Discriminating Deepfake detection

Avinash Vijayarangan (av3134), Chinmay Nivsarkar (cmn8525), Nishal Sundarraman (ns5429)

New York University
Github Repository: https://github.com/chinmay-n/ECE6953

Abstract

This project presents a comprehensive study that investigates the proliferation of Deepfakes, which have become a pressing issue due to advances in Generative Adversarial Networks (GANs). The paper aims to compare various existing methods for detecting Deepfakes with a proposed novel approach.The project addresses an intriguing question: Why can't we repurpose the discriminator, a crucial component in Deepfake generation, for detecting Deepfakes? The study delves into the intricacies of Deepfake creation using GANs, where the generator and discriminator engage in an adversarial competition. The study explores established techniques for Deepfake detection, scrutinizing their effectiveness and limitations. Concurrently, a novel method is proposed that leverages unique insights into the Deepfake generation process. By comparing the proposed approach with existing techniques, the efficacy of distinguishing Deepfakes from authentic content is evaluated. This Project aims to enhance the understanding of Deepfakes, uncover critical insights into their detection, and contribute to the ongoing efforts to combat misinformation. The study hopes to lay the groundwork for robust Deepfake detection systems that can mitigate the impact of falsified media and protect the integrity of digital information.

Introduction

The rapid advancement of Generative Adversarial Networks (GANs) has led to the proliferation of Deepfakes, posing a significant challenge in the battle against the spread of misinformation on the internet. Deepfakes refer to artificially generated media, typically videos, that convincingly alter or replace the appearance and actions of individuals. The rise of Deepfakes has raised concerns about the potential misuse of this technology for malicious purposes, including the dissemination of fabricated news, defamation, and social engineering attacks.

In response to this escalating issue, major companies such as AWS, Meta (formerly Facebook), and Microsoft have joined forces in a Kaggle competition to address Deepfake detection. This collaborative effort highlights the urgency and collective commitment to combatting the damaging effects of Deepfakes on digital trust and authenticity.

This project aims to provide a comprehensive study that compares various existing methods for detecting Deepfakes while proposing a novel approach. The central question motivating our investigation is the potential repurposing of the discriminator, a key component in Deepfake generation, for detecting Deepfakes. By delving into the intricate workings of Deepfake creation using GANs, we seek to gain a deep understanding of the underlying algorithmic mechanisms and architectural characteristics that pose challenges to effective detection.

Our project explores a range of established techniques for Deepfake detection, examining their strengths, limitations, and real-world applicability. Additionally, we propose a novel method that capitalizes on unique insights into the Deepfake generation process. By conducting a comparative analysis between our proposed approach and existing techniques, we aim to evaluate its effectiveness in accurately distinguishing Deepfakes from genuine content.

The findings of this project endeavor are intended to advance our understanding of Deepfakes, shed light on critical insights into their detection, and contribute to the ongoing efforts to combat misinformation. By comprehensively evaluating various detection methods and rigorously assessing the performance of our proposed approach, we endeavor to lay the foundation for robust Deepfake detection systems capable of mitigating the adverse impact of falsified media and preserving the integrity of digital information.

Related work

Deepfake detection has been an active area of research, with numerous works aiming to develop effective methods for identifying and mitigating the spread of manipulated media. Several notable studies have contributed to this field. The authors in Rössler et al. (2019) presented the FaceForensics++ dataset, which has played a crucial role in evaluating Deepfake detection methods. The research in this paper explored a wide range of traditional image forensic techniques and deep learning approaches for identifying manipulated facial images. The authors investigated various

architectures, including both handcrafted feature-based methods and deep neural networks, such as convolutional neural networks (CNNs). The results demonstrated the effectiveness of deep learning approaches in detecting manipulated facial images, highlighting the importance of utilizing advanced techniques in combating Deepfakes.

The lightweight architecture in Afchar et al. (2018) proposed MesoNet, a compact neural network architecture specifically designed for detecting facial Deepfakes. The proposed architecture aimed to balance accuracy and efficiency in Deepfake detection. MesoNet leveraged mesoscopic visual cues to differentiate between authentic and manipulated facial videos. The paper showcased promising results, demonstrating the effectiveness of MesoNet in detecting facial video forgeries, including Deepfakes.

