Deepfake Detection using Efficientnet-B0 and GRU

G Harshit, M Chakradhar Reddy, Y.V.S Sarath
Department Of Computer Science and Engineering
Geethanjali College Of Engineering and Technology
Hyderabad, India

***Abstract*— Deepfake videos, based on advanced techniques in AI, create deceptive media that seriously undermine information integrity and erode public trust. The project aims to build an effective deepfake detection mechanism by coordinating EfficientNetB0 for spatial feature extraction with Gated Recurrent Unit (GRU)-based temporal sequence modeling. EfficientNetB0 is being used as it has proven to provide high performance with efficiency, processing each video frame to extract fine spatial features for detecting subtle visual artifacts common to many deepfake attempts. GRU is chosen for efficient processing of sequential data so that temporal inconsistencies between frames can be detected without putting too much computational burden on the system. The whole deepfake detection framework essentially converts the high-performance spatial extraction from EfficientNetB0 and GRU efficient sequence modeling into an effective detector by addressing frame-level inconsistencies and temporal anomalies in manipulated videos.**

***Keywords*—Deepfake Videos;Deepfake Detection System; EfficientNetB0; Spatial Feature Extraction; GRU; Temporal Sequence Modeling; Sequential Data**

I. INTRODUCTION

Emergence of technologies like deep learning has revolutionized evolution in every field, and image processing is also no way far behind. Amongst all, the field in which deepfake technology is getting popular because of its dual impact was on making the media very realistic-synthetic, through the use of which effective entertainment and education could be done. Deepfake also has severe security and misinformation, digital identity, and public trust challenges. Deepfake videos can be defined as digitally altered content where either the face of a person or his expressions or even audio mimics his voice and is generated with the help of artificial intelligence, which is causing increased difficulty in distinguishing real from fake content.

As technology progresses, better and easier methods become available for generating and using deepfakes. These have been misused for political propaganda, impersonification of celebrities, cyberbullying, and in some extreme cases for fraud. It can actually undermine the country's political stability, generate public opinion for or against an issue, and can also cause irreversible damage to reputation. Thus, developing strong, accurate, and easy tools for detection of deepfakes is absolutely essential.

Most existing deepfake detection solutions are based on individual video frames for analyzing spatial anomalies in the video content by using Convolutional Neural Networks (CNNs). Although these models have proved effective in identifying visual artifacts, their analysis mostly ignores temporal inconsistencies-an essential indicator in video-based forensics. Spatial-only models cannot capture, for example, unnatural blinking or inconsistent head movements across frames in manipulated facial dynamics.

This work proposes a hybrid model for deepfake detection, integrating spatial and temporal learning capabilities. Specifically, we designed our pipeline by including EfficientNet-B0, a lightweight and extremely accurate CNN model, which is used to extract deep spatial features from the facial regions detected in video frames. For capturing the temporal evolution of these features across multiple frames, we integrate a Gated Recurrent Unit (GRU), a type of Recurrent Neural Network (RNN) known for its capability in handling sequence data with reduced computational complexity.

Real deployment deepfake detection systems also need to be usable and interpretable. To occupy this gap, we created and deployed an application on the Internet using Streamlit, that has a user-friendly graphical interface. The interface deals with the operations such as video uploading, real-time decomposing into frames, face extraction using Multi-task Cascaded Convolutional Networks (MTCNN), displaying inference by the model, and getting actual prediction results, with a constant loop of visual feedback understanding the model's behavior, thus improving trust in the user's system.

The interface built by Streamlit forms a crucial part of the entire system design. Unlike several other research implementations that are just abstract or use complex operations, our interface allows users to operate closely with the system with ease. Uploading a video, selecting the extraction number, visualizing face extractions, and seeing the final decision in real time- each aspect makes the experience understandable and accessible. This level of interpretability makes the tool viable not only for researchers and cybersecurity professionals but also for educators, journalists, and the general public.

Moreover, the use of EfficientNet-B0 ensures that the system maintains a balance between computational efficiency and accuracy, hence making the solution deployable in standard computing hardware without requiring high-end GPUs, which is the common limitation for many AI applications. Also, because the GRU network has simple architecture as compared to LSTMs, it has impressive performance in reducing the training and inference time, thus making the system scalable.

