Journal of Artificial Intelligence and Capsule Networks (ISSN: 2582-2012)
www.irojournals.com/aicn/

Gan-Enhanced Real-Time Detection of Deepfakes Videos

Amber Fatima[1], Pintu Kumar Ram[2]

Amity School of Engineering and Technology Amity University Noida Uttar Pradesh, India

Email: [1]amberfatima1303@gmail.com, [2]rampintu570@gmail.com

Abstract

Artificial intelligence is used by the deepfake mechanism to create videos that are remarkably realistic yet fraudulent, seriously undermining the legitimacy of digital media. In this study, AI techniques are used to examine real-time deepfake detection, with a particular focus on Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs). The DeepFake Detection Challenge (DFDC) dataset was used to test the models after essential features were extracted using a frame-based technique. According to the findings, GANs outperformed CNNs (83%), RNNs (85%), and other neural networks (ANNs) with an accuracy of 88%. With a 3% improvement over RNNs and a 6% improvement over CNNs in accuracy, as well as improved recall and precision measures, GANs proved to be superior generative feature learning. This proposed study demonstrates the potential of GAN-based techniques for reliable detection in difficult real-time.

Keywords: Deepfake detection, Artificial Intelligence, CNNs, RNNs, GANs, Real-time video analysis, Performance Metrics.

1. Introduction

Deepfake technology's fast development has raised a number of serious issues in the world of technology, the most significant of which is its potential misuse for producing damaging and misleading information. The sophistication of deepfakes produced by cutting-

DOI: https://doi.org/10.36548/jaicn.2024.4.005
Received: 27.10.2024, received in revised form: 21.11.2024, accepted: 05.12.2024, published: 17.12.2024

edge AI techniques like Generative Adversarial Networks (GANs) has increased, making detection strategies more important than ever. It is becoming more difficult to distinguish between real and fake media due to the increasing accessibility of tools for creating deepfake videos. This raises the possibility of abuse in situations like political manipulation, disinformation, and personal defamation [1]. This study investigates the use of AI techniques for deepfake detection, with a particular emphasis on the application of GANs, RNNs, and CNNs. These algorithms, which make use of their capacity to examine patterns in images, are at the forefront of identifying faked material as well as films. CNNs are often employed to identify gaps in visual content because of their excellent accuracy in image-processing jobs [2]. However, RNNs are good at sequence-based tasks, which means that they can be used to analyze temporal information in video frames [3]. Even though GANs are frequently linked to the creation of deepfakes, they have also been used in adversarial training to identify false content [4]. Combining these methods has produced encouraging results in recent studies for more reliable detection. For example, the system may detect visual anomalies and track discrepancies over time by merging CNNs with RNNs [5]. Moreover, by concentrating on the most enlightening segments of a video, multi-attentional networks have increased detection accuracy [6]. This research evaluates the effectiveness of these models in identifying deepfakes by contrasting their F1 score, recall, accuracy, and precision. This proposed study intends to support continuing attempts to stop the spread of deepfake content utilizing real-time datasets like DFDC. The results highlight the necessity of ongoing progress in detection technologies in order to stay up with the rapidly changing deepfake methods [7-15].

2. Literature Review

Recent years have witnessed significant improvements in the field of deepfake detection research, with several AI strategies showing potential. Afchar et al.'s early study [8] presented MesoNet, a CNN-based model that focuses on visual artifacts included in generated images to identify bogus media. Since CNN models can extract fine-grained spatial characteristics from single frames, they have emerged as a key component of deepfake detection [2]. Nevertheless, CNNs are less useful for video-based detection on their own since they have difficulty identifying discrepancies between frame sequences. In order to overcome this restriction, RNN models have been proposed. Specifically, RNNs with Long Short-Term

Memory (LSTM) capabilities function very well at identifying temporal discrepancies in videos. An RNN-based method for capturing temporal information was developed by Li et al.

