Deepfake Image Detection Using CNN

Dr. Nandini.C
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management
Bengaluru, India

Prof. Mamatha A
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management
Bengaluru, India

Shashank Kumar E
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management
Bengaluru, India

Tejas N Yadav
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management
Bengaluru, India

Jainath Y S
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management
Bengaluru, India

Vinuta R
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management
Bengaluru, India

ABSTRACT

The rapid advancement of computational power has significantly strengthened deep learning algorithms, making it easier to generate highly realistic AI-synthesized videos, commonly known as deepfakes. These manipulated videos pose serious threats, including political misinformation, fake terrorism activities, cyber harassment, blackmail, and identity fraud. The ability to create indistinguishable fake videos raises concerns regarding the credibility of digital media and the potential for misuse in various fields such as journalism, social media, and law enforcement.

In this paper, we propose a novel deep learning-based approach for effectively distinguishing AI-generated deepfake videos from real ones. Our method detects both reenactment and replacement deepfakes by leveraging Artificial Intelligence (AI) against AI-based manipulations. Specifically, we utilize a ResNeXt Convolutional Neural Network (CNN) to extract frame-level features, which are then processed using a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) to analyze temporal inconsistencies and classify manipulated content. By integrating CNN-based feature extraction with RNN-based sequential analysis, our model effectively captures spatial and temporal discrepancies, improving detection accuracy.

Furthermore, we demonstrate the effectiveness of our method through extensive experiments on benchmark datasets, showcasing its ability to generalize across different deepfake techniques. Our results indicate that even subtle distortions in facial expressions, inconsistencies in eye blinking patterns, and unnatural transitions between frames can be effectively detected. The proposed approach is computationally efficient and can be integrated into real-world applications, including social media moderation, forensic analysis, and cybersecurity systems.

By presenting a robust and scalable solution, this research contributes to the growing need for automated deepfake detection systems, ensuring the integrity of digital content in an era where AI-generated media is becoming increasingly prevalent

.Keywords—Res-Next Convolution neural network, (RNN)Long Short Team Memory(LSTM).Computer vision

I. INTRODUCTION

With the continuous evolution of social media platforms, deepfakes have emerged as one of the most significant threats posed by artificial intelligence (AI). These highly realistic, face-swapped videos can be misused in numerous ways, including creating political instability, fabricating terrorist attacks, enabling blackmail, and promoting revenge pornography. Notable examples, such as fabricated celebrity deepfake videos, illustrate the growing concerns surrounding digital manipulation and its ethical implications. As a result, distinguishing between deepfake and authentic videos has become a crucial task in maintaining the integrity of digital content.

Deepfake videos are primarily generated using AI-based tools such as FaceApp and Face Swap, which rely on pre-trained neural networks, including Generative Adversarial Networks (GANs) and autoencoders, to create these synthetic videos. Our approach employs a Long Short-Term Memory (LSTM)-based neural network for sequential temporal analysis of video frames while leveraging a pre-trained ResNeXt Convolutional Neural Network (CNN) to extract frame-level features. The ResNeXt CNN efficiently learns frame-specific characteristics, which are then utilized to train the LSTM-based Recurrent Neural Network (RNN) for accurate deepfake detection.

To enhance real-world applicability and improve model performance under real-time conditions, we trained our system using an extensive dataset comprising FaceForensics++, the Deepfake Detection Challenge dataset, and Celeb-DF. Additionally, we developed a user-friendly front-end application that allows users to upload videos for automated deepfake detection. Once processed, the model classifies the video as either authentic or manipulated and provides a confidence score for the prediction.

Deepfake images and videos exhibit subtle inconsistencies that are often imperceptible to the human eye but can be detected through computational methods. The quality of these manipulated videos largely depends on the sophistication of the generative model used,

with advanced GAN-generated deepfakes appearing nearly indistinguishable from real footage. However, certain artifacts, such as blurring, unnatural skin textures, irregular reflections, and facial misalignments, can serve as key indicators for detection.