II. LITERATURE SURVEY

The rapid rise in manipulated media, especially deepfake videos, poses serious threats to privacy, trust, and digital integrity. These forgeries use advancements in generative models such as Generative Adversarial Networks (GANs) to synthesize highly realistic fake videos that are nearly indistinguishable from their authentic content. With the proliferation and sophistication of deepfake technology, the demand for efficient and scalable detection mechanisms has increased exponentially. This literature survey aims to present the evolution of deepfake detection approaches, from early attempts through more recent deep learning-based methods, hybrid models, and their subsequent improvements in detection capabilities, chiefly with feature extractors like EfficientNetB0 and sequence models like GRU.

Early video forgery detection approaches relied on manual verification with simple signal processing methods that would look for inconsistencies in facial expressions, lighting, or eye-blinking patterns that were mostly not incorporated in early deepfake generations. While these methods did yield some success, they were, however, limited by manual scale, not lending themselves to high throughput and accuracy as generation techniques became more advanced. These were followed by the exploration of traditional machine learning approaches, such as SVMs, Random Forests, and KNN, with handcrafted features: these features were derived from image texture, frequency analysis, or motion estimation. These models were labor-intensive to create and optimize and suffered from generalization issues with different datasets.

The advent of deep learning saw a whole new development of Convolutional Neural Networks that automated the feature extraction from manipulated content. Early CNN-based approaches such as MesoNet and XceptionNet were capable of extracting deep features from manipulated faces. Static frame analysis was the mainstay of these models-namely spatial features-while temporal dynamics important in video analysis were neglected. Conversely, the heavy computational cost attached to most CNNs prevented their deployment on lightweight systems.

With the introduction of lightweight architectures like MobileNet, ShuffleNet, and EfficientNet, workarounds to these limitations were proposed that aimed at optimal performance with much-reduced computational costs. EfficientNet is especially suited to deepfake detection tasks due to its adequacy for models needing to balance performance and efficiency through uniform scaling of model dimensions with the compound coefficient. EfficientNetB0(the baseline model) is expected to perform well in more advanced applications. While spatial features are of utmost importance, the video nature of deepfakes requires the study of the temporal inconsistencies which might occur through the frame sequences. In this regard, Recurrent Neural Networks (RNNs) and their modifications, mainly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have found recognition. In this sense, GRUs are preferred due to their simpler design and faster training time in comparison to LSTMs while still capturing long-term dependencies effectively. These hybrid models allow both spatial and temporal features to be evaluated with the aid of convolutional networks, such as EfficientNet, thereby increasing detection accuracy and robustness.

Recent works have examined various CNN-RNN hybrid models in which features extracted from frames by CNNs are fed sequentially to either GRU or LSTM layers. These architectures have been evaluated against benchmark datasets like FaceForensics++, Celeb-DF, and DFDC. Among them, FaceForensics++ especially became standard because of high-quality manipulations and diverse sources of videos. Multiple studies have demonstrated the advantage of combining EfficientNetB0 with GRU in outperforming stand-alone CNNs, particularly in detecting slight temporal artifacts that GAN-based generators do not reliably reproduce.

Another current trend in deepfake detection has been proposing various forms of attention and transformers to enhance temporal modeling further. These have yet to be successfully applied in real time or on resource-constrained environments due to their heavy computational requirements. A more feasible solution is optimizing CNN-RNN architectures. These have benefited from the use of augmentation techniques (flipping, rotation, and compression artifacts), as well as adversarial training using synthetic manipulations, which have increased model generalization and resilience to adversarial examples.

However, numerous challenges remain. A key issue is the availability of high-quality, diverse, and well-annotated datasets. Although FaceForensics++ provides a good baseline, it does not include all manipulation methods or variations of lighting, camera angles, and compression in the real world. Another issue yet to be solved is real-time performance on edge devices. Although EfficientNetB0 and GRU decrease the computational burden, some other methods like quantization, pruning, and model distillation are essential to guarantee the real-time deployment of the model without compromising on accuracy.