The standard approach used in Agarwal et al. (2020) focused on automated Deepfake detection using deep learning techniques. The research explored the potential of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in identifying Deepfakes. The authors investigated different architectures and training methodologies to optimize the performance of the models. The results highlighted the potential of deep learning approaches in automated Deepfake detection, showcasing their ability to learn complex patterns and features indicative of manipulated media.

An applied deep learning approach by Li et al. (2020) contributed to the field of Deepfake detection by leveraging eye blinking inconsistencies in AI-generated fake face videos. The paper proposed a method that exploited the difficulty of replicating natural blinking patterns accurately in Deepfakes. By analyzing eye blinking cues, the authors developed a technique for identifying AI-generated fake face videos. The results demonstrated the efficacy of this approach in differentiating Deepfakes from genuine videos, showcasing the importance of considering unique characteristics and vulnerabilities in Deepfake generation.

These research papers collectively shed light on the architecture and methodologies employed in Deepfake detection. They provide insights into the efficacy of different approaches, including traditional image forensic methods, lightweight neural networks, and deep learning techniques. The findings of these papers have informed the development of novel methods and comparative analysis within the present study, contributing to the advancement of Deepfake detection and the ongoing efforts to combat misinformation.

Dataset Description

The dataset used in this project is CelebFaces Attributes Dataset (CelebA) and 1 million fake faces collected from Kaggle. CelebA is a widely used dataset in computer vision research, particularly in face-related tasks. It is a large-scale dataset that consists of over 200,000 images of 10,177 celebrity identities. The images were collected from the internet and cover a diverse range of poses, facial expressions, and lighting conditions.

Each image in the CelebA dataset is annotated with 40 attribute labels, including gender, age, facial hair, and presence of eyeglasses. These attribute labels provide valuable information for tasks such as face recognition, attribute classification, and facial attribute manipulation.

Figure 1: CelebA dataset

The 1 million fake faces collected from Kaggle contain synthetic faces which were generated using various techniques, such as Generative Adversarial Networks (GANs) and deep learning algorithms. The dataset provides a diverse range of synthetic face images, including different genders, ages, ethnicities, and facial expressions.

The synthetic faces in this dataset were created to simulate realistic human appearances but do not correspond to real individuals. They serve as a valuable resource for studying and understanding the characteristics and patterns of fake faces generated by advanced machine learning models.

Figure 2: 1 million fake faces dataset

Model Analysis

In this section, we investigate the impact of employing different backbone models in the context of deepfake detection. We aim to evaluate the performance of updated models by altering the backbone architecture and comparing them to the baseline model. Throughout our evaluation, we maintain consistent settings and configurations for the general setup of the models, unless stated otherwise. This approach allows us to assess how different backbone models influence the results and draw meaningful conclusions regarding their effec-

tiveness in deepfake detection. By exploring various backbone architectures, we gain insights into their impact on the overall performance and contribute to the advancement of robust deepfake detection systems.

DenseNet

Here, we use the DenseNet121 model as the backbone architecture for discriminating deepfake detection. DenseNet121 is a convolutional neural network (CNN) architecture that has gained popularity in various computer vision tasks, including Deepfake detection. The architecture of DenseNet121 is characterized by its densely connected layers, which facilitate feature reuse and promote efficient parameter sharing.

DenseNet121 consists of four dense blocks, each followed by a transition layer. A dense block is a collection of densely connected layers, where each layer receives feature maps from all preceding layers within the block. This dense connectivity promotes feature reuse, facilitates information flow, and allows for efficient parameter sharing.Within each dense block of DenseNet121, there are multiple convolutional layers. The exact number of convolutional layers may vary, but typically, each dense block contains several layers, such as three to five convolutional layers. The specific configuration of these layers may differ within different dense blocks.

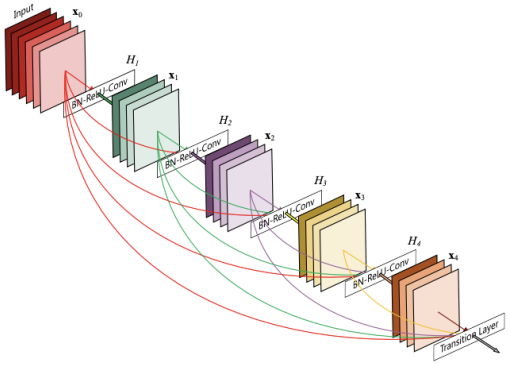

Figure 3: Densenet121 Architecture

Densenet121's architecture with dense connections and efficient parameter sharing has helped us in our Deepfake detection project. Its ability to capture intricate features, facilitate feature reuse, and reduce the number of parameters makes it an effective tool in developing accurate and efficient Deepfake detection systems.