One of the primary challenges in deepfake detection arises from the continuous advancement of generative models. Earlier deepfake techniques exhibited noticeable artifacts, such as blurred edges and inconsistent lighting, making them easier to detect. However, modern AI-driven deepfake generators produce highly realistic outputs that require advanced detection strategies. CNN architectures, which are highly effective in processing image data by leveraging spatial relationships between pixels, play a crucial role in identifying manipulated content.

As AI and deep learning technologies continue to advance at an unprecedented pace, their synthetic counterparts—deepfakes—are becoming increasingly realistic and widespread. AI-generated manipulated videos pose serious risks to privacy, security, and the spread of misinformation. From political propaganda to identity fraud, deepfakes have raised concerns across various domains, emphasizing the urgent need for reliable and efficient detection methods. Our proposed model aims to address these challenges by providing an effective AI-driven solution to identify and mitigate the growing threat of deepfake technology.

II. LITERATURE REVIEW

Face Distortion Artifacts [15] introduced a method to identify deepfake artifacts by comparing synthetic facial regions with their surrounding context using a specialized Convolutional Neural Network (CNN) model. This approach identified two types of facial artifacts, based on observations that modern deepfake algorithms generate images at limited resolutions before further processing them to align manipulated faces within the source video. However, their method does not take into account the temporal analysis of frames, which is crucial for robust detection.

Eye Blink Detection [16] presents an alternative technique that leverages eye blinking as a key indicator to classify videos as either deepfake or authentic. The approach involves a temporal analysis of cropped eye-blinking frames, using a Long-term Recurrent Convolutional Network (LRCN). However, as deepfake generation techniques have advanced, the absence of eye blinking is no longer a sole indicator of deepfake content. Other visual inconsistencies, such as unnatural teeth rendering, facial wrinkles, or missing eyebrows, must also be considered to enhance detection accuracy.

Capsule Networks for Manipulated Image and Video Detection [17] propose a technique using capsule networks to identify digitally manipulated images and videos in various scenarios, including replay attack detection and computer-generated content identification. However, their approach includes random noise during training, which can negatively impact real-time performance. While their model performed well on their dataset, it may yield inaccurate results when applied to real-world data due to the presence of noise during training. Our proposed method, in contrast, is designed to be trained on clean and real-time datasets for improved reliability.

Recurrent Neural Network (RNN) for Deepfake Detection [18] applies RNNs for sequential frame processing alongside an ImageNet pre-trained model. Their approach utilizes the HOHO dataset [19], which consists of only 600 videos. The small dataset size and lack of diverse video samples limit the model's ability to perform well on real-world data. Our model, on the other hand, is designed to be trained on a large-scale dataset that includes a vast amount of real-time data, ensuring better generalization and robustness in deepfake detection.

FakeCatcher is another method capable of detecting fake content with high accuracy, regardless of the generator, content type, resolution, or video quality. However, its results are affected by the lack of a strong discriminator, leading to difficulties in preserving biological signals. Developing a differentiable loss function based on the proposed signal processing steps remains a significant challenge, making accurate detection a complex task.

Our proposed method aims to address these challenges by leveraging AI-driven deepfake detection techniques that integrate spatial and temporal analysis for improved accuracy and reliability. By training on extensive real-world datasets and incorporating multiple detection parameters, our approach seeks to provide a more comprehensive solution for identifying manipulated digital content.

III. ARCHITECTURE

Framework Architecture

Our PyTorch-based deepfake detection model has been trained on an equal number of real and manipulated videos to prevent any potential bias. The architectural framework of our model is illustrated in the figure. During the development phase, we curated a dataset, preprocessed it, and generated a newly refined dataset that consists exclusively of face-cropped videos.

The core concept behind our architecture leverages the ability of Convolutional Neural Networks (CNNs) to learn distinct features that differentiate real images from deepfake-generated ones. The framework harnesses the power of deep neural networks to analyze textural inconsistencies, irregular facial structures, and subtle anomalies that traditional forensic tools struggle to detect.

The theoretical foundation of our system incorporates essential components such as model generalization, adversarial robustness, and interpretability—critical factors for ensuring effective deepfake detection in real-world applications.