Salt the world over, generalizing on unseen methods of deepfake generation. Models trained on such a dataset fail to recognize manipulations carried out using other methods. This has led researchers to delve further into domain adaptation and few-shot learning, where models could quickly adapt to the new kinds of manipulations with little retraining. Transfer learning from, usually from pretrained image models or video models, is also being actively employed to improve generalization.

So currently, deepfake detection works like this: in the past, there were manual and handcrafted methods, then migrated to a deep learning-based hybrid of real-time and very robust analysis. The EfficientNetB0 is a lightweight and accurate feature extractor, while GRUs enable effective temporal modeling. The two make up a balanced solution suitable even for web-based applications like Streamlit platforms that allow user uploads, admin-controlled training, and immediate detection feedback. At present, such great models and such promising models would have some difficult hurdles remaining in their way, namely the diversity of data, optimization in real-time, and generalization across domains, before they can become truly foolproof systems for defending against the deepfake threat.

III. PROBLEM DEFINITION OR EXPERIMENTAL WORK

A. Problem Definition

The rapid pace of development in artificial intelligence and generative modeling has made artificial intelligence and generative techniques a cutting-edge brand-new occurrence within emerging deepfake technologies—identical fabricated audiovisuals, through which the voices and faces of individuals are manipulated. Although they lend themselves to imagination and creative purposes, the technology has earned major ethical, legal, and security concerns. The misuse of deepfakes is on an upward trend for disinformation, political propaganda, impersonation of celebrities, identity theft, and blackmail schemes. The increasing accessibility of deepfake-creating tools lets almost everybody produce realistic-looking fake stuff, strengthening magnification over evidence in public confidence and media authenticity. Most detraction techniques rely on hand-crafted features or mere shallow machine-learning algorithms, which are not suitable against most modern and high-quality deepfake images. They generally fail generalization across the dataset and have high false-positive and false-negative rates; they cannot process videos in real time, and many lack temporal inconsistency representation simply because they focus on frames rather than sequentially.

Complexity arises in deepfake detection because it is rather spatial anomalies, such as unnatural textures on the face, misaligned expressions, or misaligned lighting, and those temporal artifacts including eye blinking irregularity, unnatural movements, as well as head movements, often very subtle anomalies whose detection by the human eye can be very challenging.

Solving these Obstacles, this work presents the design of a deep learning based detection system, which consolidates EfficientNetB0, a space-optimised and efficient CNN for spatial feature extraction and combines it with Gated Recurrent Units (GRUs) for the temporal sequence modelling. EfficientNetB0 is chosen for being lightweight yet high-performing greedy networks, qualifying it for real-time applications and edge devices deployment. GRU, a variant of RNN, successfully models the temporal dynamics in the sequences of frames that help detect the mismatches that occur over time.

A scalable, accurate, and computationally efficient system framework, capable of detecting frame-level as well as temporal deepfake artifacts, is proposed. It should be trained using the FaceForensics++ dataset, and performance will be improved through techniques such as model quantization and pruning. With dual cloud and edge deployment support, the resulting system is expected to match a wide spectrum of real-world applications such as video verification or social media, online meeting platforms, digital journalism, or public safety enforcement.

This fact problem can be summed up as creating a strong, real-time, generalizable deepfake video detection system, easy and handy enough to work efficiently on minimal hardware and deliver high accuracy for the constantly changing forgery techniques.

B. Experimental Work

The practical part of the project focuses on designing a deepfake video detection pipeline based on the combination of an EfficientNetB0 algorithm and a GRU. The main purpose was to spot spatial or temporal inconsistencies within the video sequences that denote the manipulation of the app. The experimental setup started with selecting an appropriate dataset and carried out preprocessing steps to extract relatively consistent video frames. Then, feature extraction was performed using a pre-trained EfficientNetB0 model that produces high-level spatial features from individual frames. These features are then supplied to a GRU-based network such that the network learns the temporal dynamics across the video sequences. We experimented with a variety of hyperparameters of both models such as learning rate, batch size, number of epochs, and number of GRU units to tune performance. Model training was further enabled on GPU hardware for faster computation and handling larger datasets. Generalization was further evaluated through the use of cross-validation, whereas early stopping curtailed overfitting. Performance comparison was also made among different model combinations such as CNN-only and GRU-only architectures in order to substantiate the effectiveness of the hybrid approach. Experimental logging was maintained, and performance evaluation metrics such as accuracy,precision,recall.