[3] by examining the link between successive frames, greatly enhancing the ability to detect deepfakes in video sequences. RNNs have good temporal analytic capabilities, but training them can be challenging because they frequently need a lot of labeled data and a lot of processing power. In the case of deepfakes, GAN-based techniques serve a dual purpose: they are both crucial in producing and identifying fake content. The research [6] showed that GANs are quite good at identifying deepfakes because they can take advantage of minute artifacts that are left in created content. Additionally, as demonstrated by research in [5], who created StyleGAN to produce high-quality synthetic images and used its comprehension of artifacts for detection, GANs can be integrated with CNNs to increase detection accuracy. This study expands upon these discoveries by integrating Enhancing real-time detection of deepfake films through temporal and spatial analysis is the main goal of the CNN, RNN, and GAN models. A full overview of each model's performance may be seen in the comparative table located in the results section.

3. Problem Formulation

The key problem that this study attempts to solve is the real-time identification of deepfake films by highlighting temporal and spatial irregularities in the provided data. Although there has been extensive study on recognizing deepfakes in static images, video-based detection is still more difficult because of the complicated temporal dynamics involved. The need for real-time detection methods is emphasized by this study, particularly in fields like media authentication, cybersecurity, and distortion control [3]. The lack of original video files in the dataset was a significant challenge for the study. In order to get around this, image frame sequences were joined to create.mp4 video files, which made it possible to apply both CNN-based temporal analysis and RNN-based spatial analysis. Accurately differentiating between authentic and false videos is the main issue particularly when advanced models such as GANs are employed to produce deepfake content of superior quality.

4. Overview of Techniques

Three deep learning models were assessed for the detection of deepfake videos in this study: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Generative Adversarial Networks (GAN).

CNN: By using convolutional and pooling layers, CNNs are mainly utilized for feature extraction from visual data. This study's architecture, which is trained to identify visual artifacts in films, comprises of several convolutional layers followed by fully connected layers.

RNN: RNNs are appropriate for temporal data analysis because they take advantage of sequential data linkages. Long Short-Term Memory (LSTM) layers of the RNN model are used to evaluate video frame sequences and capture temporal dependencies to improve detection accuracy.

GAN: A discriminator and a generator make up the GAN framework. The discriminator determines if a frame is real or fake, whereas the generator produces artificial deepfake frames. For classification, the discriminator model employed in this work consists of convolutional layers followed by dense layers.

CNNs are excellent at spotting pixel-level abnormalities in individual video frames and are mostly utilized for spatial analysis. These models were optimized to identify deepfakes after being pre-trained on sizable datasets such as ImageNet [2]. RNNs, in particular LSTMs, on the other hand, concentrate on examining the temporal coherence between frames in order to identify minute changes in visual patterns across time. The temporal component is important since many deepfakes show cross-frame discrepancies that might not be immediately noticeable in single-frame analysis [4]. In this study, GANs are also utilized to generate and detect deepfakes. They can find hidden artifacts left by algorithms used to create bogus content by utilizing their capacity to recreate the generative process [6]. A mix of These methods offer a deeper, more comprehensive approach to deepfake identification. The Figures 1-3 depicts the architecture of CNN, RNN and GAN.