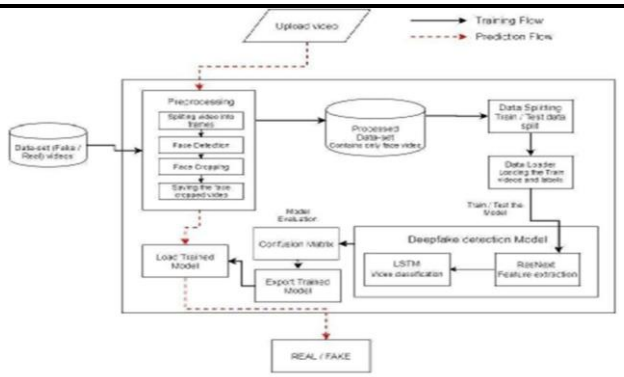

Figure 1: System Architecture

Creating Deepfake Videos

Understanding the process of deepfake video generation is essential for effective detection. Most deepfake synthesis techniques, including those based on Generative Adversarial Networks (GANs) and autoencoders, utilize a source image and a target video as inputs. These methods deconstruct the video into individual frames, identify facial regions, and seamlessly replace the target face with the source face while preserving natural expressions and movements, making detection more challenging.

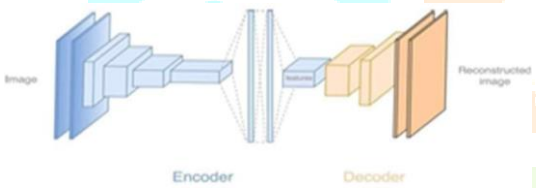

Figure 2: Deepfake generation

Architectural Design

1. Dataset Collection

To enhance the effectiveness of our model for real-time predictions, we gathered data from multiple publicly available datasets, including FaceForensics++ (FF) [1], the Deepfake Detection Challenge (DFDC) [2], and Celeb-DF [3]. We then merged these datasets to create a new, refined dataset tailored for accurate and real-time detection across various video types. To prevent training bias, we maintained a balanced dataset comprising 50% real and 50% deepfake videos.

The Deepfake Detection Challenge (DFDC) dataset [3] includes certain audio-altered videos, which fall beyond the scope of this research. To ensure dataset relevance, we preprocessed the DFDC dataset by filtering out videos with modified audio using a Python script. After preprocessing, we selected 1,500 real and 1,500 fake videos from the DFDC dataset, along with 1,000 real and 1,000 fake videos from the FaceForensics++ (FF) [1] dataset, and 500 real and 500 fake videos from the Celeb-DF [3] dataset. This resulted in a final dataset consisting of 3,000 real videos and 3,000 deepfake videos, totaling 6,000 videos. Figure 2 illustrates the data distribution across these datasets.

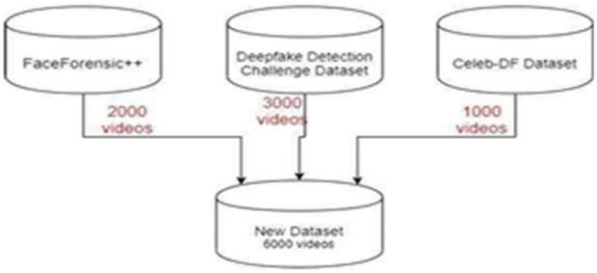

Figure 3: dataset

2. Preprocessing

During the preprocessing stage, videos undergo refinement to eliminate unnecessary noise and retain only the essential portions, specifically the facial regions. The first step involves segmenting the video into individual frames. Once divided, each frame is analyzed to detect faces, and the identified facial region is cropped accordingly. These cropped frames are then recombined to reconstruct a new video containing only the facial portions. This process is repeated for every video, resulting in a processed dataset consisting exclusively of face-focused clips. Any frame without a detectable face is discarded during preprocessing.

To maintain consistency in the number of frames, a threshold value is applied based on the average frame count across all videos. This step is necessary to optimize computational efficiency, as processing lengthy videos with high frame rates can be extremely resource-intensive. For example, a 10-minute video recorded at 30 frames per second (fps) would contain approximately 18,000 frames, making real-time processing infeasible. Given our Graphics Processing Unit (GPU) constraints, we set a threshold of 150 frames per video.