Vanilla GAN

The Vanilla GAN architecture consists of two main components: a generator and a discriminator. The generator aims to generate realistic samples, while the discriminator aims to distinguish between real and generated samples.A commonly used variant of Vanilla GAN, known as DCGAN (Deep Convolutional GAN), typically follows a structure where the generator and discriminator consist of several convolutional layers.

The generator in Vanilla GAN typically consists of multiple transposed convolutional layers or upsampling layers, with each layer gradually increasing the resolution of the generated output. The generator starts with a low-resolution representation (e.g., noise vector) and progressively transforms it through upsampling operations, often with batch normalization and activation functions such as ReLU. The number of convolutional layers and blocks in the generator can vary depending on the complexity of the desired output.

The discriminator, on the other hand, consists of several convolutional layers that progressively downsample the input. The number of convolutional layers and blocks in the discriminator can also vary. Each layer in the discriminator typically applies convolutional operations followed by batch normalization and activation functions, such as LeakyReLU. The final layer of the discriminator outputs a probability score indicating the likelihood of the input being real or generated.

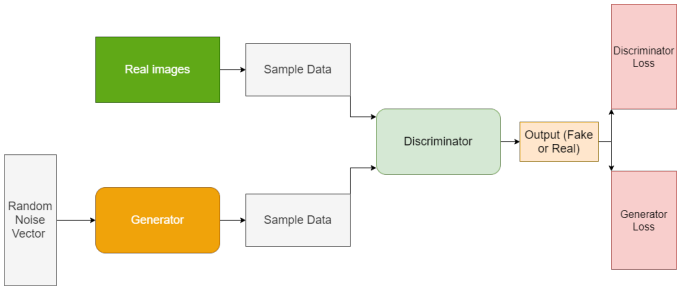

Figure 4: GAN Architecture

Methodology

The methodology of this project involved comparing a proposed method with an existing methodology to detect Deepfakes. The project consisted of two main parts.

In the first part, a pre-trained model was used to train a classifier for detecting Deepfake images. The dataset used for training the classifier included the CelebA Dataset, which contained over 200K real human images, and the 1-million-fake-faces dataset obtained from Kaggle, which consisted of over 200K Deepfake images. To leverage transfer learning, DenseNet-121 was selected as the architecture. Several reasons supported this choice, including the fact that DenseNet-121 is a densely connected convolutional neural network (CNN) with connections across layers, allowing for parameter reuse. The skip connections in DenseNet facilitated the transfer of information throughout the network during both forward and backward flows of parameters and gradients.

The project began with preprocessing the raw data from the two datasets. The data was converted into Pandas DataFrames and then classified as ”REAL” or ”FAKE”. Subsequently, the data was split into training and test datasets. To provide flexibility for the architecture to learn from the available data, the top layer of the pre-trained model was removed, and the weights of the previous layers were frozen. Custom layers were added to enhance feature learning. The model was trained for several epochs, resulting in high performance on the validation set, achieving an accuracy of approximately 99%.

The second part of the project involved training a Generative Adversarial Network (GAN) on the CelebA dataset. A GAN consists of two key components: a Generator and a Discriminator. The Generator's role is to generate near-perfect images of human faces while attempting to deceive the Discriminator. Conversely, the Discriminator aims to correctly classify the images produced by the Generator as fake. This process of generation and discrimination iterates until the Discriminator recognizes the Generator's output as real. The loss function of the Generator depends on the Discriminator's performance for the generated images. Once the GAN was able to generate near-perfect images of human faces, the training process was stopped.

Next, the Discriminator alone was extracted from the GAN architecture, and hyperparameter tuning was applied. The Discriminator was evaluated on the same test data prepared for the DenseNet-121 architecture, which also included the fake faces dataset that the model had not seen before. This evaluation aimed to assess the performance of the Discriminator against unseen data.