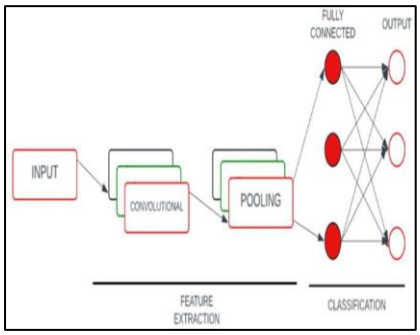

Figure 1. Architecture of CNN

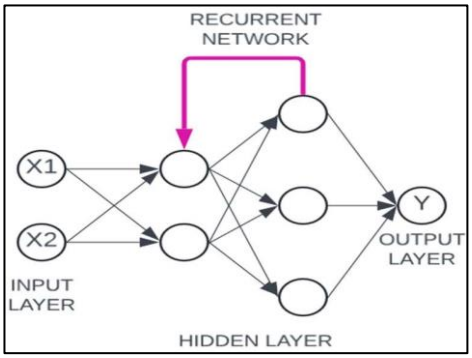

Figure 2. Architecture of RNN

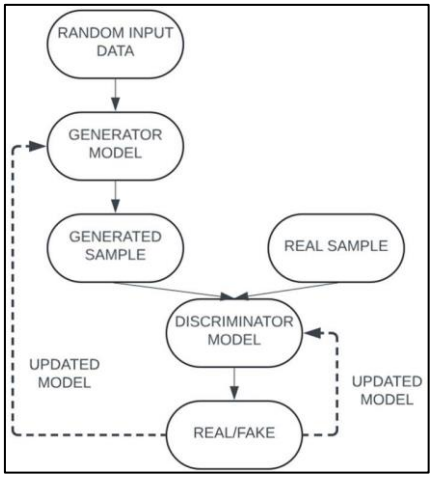

Figure 3. Architecture of GAN

5. Proposed Work

This study uses CNNs, RNNs, and GAN techniques to create a reliable deepfake detection system. This implementation approach included data collection, model training, cross-validation, and evaluation. The dataset was made up of frames that were taken from videos.

5.1 Dataset and Preprocessing

The DeepFake Detection Challenge (DFDC) dataset, which includes over 10,000 video samples of both real and false videos, was utilized in the present study. The movies were broken up into individual frames to make analysis easier. This gave temporal sequences for video-based models and static image data for spatial analysis. This preparation made sure the dataset was appropriate for testing the deepfake detection capabilities of CNN, RNN, and GAN models. As part of the preparation stages, the input data was standardized by normalizing pixel values between 0 and 1. For homogeneity throughout the collection, frames were shrunk to a standard 128 x 128 pixel size. In order to improve the models' capacity for generalization and avoid overfitting during training, data augmentation methods like zooming, random rotation, and horizontal flipping were also used. To guarantee a fair evaluation of the models, the dataset was split into three sections: 70% for training, 15% for validation, and 15% for testing.Python was used for the implementation, utilizing the PyTorch and TensorFlow frameworks. To effectively manage the computational demands of deep learning model training, a GPU-enabled system was used. Figure 4 illustrates the flowchart of evaluation of AI models.

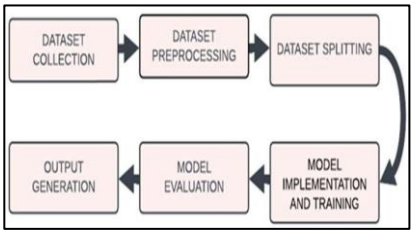

Figure 4. Flowchart of Evaluation of AI Models.

5.2 CNN Implementation

Frame-based deepfake detection was the first task tackled using Convolutional Neural Networks (CNNs). Because the raw dataset only included.jpg frames, then preprocessed the data by normalizing the pixel values and resized each frame to a resolution of 128 by 128 pixels. Utilizing 5-fold cross-validation, partitioned the dataset into training, validation, and test sets in order to guarantee the generalization of the model and minimize overfitting. The CNN model had five convolutional layers that were tuned at a learning rate of 0.001 using the Adam optimizer. It was trained with 32 batches over 50 epochs. To identify visual artifacts suggestive of deepfakes, the model extracted pixel-level information from individual frames, concentrating on spatial analysis, which was then followed by layers for maximum pooling. After being flattened, the output of the last convolutional layer was divided into fully connected layers and classified. By using data augmentation techniques including random rotation, horizontal flipping, and zoom to the input frames, overfitting was prevented and the model's performance was enhanced. The loss function employed for this model was binary cross-entropy, and the training utilized the Adam optimizer. The metrics of accuracy, precision, recall, and F1-score were recorded for each fold of the cross-validation process. Across all folds, the CNN's accuracy average was approximately 83%. A graph showing the training and validation over epochs was used to visualize the performance and show a consistent convergence. The CNN model displays a V-shaped pattern in its performance across several cross-validation folds, with accuracy and other metrics varying between high and low values. This variability is usual and represents the differences in the model's performance across different data subsets. The reason for these swings is that the data in each fold varies slightly, which causes variances in the model's generalization ability. This variability, showing the model's adaptability to various data samples and potential flaws, is a typical part of cross-validation. Figure 5 illustrates the performance of CNN model.

