Deep Learning Project

Guilherme Vaz, Sebastião Santos Lessa

I. INTRODUCTION

In this report, we present a comprehensive exploration of deep learning techniques for addressing the problem of deep fakes. The objective is to design, implement, and evaluate both a discriminative model for classification and a generative model for synthetic image generation. This report includes model descriptions, training strategies, evaluation results, and conclusions.

II. BRIEF DESCRIPTION OF DEEP LEARNING SOLUTIONS

A. Discriminative Model

For the discriminative component of the project, we developed a model based on a Deep Convolutional Neural Network (CNN). The goal of this model is to classify images as ”real” or ”fake”. The choice of CNN is due to its ability to capture spatial and hierarchical features of images, making it ideal for image classification tasks.

- **Architecture:** The CNN is designed with a sequence of convolutional layers that gradually increase the number of filters, starting with 64 filters and reaching 512. This allows the extraction of complex features as the image is processed. Batch normalization layers and LeakyReLU activation functions are used to improve convergence and reduce the risk of overfitting.
- **Training:** The model is trained using the binary cross-entropy loss function, which is suitable for binary classification tasks, with the Adam optimizer, known for its efficiency in optimization problems in deep learning.

B. Generative Model

The generative component of the project was implemented using a neural network architecture designed to generate realistic synthetic images. The focus is on replicating the distribution of real celebrity images present in the dataset.

- **Architecture:** The generator is based on an upsampling architecture, which begins with a latent noise vector and gradually increases spatial resolution. Nearest-neighbor upsampling followed by 2D convolutions are used to avoid visual artifacts common in transposed convolution networks.
- **Training:** The generator was integrated into a GAN framework. The use of spectral normalization in convolutional layers helps stabilize training, while the final tanh activation ensures that the generated pixel values are within an appropriate range.

C. Generative Adversarial Network (GAN)

To integrate the discriminative and generative capabilities, we implemented a Generative Adversarial Network (GAN). This model consists of a generator and a discriminator trained jointly: the generator attempts to produce images indistinguishable from real ones, while the discriminator tries to differentiate them.

- **Interaction:** The adversarial training between the two models creates a dynamic where the generator is encouraged to improve its outputs to deceive the discriminator, while the discriminator becomes more proficient at identifying fake images.

III. DISCRIMINATIVE MODEL: IMPLEMENTATION STEPS AND JUSTIFICATION

A. ***Iteration 1***: Individual Training with Model-Specific Discriminators

Approach: Initially, the discriminative model was trained using separate subsets of images, each containing fake images generated by a specific deepfake model: Stable Diffusion v1.5, Stable Diffusion Inpainting, and InsightFace. This strategy aimed to tailor each discriminator to the specific characteristics of its corresponding fake image type.

We sourced these three types of images from the following dataset:OpenRL/DeepFakeFace on Hugging Face.

This strategy aimed to tailor each discriminator to the specific characteristics of its corresponding fake image type.

Model Architecture:

- **Convolutional Layers:** Multiple convolutional layers with increasing filter sizes (64, 128, 256, 512) to progressively capture complex visual features.
- **Spectral Normalization:** Applied to stabilize training and prevent exploding gradients.
- **Batch Normalization and Dropout:** Batch normalization (after all convolutional layers except the first) aids convergence; dropout (rate = 0.3) reduces overfitting.
- **LeakyReLU Activation:** Used to introduce non-linearity while avoiding the “dying ReLU” problem. Negative slope: 0.2.
- **Final Layer:** Fully connected layer followed by a sigmoid activation, outputting a probability of the image being fake.

Training Strategy:

- **Epochs:** 6 epochs per discriminator model.
- **Loss Function:** Binary cross-entropy, well-suited for binary classification tasks.
- **Optimizer:** Adam optimizer for efficient training and adaptive learning.

Motivation: This setup tests whether discriminators specialized for different generation methods can achieve better accuracy by focusing on specific image-generation artifacts.

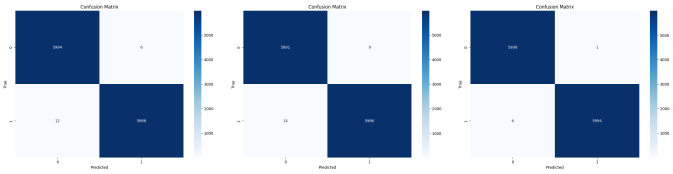

Fig. 1: Confusion Matrix - Iteration 1

Result Analysis:

- The discriminators trained individually on each type of deepfake yielded satisfactory results, demonstrating effective detection capabilities within their respective domains.
- However, given the objective of developing a single, unified discriminator, we proceeded to train a model using all three deepfake datasets combined.
- This approach aims to produce a general-purpose discriminator, capable of detecting a wider range of synthetic content in a more robust and scalable manner.