The criterion used for all the architectures in the project was Binary Cross-Entropy. PyTorch and TensorFlow, along with other commonly used machine learning libraries in Python, were employed to achieve the desired results. TensorFlow was used to train the DenseNet-121 architecture, while PyTorch was utilized for training the GAN. The entire project was contained within the Anaconda environment, and two different hardware pieces were utilized: NVIDIA 1060 GTX with 6GB memory and NVIDIA 1050Ti GTX with 4GB memory graphics cards. These hardware choices aimed to maximize parallelization and leverage the full potential of these devices for efficient computation.

Results

The results of the project are presented in this section, comparing the performance of two proposed methods for detecting Deepfakes. The first method involved using a pre-trained classifier with a DenseNet-121 architecture, with two subparts.

In the first subpart, the weights of all layers of DenseNet-121, except the top layer, were frozen. Additional layers were added on top to enable the architecture to learn the features of the data. This subpart achieved a high level of performance, with an accuracy of approximately 99% on the test dataset. The model successfully learned the relevant features and demonstrated strong classification capability.

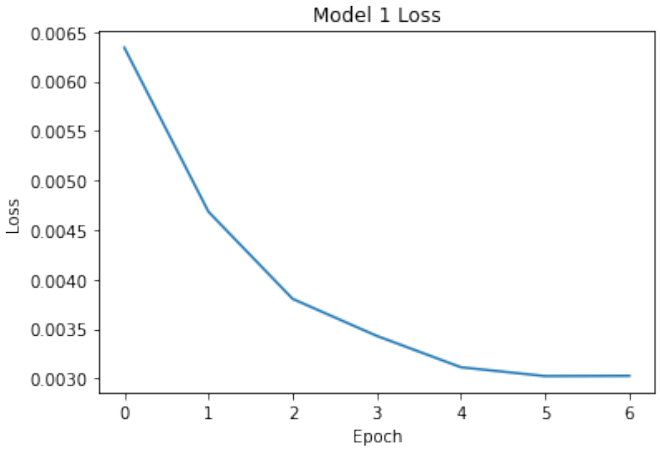

Figure 5: Transfer Learning DenseNet-121 Model Loss

In the second subpart of the first method, all the weights in every layer were unfrozen, allowing for further fine-tuning of the model. This subpart also achieved close to 99% accuracy, indicating the effectiveness of fine-tuning and leveraging the pre-trained model.

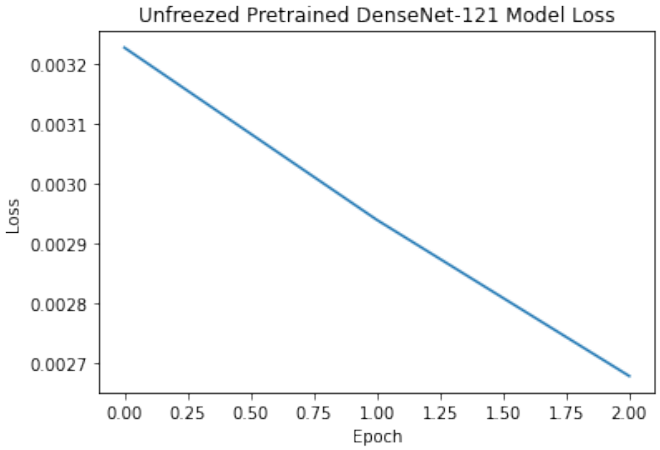

Figure 6: Unfreezed Pretrained DenseNet-121 Model Loss

Moving on to the second method, it focused on utilizing the discriminator of a GAN trained to generate Deepfakes for the purpose of detection. The GAN was initially trained to generate Deepfakes, and over a course of 30 epochs, its Loss was plotted to monitor the convergence and improvement of the training process. Once the GAN training was completed, the discriminator component of the GAN, which had learned to distinguish between real and fake images, was extracted and repurposed to act as a classifier

for Deepfake detection.

Upon evaluation, the discriminator-as-classifier achieved an accuracy of 40% on the test dataset. This lower accuracy suggests that the discriminator's ability to distinguish between real and fake images, when repurposed as a standalone classifier, was limited. The results indicate that relying solely on the GAN discriminator may not be sufficient for effective Deepfake detection.