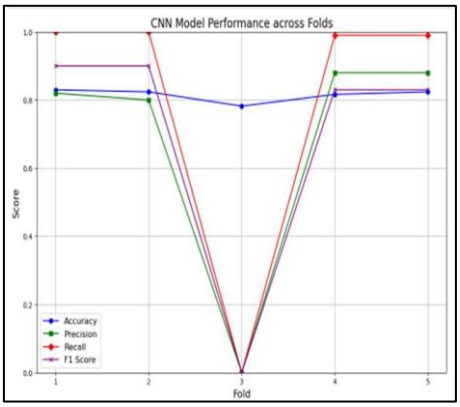

Figure 5. Performance of CNN Model

5.3 RNN Implementation

Another used technique was LSTM units to create an RNN for processing data sequentially. Manually arranged the individual frames into sequences to mimic temporal data that was similar to video because the dataset lacked complete video files. The RNN was then given these sequences, and each sequence was handled as a mini-video. Then used 5-fold cross-validation to train the RNN, much like with the CNN method. Three LSTM layers were used by the RNN model to capture temporal dependencies between 10-frame sequences. It was trained over 50 epochs using the Adam optimizer with a learning rate of 0.001. By analyzing temporal coherence, this architecture made it possible to identify minute discrepancies between video frames. In order to identify minute modifications in deepfake films, the LSTM-based model was engineered to capture temporal correlations between frames. But dealing with frame sequences presented other difficulties as well, such controlling memory consumption and sure the model learns the temporal patterns efficiently. Throughout multiple epochs, the RNN model was trained using the Adam optimizer and binary cross-entropy.

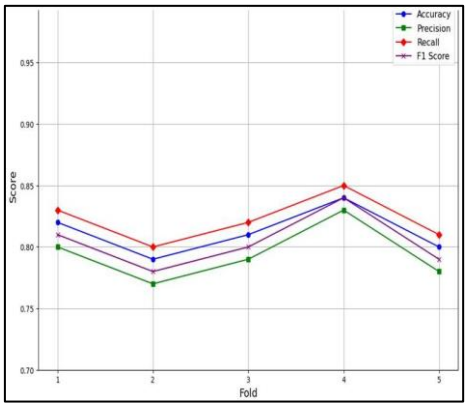

Figure 6. RNN Model Performance Across Folds

The degree of precision attained was about 85 percent. During training, early halting was used to avoid overfitting. The model's learning curve was illustrated by a performance graph that displayed the drop in training and validation loss. Figure 6 illustrates the RNN Model performance across folds.

5.4 GAN Implementation

Lastly, fictitious frames for adversarial training using a Generative Adversarial Network (GAN) was created. A discriminator that was trained to distinguish between real and fake frames made up the GAN, while a generator produced plausible fake frames as its goal. The primary tool for classification was the discriminator. In particular, training stability presented several difficulties during the GAN training procedure. In training, GANs are known for their instability, where one of the two components might overwhelm the other: the discriminator or the generator. The GAN model had a discriminator with five convolutional layers that used Leaky ReLU activation and a generator with six layers. A batch size of 32, a learning rate of 0.0002, and 100,000 iterations using the Adam optimizer were all part of the training procedure. Following adversarial training, real and produced video frames were classified using the discriminator model. Using cross-validation throughout the evaluation phase. It was able to reach an accuracy of 88%. The discriminator's accuracy with time on a graph shows how well the GAN model performed. The model could produce incredibly realistic frames that could not be identified even by trained classifiers. Here implementation approach included data collection, model training, cross-validation, and evaluation. The dataset

was made up of frames that were taken from videos. Figure 7 represents the GAN model performance across folds.

