglasses, Mustache, Wearing Lipstick, Bald, and Smiling, showing high accuracy in most categories with some misclassifications in visually subtle attributes.

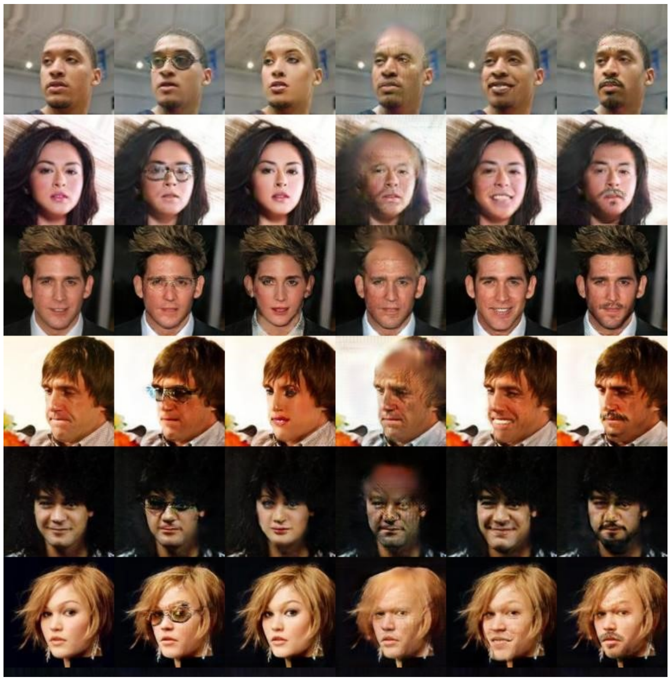

Fig. 6. Generated Output Images Using StarGAN on the CelebA Dataset. The first image in each row is the original image, while the subsequent images show transformations across various facial attributes, such as Eyeglasses, Bald, Mustache, and more. This demonstrates the model's ability to manipulate multiple attributes simultaneously while preserving the identity of the individuals.

mistake one for another. Despite these minor issues, the confusion matrices confirm the model's overall proficiency in distinguishing between the selected attributes with high accuracy.

c) Visual Inspection

Upon visual inspection of the generated images, it shows that the model successfully learns and applies the selected attributes to the input images. However, certain attributes, such as *Mustache* and *Eyeglasses*, which involve more intricate and localized changes, might not always be perfectly rendered. This is reflected in both the slight discrepancies observed in the confusion matrices and the overall Recall score. Additionally, while the Precision score suggests that the model is quite adept at avoiding false positives, the slightly lower Recall score indicates that there are instances where the model may fail to apply certain attributes consistently, particularly in cases where those attributes are subtle or overlap with others.

Hyperparameter Experimentation and Analysis

In this experiment, we aimed to explore the impact of specific hyperparameter adjustments on the performance of the StarGAN model, focusing on both the generator and the discriminator. The learning rates for both the generator and discriminator were set to 0.0002, with the beta1 parameter set to 0.4 and beta2 to 0.999, reflecting a conservative approach to learning rate adjustment. The model was trained for a total of 500,000 iterations, a substantial increase compared to previous experiments, to assess its stability and classification performance over an extended period. Throughout the training process, we observed the generator consistently outperforming the discriminator in terms of classification metrics, particularly towards the final iterations. The generator's F1 score reached as high as 1.0 in several instances, indicating that it was highly effective in capturing and reproducing the intended attributes in the generated images. Additionally, the generator maintained high precision and recall values, both peaking at 1.0, demonstrating its robustness in accurately rendering the targeted attributes without significant loss of detail or accuracy.

In contrast, the discriminator exhibited a more variable performance, struggling particularly with attribute-specific classification tasks. The discriminator's F1 score peaked at 0.4111, with precision and recall values hovering around 0.4, indicating that it was less effective at distinguishing between the generated and real attributes. This disparity between the generator and discriminator suggests that while the generator has become proficient in generating realistic and accurate attribute representations, the discriminator may require further tuning or architectural enhancements to improve its classification capabilities. The observed differences highlight the challenges inherent in balancing the performance of the generator and discriminator within the StarGAN framework, particularly when both components are tasked with different yet complementary roles. Overall, this experiment underscores the importance of careful hyperparameter tuning, not just to enhance individual component performance but to maintain an equilibrium that allows both the generator and discriminator to contribute effectively to the model's overall objectives.