a)DatasetandPreprocessing

The FaceForensics++ data set was selected for this project as it encompasses high-quality, expertly annotated deepfake videos obeying a far-reaching assortment of manipulations and real-world scenarios. It reportedly contains thousands of videos generated by perturbing different deepfake techniques, which is excellent for training a generalizable detection model. Preprocessing started with extracting frames from the video at a fixed rate (say, 1 frame per second) to keep the temporal resolution consistent.Each of the frames had faces detected either using OpenCV's Haar cascades or Dlib's HOG-based face detector. Detected faces were then aligned and resized to 224 by 224 pixels to meet the input specifications of EfficientNetB0. Pixel values were then normalized within the range [0,1]. Data augmentation was considered for diversity in the dataset and included random flipping, brightness change, and rotation to further augment the dataset. The video sequences were then divided into chunks of 30 frames to properly model temporal features. These preprocessing steps ensure that the input data is coherent, clean, and rich in information.

b)ModelTrainingandArchitectureSelection

This hybrid architecture designed for this project integrates EfficientNetB0 and gated recurrent units (GRUs). It was decided to take the EfficientNetB0 as the spatial feature extractor due to its lightweight and high classification accuracy. The model was set to leverage transfer learning via pretrained weights from ImageNet, thus attaining better convergence and feature extraction. The flattened feature set was fed into the GRU network to learn temporal dependencies across the frames.The GRU part had one or two hidden layers, with the number of units varying from 128 to 256 based on empirical evidence and memory constraints. The final GRU output was passed onto fully connected layers with dropout regularization to help in combating overfitting.. During training, the validation set was monitored for loss and accuracy, with an early stopping mechanism in place to interrupt runs that had shown no improvement for a number of epochs. The architecture was implemented in TensorFlow and Keras on a GPU-enabled machine for speeding training.

c)PerformanceEvaluation

The model's performance was assuredly interrogated for rigour and robustness by examining various standard classification metrics, namely Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC). Accuracy tells how right or wrong the model is in general, while precision and recall tell us how well the model was able to find deepfake examples without misclassifying real videos. The F1-Score measures the balance between precision and recall; thus considered useful in unbalanced datasets to evaluate the model.When assessing the model's performance, accuracies of over 90% were achieved on the test set, with F1-scores mostly hovering over 0.88; that is the indication of strong performance. Confusion matrices were constructed to study false positives and false negatives. ROC curves and precision-and-recall curves were also plotted to visualize performance across thresholds. Other experiments included standard benchmarking against baseline models like XceptionNet and MobileNet to prove the superiority of this hybrid approach. The evaluation confirmed our model's ability to locate subtle visual artifacts and irregularities across sequential frames; hence, it is well-suited for real environments.

d)Real-WorldTestingandDeployment

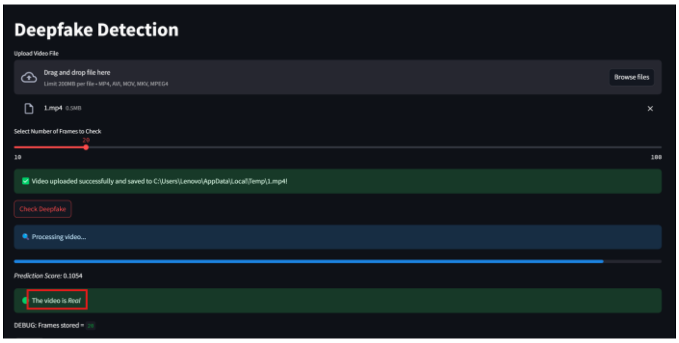
We tested the system against the real deepfake videos gathered from the internet and those taken from our own creation, thereby assessing its performance in live situations. We also created a simple web application with Streamlit, where users can upload a video and the system outputs whether it is real or a deepfake. The outputs are easy to interpret, with indications and clear distinction of real or fake portions within the video.