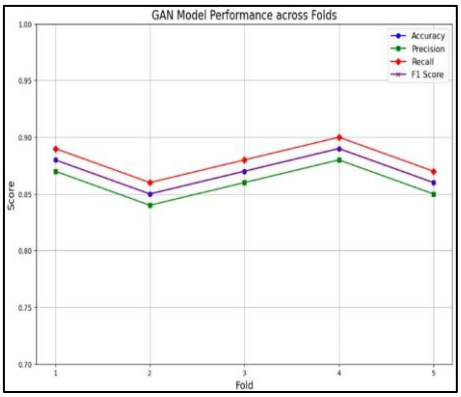

Figure 7. GAN Model Performance across Folds

6. Results and Discussion

The models' effectiveness in identifying deepfakes was assessed using common metrics such as accuracy, precision, recall, and F1-score.

Precision: Out of all the positive predictions the model makes, this metric assesses the percentage of accurately detected positive cases.

Recall: This metric assesses how well the model can recognize every important incident in the dataset.

F1 Score: Is a fair assessment of accuracy derived from the average of precision and recall.

Accuracy: By dividing the number of accurate predictions by the total number of samples, this metric calculates how accurate forecasts are overall. To underline the performance benefits of the GAN model, accuracy was given special attention.

A 5-fold cross-validation approach was used to guarantee generalization and robustness, which reduced overfitting and produced a more trustworthy assessment of the models' performance.

6.1 Simulation Setup

The first step in setting up the simulation for this study was obtaining the DeepFake Detection Challenge (DFDC) dataset, which was initially made up of multiple image frames. The images in this dataset were organized into continuous streams and saved in MP4 format in order to transform them into video sequences suitable for use with video-based models like RNNs and GANs. For video analysis models, it is essential that the frames be used in a structure like a movie, which was made possible by this conversion. Preprocessing these video sequences was the next step: frames were taken out of the videos, compressed to provide comparable dimensions throughout all input data, and normalized to scale pixel values correctly. For the data input into the models to remain consistent and of high quality, this preprocessing phase is essential. The dataset was then split up into subsets that could be utilized for various phases of model development and evaluation: training, validation, and testing. This ensured that each subset was unique. A GPU-accelerated high-performance computing system was used to run the simulation in order to effectively meet the computational needs. The models were assessed using both generated and real video frames during training, with a focus on accuracy, precision, recall, and F1 scores.

6.2 Results

The convolutional neural network (CNN) model achieved an accuracy of 83%, while the recurrent neural network (RNN) model recorded an accuracy of 85%. Despite an accuracy of approximately 88%, compared to CNNs and RNNs, GANs showed notable performance gains. The accuracy they attained was 3% better than RNNs and 6% better than CNNs. Because of the adversarial training process, GANs were able to discover tiny artifacts that CNNs and RNNs frequently missed. This made them especially useful for detecting high-quality forgeries and improving deepfake detection capabilities. The GAN model performed well better than both. Similar trends were seen in the precision, recall, and F1 scores; the GAN- based model continuously outperformed the others in all metrics. The values of accuracy and F1-score for the GAN model are almost identical, as indicated in the table (both around 0.88). As a result, in Fig. 7, the F1-score (purple line) overlaps with the accuracy (blue line), making the accuracy line less distinguishable.

Each model's positive and negative aspects are succinctly summarized in a comparison Table 1 that presents these findings. Because of its superior ability to produce realistic, high-quality samples, the GAN model outperforms CNN and RNN. Thanks to adversarial training, GANs are particularly good at identifying subtleties and complicated patterns in data, which helps them distinguish between real and deepfake content. As a result, GANs outperform conventional CNN and RNN models in deepfake detection, providing higher accuracy, precision, recall, and F1 scores.