VI. Discussion

A. Advantages of Using StarGAN for Classification Tasks

- **Unified Model for Generation and Classification:** The use of StarGAN's generator as a classifier provides a streamlined approach where a single model is capable of both generating high-quality images and performing accurate classifications. This reduces the need for separate models and optimizes computational resources.
- **Generative Augmentation:** StarGAN's ability to generate synthetic images has proven beneficial for augmenting

training datasets, particularly in scenarios where data is scarce or imbalanced. This augmentation capability can enhance the model's learning process and improve overall classification accuracy, especially in smaller datasets.
- **Effective Feature Learning and Flexibility:** StarGAN's generator effectively captures high-level, abstract features, which enhances its ability to differentiate between classes. This deep feature learning, combined with the model's inherent flexibility to handle various domains and tasks, makes StarGAN a powerful tool for image-based classification.

B. Performance Discrepancies Across Datasets

The performance of the adapted StarGAN model varied between the MNIST and CelebA datasets, highlighting the model's strengths and areas for improvement:

- **Exceptional Performance on MNIST:** On the MNIST dataset, which is relatively small and less complex, the adapted StarGAN model excelled, achieving near-perfect classification accuracy of 99.97%. The model demonstrated its ability to generate and classify digit images with remarkable precision, benefiting from the simplicity and clarity of the dataset.
- **Challenges with CelebA:** In contrast, the CelebA dataset presented more significant challenges due to its complexity and the intricacy of the attributes involved. Despite achieving strong overall accuracy and high precision, the model struggled with certain attributes like *Eyeglasses* and *Mustache*, which require fine detail and precise representation. The slightly lower recall and observed misclassifications in these areas suggest that the model, while robust, may require further refinement to handle more complex and diverse datasets effectively.
- **Impact of Dataset Size and Complexity:** The differences in performance between MNIST and CelebA underscore the impact of dataset size and complexity on the model's effectiveness. While StarGAN performs exceptionally well on simpler, smaller datasets, it faces challenges when applied to larger, more intricate datasets, where the complexity of features and potential class imbalances can affect the accuracy and fidelity of the generated images.

C. Future Research Directions

To address these challenges and enhance the model's performance across different datasets, future research could focus on the following areas:

- **Advanced Architectures for Complex Features:** Implementing more sophisticated model architectures or techniques such as attention mechanisms could help the generator learn and represent complex features more accurately, particularly for attributes that involve fine details.
- **Balancing and Expanding Datasets:** Ensuring that training datasets are well-balanced and diverse across all attributes can enhance the model's generalization capabilities and minimize misclassifications, especially in complex datasets such as CelebA.
- **Refining Noise and Artifact Reduction:** Developing methods to minimize noise and artifacts in the generated images could lead to clearer, more accurate visual outputs, enhancing both the visual quality and classification performance of the model.

Adapting the generator within a StarGAN framework for classification tasks presents a promising area for future research, especially in its ability to expand StarGAN's multi-domain image-to-image translation functionalities into classification tasks. Modifying the generator's architecture to incorporate class-specific feature extraction and multi-task learning could enhance its classification accuracy across multiple domains. This could involve redesigning output layers to produce classification probabilities alongside translated images, integrating attention mechanisms to focus on key features relevant to classification, and applying conditional inputs that guide both image translation and classification outputs. Adversarial training techniques, coupled with few-shot learning methods, could further improve the robustness and efficiency of the generator in classification tasks. The exploration of these modifications in StarGAN generators for classification remains an open and largely unexplored area, offering substantial opportunities for pioneering research. [6] [5] [16]

VII. Conclusion

This study demonstrates the potential of using StarGAN's generator within a multi-domain image classification framework. By leveraging the original StarGAN architecture, and utilizing the discriminator for classification, the study showcases how the generator can indirectly contribute to classification tasks without being reconfigured. Extensive experiments on the MNIST and CelebA datasets revealed that while the generator produced high-quality, attribute-specific images, it was the discriminator that effectively classified the generated images. On the MNIST dataset, the model showed excellent classification performance, with the discriminator achieving near-perfect accuracy in predicting digit classes. For the more complex CelebA dataset, the results remained strong but highlighted challenges related to the classification of intricate facial attributes and the effects of class imbalances.