For storage in the new dataset, only the first 150 frames of each video are retained. To ensure the correct implementation of Long Short-Term Memory (LSTM) networks, these frames are arranged sequentially rather than randomly selected. The newly generated videos are saved with a frame rate of 30 fps and a resolution of 112 × 112 pixels.

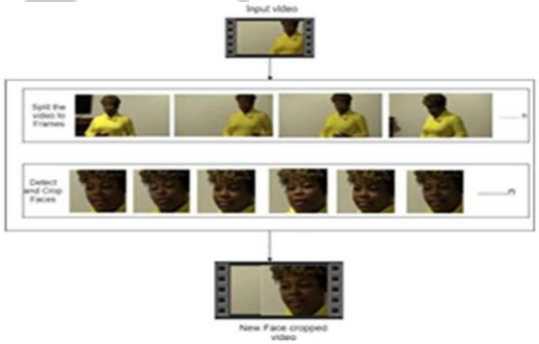

Figure 4- Pre-processing of video

3. Dataset Splitting

The dataset is divided into training and testing sets in a 70:30 ratio, ensuring a balanced distribution of real and deepfake videos in both

subsets. Specifically, 4,200 videos (70%) are allocated for training, while the remaining 1,800 videos (30%) are reserved for testing. Each subset maintains a 50:50 split between real and fake videos to prevent bias and ensure robust model evaluation.

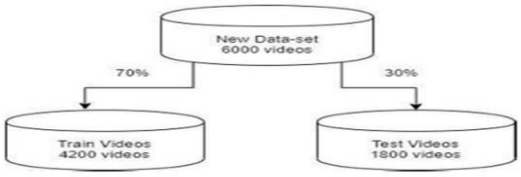

Figure 5- train test split

IV. METHODOLOGY

1. Dataset Preparation

The model is trained using datasets such as FaceForensic++, Deepfake Detection Challenge (DFDC), and Celeb-DF. The dataset consists of 6,000 videos (3,000 real, 3,000 fake), ensuring a balanced distribution.

Steps:

- Extract frames from videos and detect faces using OpenCV.
- Crop and process the first 150 frames per video for uniformity.
- Convert cropped frames back to video format (112x112 resolution, 30 fps).
- Split the dataset: 70% for training, 30% for testing, ensuring equal real and fake distributions.

2. Model Architecture

The system integrates ResNext CNN for feature extraction and LSTM-based RNN for temporal sequence analysis.

Components:

- ResNext CNN: Extracts 2048-dimensional feature vectors from video frames.
- LSTM Layer: Single LSTM layer with 2048 hidden units and 0.4 dropout to capture temporal inconsistencies.
- Additional Layers: Leaky ReLU activation, linear mapping, and Softmax for confidence scoring.

3. Training & Optimization

- Loss Function: Cross Entropy Loss.
- Hyperparameters: 20 epochs, Adam optimizer (learning rate: 1e-5), batch size: 4.
- Evaluation: Confusion matrix used to compute accuracy.

4. Model Performance

Accuracy on different datasets:

- FaceForensic++ (100 frames): 97.76%
- Celeb-DF + FaceForensic++ (100 frames): 93.98%
- Combined Dataset (20 frames): 87.79%

5. Deployment

The model is deployed as a Django web application with the following workflow:

- Users upload a video via the web interface.
- The system preprocesses the video (face detection, cropping, frame extraction).
- The model classifies the video as real or deepfake and provides a confidence score.

6. Testing & Validation

Functional Testing: Ensures correct system operation (e.g., invalid file uploads trigger error messages).
Non-Functional Testing:

- Performance Testing: Evaluates processing efficiency.
- Load Testing: Assesses system stability under high traffic.
- Compatibility Testing: Verifies operation across different devices and platforms.