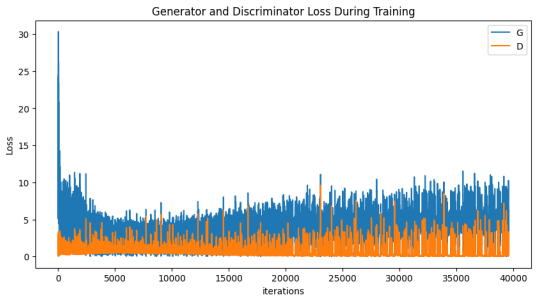

Figure 7: Generator and Discriminator Loss During Training

Overall, the pre-trained classifier using the DenseNet-121 architecture outperformed the GAN discriminator approach in terms of accuracy for Deepfake detection. The first method achieved a significantly higher accuracy of around 99%, demonstrating its effectiveness in distinguishing between real and fake images. On the other hand, the GAN discriminator's performance was less reliable, with an accuracy of 40% suggesting room for improvement in detecting Deepfakes using this approach.

Conclusion

In conclusion, our exploration of using the discriminator as a classifier for deepfake detection has revealed that it is not the optimal solution. Our research has led us to uncover several reasons for this conclusion. Firstly, we have found that the discriminator alone is not robust enough to effectively capture the intricate features present in deepfake images generated by sophisticated techniques. Deepfakes are designed to deceive human perception, and relying solely on the discriminator's capabilities may limit our ability to accurately distinguish between real and fake content.

Furthermore, the complex nature of adversarial generation in deepfakes cannot be fully comprehended by a relatively small network such as the discriminator. Deepfake techniques employ advanced algorithms and architectures that constantly evolve and become increasingly sophisticated. The discriminator, which is originally designed to differentiate between real and fake samples, may not possess the capacity to fully understand and adapt to the intricacies of these techniques.

As a result, our research suggests the need for alternative approaches and more advanced methods for deepfake detection. This could involve the development of more robust and specialized neural network architectures specifically designed for deepfake detection. Additionally, incorporating additional techniques such as feature engineering, ensemble learning, or leveraging other domains such as audio and text analysis may further enhance the accuracy and effectiveness of deepfake detection systems.

While our exploration has shown the limitations of using the discriminator as a classifier, it has also sparked further questions and avenues for future research. The ongoing battle against deepfakes necessitates continuous advancements in detection methods, and our findings contribute to the broader understanding of the challenges and complexities involved in combating the spread of misinformation through manipulated media.

Future Work

This project could focus on further fine-tuning the discriminator model to improve its ability to classify deepfake images on unseen data. While the current approach utilizes the discriminator as a classifier, additional layers could be added to the discriminator architecture to enhance its detection capabilities. By incorporating more layers and training the model with a larger and more diverse dataset, the discriminator can potentially learn more intricate patterns and features associated with deepfake images. This extended architecture would allow for more nuanced discrimination between real and fake content, leading to improved accuracy and robustness in deepfake detection. Exploring advanced techniques such as transfer learning and ensemble methods could also contribute to enhancing the performance of the discriminator in classifying deepfakes on previously unseen data. By continually refining and expanding the discriminator's architecture, researchers can strive for even better detection results in the ongoing fight against deepfake proliferation.

References

[1] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1-11).

[2] Afchar, D., Nozick, V., Yamagishi, J., Echizen, I. (2018, December). Mesonet: a compact facial video forgery detection network. In 2018 IEEE international workshop on information forensics and security (WIFS) (pp. 1-7). IEEE.

[3] Agarwal, S., Farid, H., Fried, O., Agrawala, M. (2020). Detecting deep-fake videos from phoneme-viseme mismatches. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 660-661).

[4] Li, Y., Chang, M. C., Lyu, S. (2018). In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking. arXiv preprint arXiv:1806.02877.

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[6] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[7] Pytorch documentation https://pytorch.org/docs/stable/index.html

[8] TensorFlow documentation https://www.tensorflow.org/api_docs

[9] CelebA dataset https://www.kaggle.com/datasets/jessicali9530/celeba-dataset?resource=download

[10] 1 Million fake faces on Kaggle https://www.kaggle.com/c/deepfake-detection-challenge/discussion/121173