This research highlights StarGAN's dual capability—not only generating high-quality, domain-specific images but also supporting precise classification through the discriminator. The findings suggest promising directions for further exploration in machine learning, where generative models can serve dual purposes in image generation and classification. Future research could refine these models, optimize training processes, and explore broader datasets to fully unlock the potential of generative networks like StarGAN in both creative and discriminative tasks.

Acknowledgments

I would like to express my sincere gratitude to Professor Dimitrios Kollias, my supervisor at Queen Mary University of

London, for his invaluable guidance, continuous support, and insightful feedback throughout this project. His expertise and dedication have been instrumental in shaping the direction and outcomes of this work. I am deeply appreciative of his encouragement and the creative input he provided, which have greatly enriched my research experience. I would like to extend my sincere gratitude to the developers of the official StarGAN code, whose work provided the foundation for this project. Their contributions to the open-source community have been invaluable in advancing research in GANs and enabling the exploration of new applications. Special thanks to the team at Clova AI Research, NAVER, particularly Donghyun Kwak, for their insightful discussions and for making their work accessible to the broader research community.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ”Generative adversarial nets,” *Advances in neural information processing systems*.

[2] A. Pathak and N. Dufour, ”Sequential training of GANs against GAN-classifiers reveals correlated ’knowledge gaps’ present among independently trained GAN instances,” *arXiv preprint arXiv:2303.15533*, 2023.

[3] H. Li, ”Use Classifier as Generator,” *arXiv preprint arXiv:2209.09210*, 2022.

[4] M. Miranda, ”Use Classifier as Generator,” *arXiv preprint arXiv:2209.09210*, 2022.

[5] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, ”Self-Attention Generative Adversarial Networks,” in *Proceedings of the 36th International Conference on Machine Learning*.

[6] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, ”StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[7] A. Radford, L. Metz, and S. Chintala, ”Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[8] X. Huang, M. Liu, S. Belongie, and J. Kautz, ”Multimodal unsupervised image-to-image translation,” *arXiv preprint arXiv:1711.06890*, 2017.

[9] Y. Shahzad, N. Akhtar, A. Mian, and M. Shah, ”Towards GAN Benchmarks Which Require Generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ”Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, 1998.

[11] Z. Liu, P. Luo, X. Wang, and X. Tang, ”Deep Learning Face Attributes in the Wild,” *arXiv preprint arXiv:1411.7766*, 2015.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, vol. 1, Cambridge, MIT Press, 2016.

[13] D. P. Kingma and J. Ba, ”Adam: A method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015.

[15] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[16] M. Mirza and S. Osindero, ”Conditional Generative Adversarial Nets,” arXiv:1411.1784 [cs.LG], 2014.

[17] L. Guarnera, O. Giudice, C. Nastasi, and S. Battiato, “Preliminary Forensics Analysis of DeepFake Images,” 2020. Accessed: Aug. 22, 2024.

[18] “Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part II [1st ed. 2019] 978-3-030-30483-6, 978-3-030-30484-3 - DOKUMEN.PUB,” dokumen.pub, 2019.

[19] Wang, Liming Zhu, Hongqin Sun, Jiawei Dai, Ran Ma, Qi, Wei, Xin. (2020). Trust Assessment in Internet of Things Using Blockchain and Machine Learning. 10.21203/rs.3.rs-110210/v1.