7. Future Enhancements

- Browser Plugin: Extending the model to a browser extension for real-time detection.
- Full-Body Deepfake Detection: Expanding detection beyond facial alterations.
- Social Media Integration: Embedding the model into platforms like WhatsApp and Facebook for automatic detection.

8. Tools & Technologies

- Languages: Python 3, JavaScript
- Frameworks: PyTorch (deep learning), Django (web development)
- Libraries: OpenCV, Face-Recognition, NumPy, Matplotlib, Scikit-learn
- Cloud Services: Google Cloud Platform (GCP)
- Development Environments: Google Colab, Jupyter Notebook, Visual Studio

VI. CONCLUSION

In this study, we explored a neural network-based approach for classifying videos as either deepfake or authentic, along with the confidence level of the prediction. Our method achieves high accuracy by processing one-minute videos (at 10 frames per second).
The model is built using a pre-trained ResNext CNN for frame-level feature extraction and an LSTM network for temporal sequence analysis, enabling it to detect variations between consecutive frames (t and t-1). Additionally, it is capable of processing videos with frame sequences of 10, 20, 40, 60, 80, and 100 frames, ensuring robust detection.

VII. REFERENCES

1. Nagaraj, G., & Channegowda, N. (2024). Video Forgerym Detection using an Improved BAT with Stacked AutoEncoder Model. Journal of Advanced Research in Applied Sciences and Engineering Technology, 42(2), 175– 187.

2. Girish, C., & Nandini, C. "Detection of Frame Duplication Using Multi-Scale Local Oriented Feature Descriptors." International Journal of Intelligent Engineering Systems,Vol.14,No.3,2021.[DOI:10.22266].

3. N. Girish, C. Nandini "Inter-Frame Video Forgery Detection Using UFS-MSRC Algorithm and LSTM Network. "International Journal of Modelling, Simulation, and Scientific Computin(2023).[Q4][DOI:https://doi.org/10.1142/S1793962323410131]

4. Jahnavi, S., & Nandini, C. "Secure Two-Fold Transmission of Features Using Multimodal Mask Steganography and Naive- Based Random Visual Cryptography System. "International Journal of System Assurance Engineering and Management (2022). [Scopus Indexed, Springer Q2] [DOI:https://doi.org/10.1007/s13198-022-01701-6]

5. Jahnavi, S., & Nandini, C. "Hybrid Hyper Chaotic Map with LSB for Image Encryption and Decryption." Scalable Computing: Practice and Experience, Volume 23, Issue 4,pp.181191,Dec22,2022[ScopusQ2][DOI:10.12694/scpe.v23i4.2018]

6. Merikapudi, S., Math, S., Nandini, C., & Rafi, M. "A Google Net Assisted CNN Architecture Combined with Feature Attention. Blocks and Gaussian Distribution for Video Face Recognition and Verification." International Journal of Electrical Engineering and Technology (IJEET), Volume 12,Issue 1, January 2021, pp. 30- 42. [DOI: IJEET_12_01_004]

7. Nandini, C., & Shiva Sumanth Reddy. "Detection of Communicable and NonCommunicable Disease Using Lenet- Bi-LSTM Model in Pathology Images." International Journal of System Assurance Engineering and Management, Springer India, 2022. [Q3] [DOI: 10.1007/s13198-022-01702-5]

8. Girish, C., & Nandini, C. "Detection of Frame Duplication Using Multi-Scale Local Oriented Feature Descriptors."International Journal of Intelligent Engineering Systems, Vol.14, No.3, 2021. [DOI: 10.22266] find laser light in frames of video:

9. Manasa Sandeep, C. Nandini. "An Extensive Survey on 3D Face Reconstruction Based on Passive Method." InternationalResearch Journal of Engineering and Technology (IRJET),Volume 8, Issue 12, December 2021.

10. Andreas Rossler, Davide Cozzolino, Luisa Verdoliva,Christian Riess, Justus Thies,Matthias Nießner,"FaceForensics++: Learning to Detect Manipulated Facial Images" in arXiv:1901.08971. 11. J. Thies et al. Face2Face: Real-time face capture 33333333333 and reenactment of rgb videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2395,june