*B. **Iteration 2**: General Discriminator with Combined Training*

Approach: A single, general-purpose discriminator was trained on the complete dataset, which included all types of deepfake images. The model was fine-tuned sequentially using combinations of real data with each type of fake data. Each stage was trained for 2 epochs, resulting in a total of 10 training epochs.

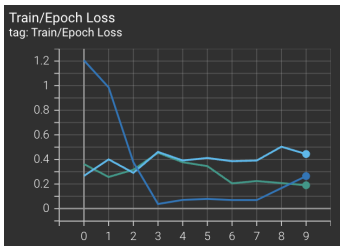

Fig. 2: Losses - Iteration 2

As shown in the figure above, the training progression of the general-purpose discriminator is illustrated, starting with the dark blue phase, followed by light blue, and finally green. Each color represents the training stages with a different deepfake dataset. A consistent decrease in loss is observed over time and across epochs, indicating effective learning throughout the sequential training process.

Model Architecture Adjustments:

- **Extended Training:** The number of epochs was later increased from 2 to 10 per model.While trying to improve generalization across all deepfake types.

Motivation: This iteration explores whether a single, unified discriminator can better generalize across a diverse range of fake images and outperform specialized models.

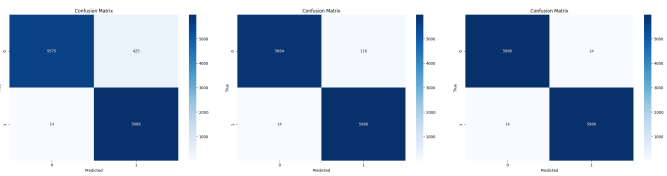

Fig. 3: Confusion Matrix - Iteration 2

Key Insights:

- The general-purpose discriminator maintained performance levels comparable to those achieved by the individual models, confirming that combining datasets did not lead to a degradation in detection capabilities.
- Importantly, the unified approach simplifies deployment by removing the need for generator-specific discriminators, offering a scalable solution for real-world applications where the type of fake content is unknown.
- Overall, these findings validate the strategy of consolidating training data to build a robust discriminator, balancing accuracy with generalization in a practical and efficient manner.

Folder	*Accuracy*	*F1-score (Macro Avg)*
inpainting	0.989167	0.989166
insight	0.963417	0.963374
text2img	0.997667	0.997667

TABLE I: Classification Performance on Different Folders

IV. Generative Model: Implementation Steps and Justification

A. Iteration 1: Initial GAN Training

Approach: The initial iteration involved training the generator within a GAN framework using real images from the dataset. The goal was to generate synthetic images that closely resemble real ones, leveraging the adversarial setup where both the generator and discriminator are trained together.

Model Architecture: **Generator Network:**

- **Latent Space:** The generator maps a latent vector (default: 512 dimensions) to a high-resolution image. The transformation starts with a fully connected layer that projects the latent vector into a low-resolution feature map.
- **Upsampling Architecture:** The generator employs nearest-neighbor upsampling followed by convolutions to increase spatial resolution. This choice avoids checkerboard artifacts that commonly appear with transposed convolutions.
- **Spectral Normalization and Batch Normalization:** Applied to convolutional layers to stabilize training and normalize feature distributions.

- **Activation Functions:** ReLU activations are used in hidden layers, while a `tanh` activation in the final layer constrains pixel values within the standard range.

GAN Framework:

- **Discriminator:** The discriminator architecture follows the previously described CNN setup and is tasked with distinguishing between real and synthetic images.
- **Training Dynamics:** GAN training alternates between updating the discriminator and the generator. The discriminator learns to classify real vs. fake images, while the generator aims to produce images that deceive the discriminator.

Training Strategy:

- **Epochs and Learning Rates:** The GAN was trained for 100 epochs with a generator learning rate of 0.0002 and a discriminator learning rate of 0.00001, ensuring a careful balance between the two networks.
- **Label Smoothing:** Introduced to prevent overconfidence in discriminator predictions and improve training stability.
- **Gradient Clipping:** Applied to maintain training stability and prevent gradient explosions.