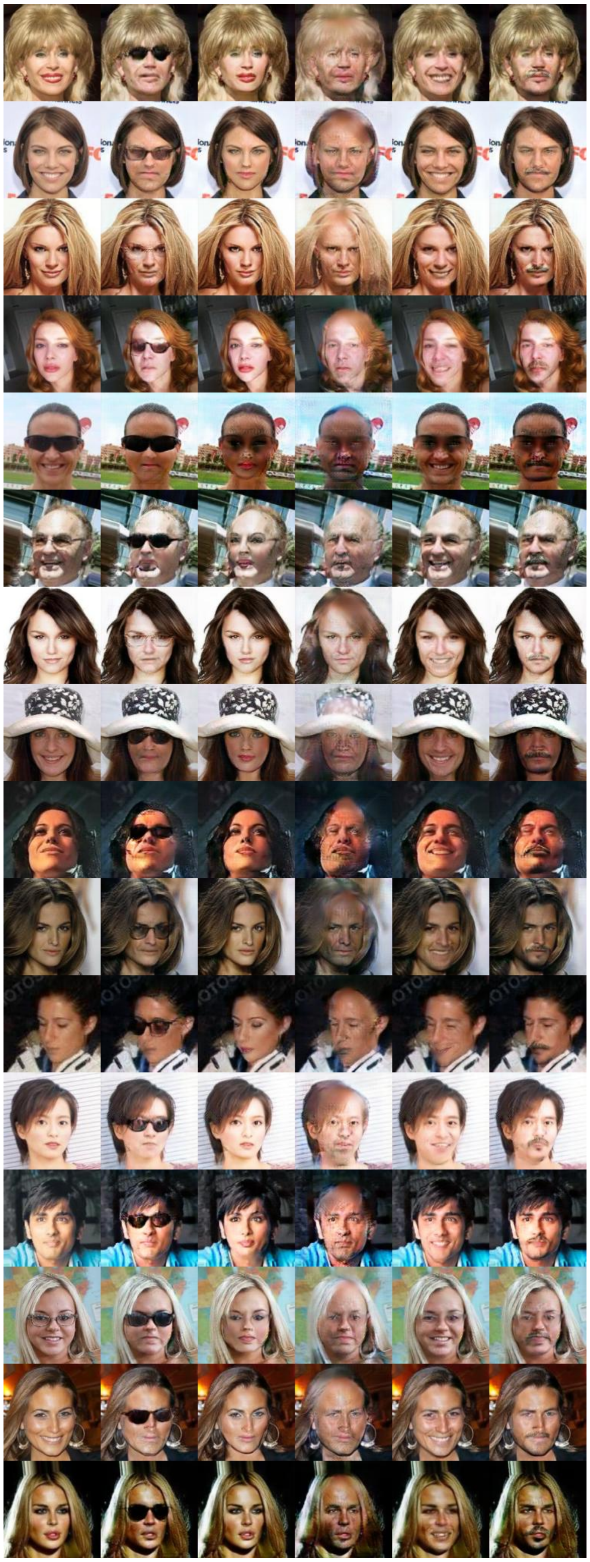

Fig. 7. Generated Output Images Using StarGAN on the CelebA Dataset at 279000th Iteration. The image showcases the transformations of various facial attributes, such as Smiling, Bald, and Wearing Lipstick, across multiple identities. These results, produced by the StarGAN model after 279,000 iterations, demonstrate the model's ability to effectively manipulate different attributes while maintaining the identity and realism of the faces.