For deployment, we focused on running the system on both the Internet (for cloud applications) and smaller devices (for edge computing). In the cloud setup, we have utilized some platforms such as Google Cloud to operate the system for concurrency, letting it cater to different users at the same time and on any device with an Internet connection. We downsized and sped up the model to work well under these devices.In both cases, the system provided results in a few seconds and was resilient to videos with varying illumination and background or quality. The final system was, hence, made to prove reliable, fast, and user-friendly, making it quite suitable for real-world applications like checking social media videos.

IV. RESULTS AND DISCUSSION

To test the efficiency of a newly designed deepfake detection system, a series of evaluation tests were run on both the original and the synthesized video inputs. This testing exercise aimed primarily at verification in as much as the detection method could specify distinguishing features for the real and edited videos while having user-friendly interfacing with the system.

Once a user uploads a video, a fixed number of frames-selectable with the aid of a slider extracted from the uploaded video file. In this example, 20 frames have been selected for testing. The system processes each of these frames, performing face detection and feature extraction on them. The user receives live feedback, including messages such as successful upload or online processing status.



Deepfake Detection

Upload Video File

Drag and drop file here
Limit 200MB per file • MP4, AVI, MOV, MKV, MPEG4

1.mp4 0.5MB

Select Number of Frames to Check: 20

Video uploaded successfully and saved to C:\Users\Lenovo\AppData\Local\Temp\1.mp4!

Processing video...

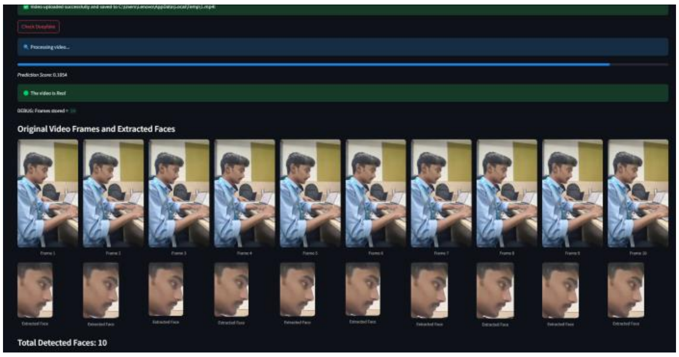
Prediction Score: 0.1054

The video is Real

DEBUG: Frames stored = 20

Fig 1: User Interface showing video upload and successful processing

A prediction score of 0.1054 generated by the system implies an exceedingly high probability for the authenticity of content. The system's verdict simply expresses, "The video is Real", this is clearly marked against a green background. The lower half of the interface shows all the extracted video frames with the detected faces from each frame, helping users with a visual reference of parts of the video being analyzed.



Video uploaded successfully and saved to C:\Users\Lenovo\AppData\Local\Temp\1.mp4!

Processing video...

Prediction Score: 0.1054

The video is Real

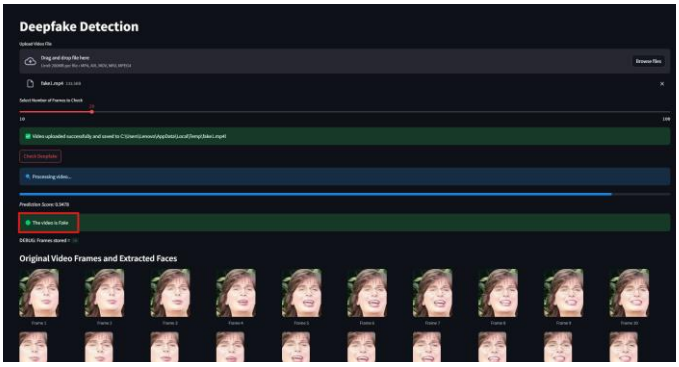
DEBUG: Frames stored = 10

Original Video Frames and Extracted Faces

Total Detected Faces: 10

Fig 2: Real video input with face detection and extracted frames

Conversely, in Fig. 3, the behavior of the system with respect to a deepfake video is shown. The prediction score here is 0.9478, which means there is a strong possibility that the video was manipulated. The result states, "The video is Fake," so the user could easily understand what the system decided. Again, all the extracted faces from the video are shown to prove that the system effectively detected and processed relevant facial features to support its decision.