Observations: During the initial GAN training, the generated images exhibited low quality and incoherence, indicating an imbalance in the adversarial dynamics. Instead of experiencing classic mode collapse, the generator seemed to enter a state of confusion, producing increasingly erratic outputs as training progressed. This behavior became particularly noticeable after epoch 75, when the discriminator had become highly proficient and consistently rejected the generator's outputs. As a result, the generator struggled to receive meaningful feedback and failed to converge toward realistic image generation. These observations highlight the critical need to maintain equilibrium between the capacities of the generator and the discriminator to ensure stable GAN training.

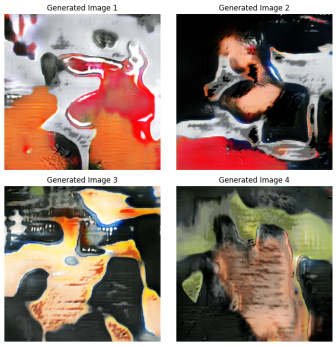

Fig. 4: Initial GAN training results

One of the challenges identified in the training process was the variability within the dataset, which included a diverse range of images rather than focusing solely on headshots. This diversity likely led to confusion for the generator, as it was tasked with synthesizing images of various types and orientations. To address this, we decided to retrain the GAN using a more homogeneous dataset, specifically the CelebA dataset, which contains only headshot images. This decision was made in the subsequent iteration to provide the generator with a more consistent target, thereby reducing potential confusion and improving the overall quality and stability of the generated outputs.

B. Generative Model: Iteration 2 with Enhanced GAN Training

Approach:

The second iteration focuses on refining the GAN training process by incorporating dynamic training strategies and architectural adjustments to improve image quality. Key enhancements include adaptive training steps, instance noise, and evaluation metrics such as the Frechet Inception Distance (FID).

Model Architecture Adjustments:

- **Reduced Image Size:** The generator now produces images at a resolution of 128x128 pixels, which allows for faster training and experimentation with architectural changes.
- **Latent Dimension:** Reduced from 512 to 256, streamlining the model's complexity while maintaining sufficient representational power to generate diverse images.
- **Generator and Discriminator Configurations:** Both models now share the same image resolution, ensuring compatibility and synchronized training dynamics.

Training Strategy Adjustments:

- **Adaptive Training Steps:** The number of steps for generator and discriminator updates is dynamically adjusted based on the current performance, allowing the model to adapt its focus on either generation or discrimination as needed.
- **Generator Steps:** Initially set to 2, can increase up to 5 if the generator's performance requires improvement.
- **Discriminator Steps:** Initially set to 1, can increase up to 3 if the discriminator's accuracy on real images needs enhancement.
- **Instance Noise:** Added to both real and fake images during training to improve the robustness of the discriminator and prevent overfitting by simulating minor variations in image data.
- **Label Smoothing:** Applied to real image labels to reduce sharp decision boundaries and enhance training stability.
- **Learning Rate Schedulers:** Both generator and discriminator have learning rate schedulers to gradually decrease the learning rate, promoting convergence and reducing oscillations in training.

Metrics and Evaluation:

- **Frechet Inception Distance (FID):** Calculated every 5 epochs to quantitatively assess the quality of generated images compared to real images, providing a benchmark for generative performance improvements.
- **Dynamic Accuracy Monitoring:** Accuracy of both real and fake classifications is tracked to guide the adjustment

of training steps and ensure balanced adversarial dynamics.

Motivation:
The updates aim to address the limitations observed in the initial iteration by introducing mechanisms for adaptive learning and robust evaluation. The goal is to produce higher-quality synthetic images that are indistinguishable from real ones, using metrics like FID to drive continuous improvements.

Results and Insights:

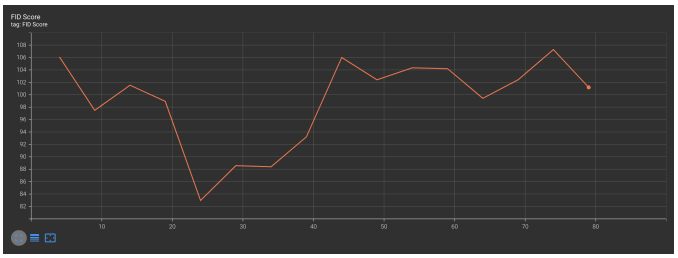

Fig. 5: FID

The FID is a metric that compares the distribution of real images to that of generated images. It works by calculating the distance between the feature distributions of real and generated images in the feature space of a pre-trained Inception model. A higher FID score indicates that the generated images are farther from the real image distribution, meaning the generated images are of poorer quality. Conversely, a lower FID score suggests that the generated images are closer to the real ones, implying higher quality.