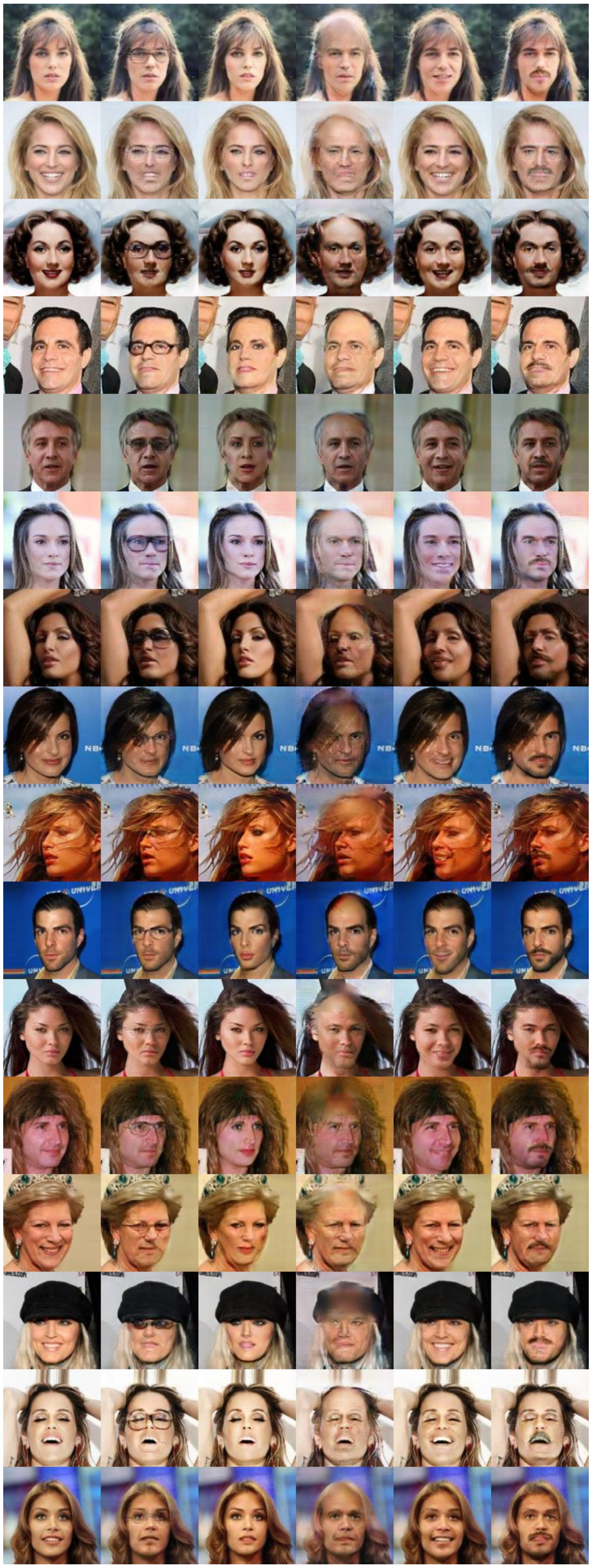

Fig. 8. Test Image Outputs Using StarGAN on the CelebA Dataset at 279000th Iteration. This image showcases the transformations applied to various facial attributes such as Smiling, Bald, and Wearing Lipstick on faces that are not directly facing forward. Despite the varied angles, the StarGAN model effectively manipulates these attributes while maintaining the identity and realism of the faces during testing.

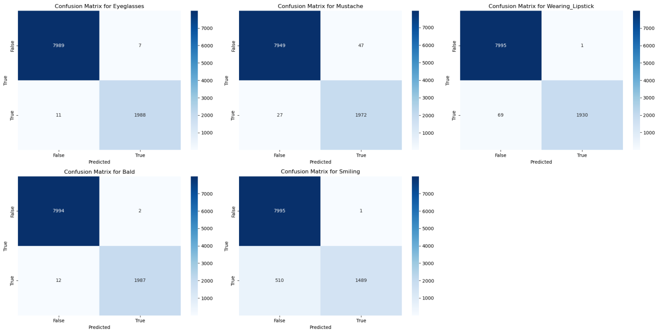

Fig. 9. Confusion Matrices for Various Attributes in the CelebA Dataset. The matrices display the classification performance for attributes such as Eyeglasses, Mustache, Wearing Lipstick, Bald, and Smiling. The results indicate the model's effectiveness in correctly classifying these attributes, with some instances of misclassification observed.

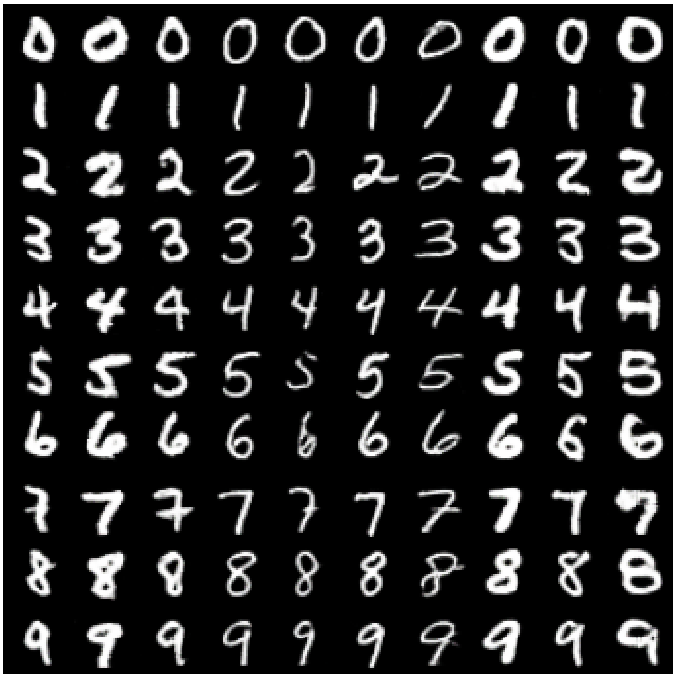

Fig. 10. Generated Images of MNIST Digits Using StarGAN. The image displays a grid of digits generated by the StarGAN model, showcasing a high level of realism. The generated digits closely resemble real MNIST digits, indicating the model's effectiveness in producing accurate and convincing outputs.