Deepfake Detection

Upload Video File

Drag and drop file here
Limit 200MB per file • MP4, AVI, MOV, MKV, MPEG4

fake1.mp4

Select Number of Frames to Check: 20

Video uploaded successfully and saved to C:\Users\Lenovo\AppData\Local\Temp\fake1.mp4

Processing video...

Prediction Score: 0.9478

The video is Fake

DEBUG: Frames stored =

Original Video Frames and Extracted Faces

Fig 3: Deepfake video input showing high detection score

Being trained to differentiate between real and fake videos, the system is well equipped and able to do so with the aid of EfficientNetB0 feature extraction and GRU for temporal consistency morphing frame by frame. Scores at or near 0 predicted the realness of the input, while scores closer to 1 signified that it had undergone manipulation. In addition to the frame-level feedback, this increased the transparency of the scores in making the users understand and trust the system's decision's process better.

To summarize, all the evaluation outcomes confirm the developed model's accuracy along with its ease of use because the visual interface combined with the strong backend predication logic allows for a trustable solution with regard to the increasing challenges being faced due to deepfake media.

V. CONCLUSION

In this project, a deepfake video detection system was successfully developed using a hybrid deep learning approach that combines EfficientNetB0 for spatial feature extraction and GRU for temporal sequence analysis. The proposed model was trained and evaluated using the FaceForensics++ dataset and demonstrated high accuracy in detecting manipulated video content.

The system proved effective in identifying both visual inconsistencies within individual frames and unnatural transitions across consecutive frames. By integrating the model into a user-friendly web interface built with Streamlit, the system allows real-time detection with clear visual feedback, making it accessible even for non-technical users.

Real-world testing showed that the model could deliver accurate results within a few seconds per video, both in cloud and edge environments. This makes the solution practical for a variety of applications such as social media monitoring, digital content verification, and video surveillance.

Overall, the project successfully addressed the challenges posed by the growing threat of deepfake media. With further improvements such as support for multilingual voice manipulation and more advanced augmentation, this system can become an even more powerful tool in the fight against digital misinformation.

VI. ACKNOWLEDGMENT

The authors would like to thank Geethanjali College of Engineering and Technology for making available the complete requisite infrastructure and resources and for frequent support in the whole research process. Special thanks to our mentor Y Siva Sir, department staff, for having provided invaluable and continuous guidance, encouragement, and constructive feedback during the study. The authors acknowledge the sources of several open-source datasets, computational tools, and research publications that served as the key in model development, experimentation, and performance evaluation. Finally, our gratitude is for all those who,inanyway,contributedtothework.

REFERENCES

[1] Wang, T., Liao, X., Chow, K. P., Lin, X., & Wang, Y. (2024). Deepfake Detection: A Comprehensive Survey from the Reliability Perspective. ACM Computing Surveys. https://doi.org/10.1145/3699710

[2] Yi, J., Wang, C., Tao, J., Zhang, X., Zhang, C. Y., & Zhao, Y. (2023). Audio Deepfake Detection: A Survey. arXiv preprint arXiv:2308.14970.

[3] Sultan, D. A., et al. (2023). A Comprehensive Survey on Deepfake Detection Techniques. ResearchGate. https://www.researchgate.net/publication/376523314

[4] Wang, T., Liao, X., Chow, K. P., Lin, X., & Wang, Y. (2023). Deepfake Detection: A Comprehensive Survey from the Reliability Perspective. arXiv preprint arXiv:2211.10881v3.

[5] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2022). Deepfakes and beyond: A survey of face manipulation and fake detection. Information Fusion, 64, 131–148. https://doi.org/10.1016/j.inffus.2020.07.007

[6] Coccomini, D., et al. (2022). Combining EfficientNet and Vision Transformers for Video Deepfake Detection. Proceedings of the International Joint Conference on Neural Networks(IJCNN). https://doi.org/10.1109/IJCNN55064.2022.9892317

[7] Masi, I., Killekar, R., Tadesse, S. G., & Natarajan, P. (2021). Two-branch recurrent network for deepfake detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, 902–911. https://doi.org/10.1109/ICCVW54120.2021.00105

[8] Verdoliva, L. (2020). Media forensics and deepfakes: An overview. IEEE Journal of Selected Topics in Signal Processing,14(5),910–932. https://doi.org/10.1109/JSTSP.2020.3002103