By analyzing the FID metric throughout the training process, we can track the performance of the generator. From the results, it became evident that the quality of the generated images significantly decreased after epoch 25. This was reflected in a sharp increase in the FID score, indicating that the generated images were becoming increasingly distinct from the real ones. However, it was found that at epoch 24, the generator produced the highest quality images, as indicated by the lowest FID score during the training process.

Fig. 6: Generated Images Evolution Epochs: 14/25/81

As observed from the evolution of the epochs towards the end of the training, and with a high FID score, the generated images began to converge towards a specific image type.In this case, the generator effectively identified the patterns of images that were more likely to deceive the discriminator. As a result, it began to focus on generating images that closely matched these recognized patterns, leading to a reduction in diversity. This phenomenon is indicative of a shift towards mode collapse, where the generator learns to exploit weaknesses in the discriminator by producing a narrow range of outputs that are consistently classified as real. Unlike in the first iteration, where the generator struggled to find a reliable strategy, it eventually adapted to the discriminator's decision boundary and started to produce more consistent images.

V. Discussion and Conclusion

In this report, we have explored the application of deep learning techniques to tackle the problem of deep fakes, focusing on both discriminative and generative models. Our approach involved iterative improvements and evaluations to optimize model performance in detecting and generating synthetic images.

A. Discriminative Model

The discriminative model was implemented using a Deep Convolutional Neural Network (CNN) designed to classify images as "real" or "fake." The iterative training process involved two main strategies:

- **Individual Training with Model-Specific Discriminators:** Each discriminator was trained on subsets of images generated by specific deepfake models, achieving satisfactory results within their respective domains. This approach explored the potential for specialized models to capture unique generation artifacts.
- **General Discriminator with Combined Training:** A unified discriminator was trained using a combined dataset, demonstrating comparable performance to individual models. This approach simplified deployment and showed effective generalization across diverse fake image types.

Key Insights:

- The unified discriminator maintained robust detection capabilities without the need for specialized models, offering a scalable solution for real-world applications.
- The consistent performance across datasets supports the notion that artifact-based signals can be effectively learned in a mixed training regime, validating the strategy of consolidating training data.

B. Generative Model

The generative model was developed using a Generative Adversarial Network (GAN) framework. This involved two main iterations:

- **Initial GAN Training:** The first iteration revealed challenges such as incoherent image quality and imbalance in adversarial dynamics. This highlighted the need for equilibrium between the generator and discriminator capacities.
- **Enhanced GAN Training:** The second iteration incorporated dynamic training strategies, instance noise, and metrics such as the Frechet Inception Distance (FID) to improve image quality. Architectural adjustments and

adaptive learning steps led to significant enhancements in synthetic image realism.

Key Insights:

- The FID metric provided valuable insights into generative performance, indicating optimal image quality at epoch 24 before a decline in diversity and onset of mode collapse.
- The adjustments in training strategy and architecture demonstrated the importance of adaptive learning and robust evaluation in achieving high-quality synthetic images.

C. Conclusion

The exploration of deep learning techniques for addressing deep fake challenges has yielded promising results, particularly in the discriminative model's performance. The unified discriminator demonstrated robust detection capabilities, effectively generalizing across diverse fake image types without the need for specialized models.

However, the generative model presented more difficulties. While iterative refinements and dynamic training strategies led to improvements in image quality, several challenges persisted:

- **Adversarial Dynamics Imbalance:** Maintaining equilibrium between the generator and discriminator proved challenging. The generator often struggled to produce high-quality images when the discriminator became too proficient, highlighting the need for continuous adjustment and monitoring of adversarial dynamics.
- **Dataset Variability:** The initial diversity within the dataset contributed to confusion in the generator's outputs. Transitioning to a more homogeneous dataset, like CelebA, helped stabilize the training process, but adapting the generator to varied image types remains a complex task.
- **Mode Collapse:** Despite efforts to enhance training strategies, the generator exhibited signs of mode collapse, producing a narrow range of outputs that exploited weaknesses in the discriminator. This underscores the importance of ongoing refinement in training techniques to maintain diversity in generated images.

These challenges illustrate the complexities involved in synthetic image generation and emphasize the need for further research to enhance model adaptability and robustness. Integrating alternative architectures and evaluation metrics could drive continuous improvements, contributing to more effective solutions in combating deep fake content and advancing generative technologies.