Table 1. Performance Comparison of Models

Model	Accuracy	Precision	Recall	F1 Score
GANs	0.88	0.845	0.870	0.881
RNNs	0.85	0.815	0.810	0.810
CNNs	0.83	0.850	0.845	0.845

7. Conclusion

The study shows that, particularly in video sequences, GAN-based models currently provide the greatest performance for real-time deepfake identification. While CNNs and RNNs were used to detect spatial and temporal discrepancies, GANs were shown to be more effective at identifying the minute artifacts found in deepfakes. Converting image sequences into video format was a successful way to address data management difficulties, like the lack of native video files. Subsequent research endeavours may investigate hybrid models that include CNN and RNN to enhance detection performance.

References

[1] Dolhansky, B., Howes, R., Pflaum, B., Baram, N., & Canton-Ferrer, C. (2019). The Deepfake Detection Challenge (DFDC) Preview Dataset. ArXiv, abs/1910.08854.

[2] Guarnera, Luca, Oliver Giudice, and Sebastiano Battiato. "Deepfake detection by analyzing convolutional traces." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops,2020. 666-667

[3] Li, Y. "Exposing deepfake videos by detecting face warping artif acts." arXiv preprint arXiv:1811.00656 (2018).

[4] Zhao, Hanqing, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. "Multi-attentional deepfake detection." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021. 2185-2194.

[5] Agarwal, Shruti, Hany Farid, Ohad Fried, and Maneesh Agrawala. "Detecting deep-fake videos from phoneme-viseme mismatches." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020. 660-661.

[6] Zhou, Yipin, and Ser-Nam Lim. "Joint audio-visual deepfake detection." In Proceedings of the IEEE/CVF International Conference on Computer Vision, 021. 14800-14809.

[7] Hou, Xianxu, Linlin Shen, Ke Sun, and Guoping Qiu. "Deep feature consistent variational autoencoder." In 2017 IEEE winter conference on applications of computer vision (WACV), IEEE, 2017. 1133-1141.

[8] Afchar, Darius, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. "Mesonet: a compact facial video forgery detection network." In 2018 IEEE international workshop on information forensics and security (WIFS), IEEE, 2018. 1-7.

[9] Chen, Tianxiang, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. "Generalization of Audio Deepfake Detection." In Odyssey, 2020. 132-137.

[10] Grm, Klemen, Vitomir Štruc, Anais Artiges, Matthieu Caron, and Hazım K. Ekenel. "Strengths and weaknesses of deep learning models for face recognition against image degradations." Iet Biometrics 7, no. 1 (2018): 81-89.

[11] Kwon, Patrick, Jaeseong You, Gyuhyeon Nam, Sungwoo Park, and Gyeongsu Chae. "Kodf: A large-scale korean deepfake detection dataset." In Proceedings of the IEEE/CVF international conference on computer vision, 2021. 10744-10753.

[12] Zhao, Tianchen, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. "Learning self-consistency for deepfake detection." In Proceedings of the IEEE/CVF international conference on computer vision, 2021. 15023-15033.

[13] Lyu, Siwei. "Deepfake detection: Current challenges and next steps." In 2020 IEEE international conference on multimedia & expo workshops (ICMEW), pp. IEEE, 2020. 1-6.

[14] Güera, David, and Edward J. Delp. "Deepfake video detection using recurrent neural networks." In 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS), IEEE, 2018. 1-6.

[15] Dhar, Akanksha, and Ekansh Agrawal. "Detecting AI-Generated Deep Fakes Using ResNext CNN and LSTM-Based RNN: A Robust Approach for Real-Time Video Manipulation Detection." In International Conference on Cryptology & Network Security with Machine Learning, Singapore: Springer Nature Singapore, 2023. 543-554.