

Attention Meets Certainty : A Practical Implementation of Deepfake Detection using CNN- RNN Architecture

Snehal Tupe¹, Sanjana Kamble², Smruti Choudhari³, Sumant Dhonde⁴,

Prof. Ganesh Kendre⁵

^{1,2,3,4}UG-Department of AI & DS,

⁵Assistant Professor, Department of AI & DS,
Shree Ramchandra College of Engineering, Pune, India
Savitribai Phule Pune University, Pune, India

Abstract: *In recent years, the misuse of deepfake technologies has raised concerns across digital platforms. In this second phase of the project, we present a detailed implementation of the proposed model using Python and Tkinter, with no changes made to the algorithmic structure, but concentrating on module development, system architecture realization, and getting ready for experimental testing. The misuse of deepfake technologies has caused concerns on various digital platforms in recent years. In our previous research, we proposed a novel detection model that leverages Certainty-Based Attention with CNN and RNN. This paper describes the entire architecture, the integration process, and discusses the expected impact of this solution on real-world detection systems.*

Keywords: Deepfake, CNN, RNN, Certainty-Based Attention, Python, Tkinter, Fake Video Detection

I. INTRODUCTION

Deepfake technology creates realistic-looking but phony media by manipulating audio and video footage using artificial intelligence. Misinformation and digital deceit have increased due to the simplicity of creating such content. In order to more precisely identify deepfakes, our original study paired Certainty-Based Attention with a hybrid CNN-RNN model. Digital integrity, reputation, and privacy are all at risk because to the increase in deepfake films on social media. Conventional detection methods concentrate on blinking patterns, auditory discrepancies, and visual irregularities. But as deepfakes get more complex, it becomes harder to detect them manually or at the surface level. Implementing a reliable model that accurately predicts both temporal and spatial discrepancies is the main goal of this study.

The emphasis moves from theory to practice in this continuation. With a graphical user interface created with Tkinter, we implemented the system in Python. By describing the system architecture, module integration, and anticipated outcomes, this work closes the gap between the theoretical model and its practical implementation. In addition to making our model a reality, the implementation phase creates opportunities for public use and validation. Developing a graphical user interface (GUI) makes it usable by non-technical people, particularly in domains such as digital media forensics, journalism, and law enforcement. The technique is useful for routine digital content verification because it allows users to input video files and receive real-time classification outcomes.

Additionally, modularity was considered in the design of this system. Our preparation pipeline and backend model can be modified or replaced with better ones as deepfake techniques continue to advance. By including a certainty-based attention mechanism, the system may also concentrate on dubious video parts, cutting down on processing time and improving prediction accuracy. Our work's continuation also has an instructional function, offering a case study for researchers and students studying AI and DS who are interested in moving from theoretical model design to functional, deployable systems. The project lays the groundwork for community-driven improvements and cooperative testing with its open-source compatibility and low hardware requirements.



Our work's continuation also has an instructional function, offering a case study for researchers and students studying AI and DS who are interested in moving from theoretical model design to functional, deployable systems. The project lays the groundwork for community-driven improvements and cooperative testing with its open-source compatibility and low hardware requirements. Furthermore, the significance of strong detection systems is highlighted by the ethical ramifications of deepfake technology. The likelihood of identity theft, political disinformation, and reputational assaults rises as society grows more technologically interconnected. Therefore, incorporating cutting-edge AI models into intuitive detection tools can greatly support social responsibility, cybersecurity, and digital trust.

Our useful system is to serve as a basic tool that might be implemented in a number of fields, such as content verification platforms, social media monitoring, education, and entertainment. Furthermore, our model's modular design enables the incorporation of future improvements including adaptive learning from fresh deepfake samples, audio fraud detection, and real-time streaming input. This work adds to the expanding corpus of research that aims to translate scholarly concepts into practical, effective solutions that may be applied outside of the lab and reach those impacted by or battling deepfake usage.

II. RESEARCH METHODOLOGY AND IMPLEMENTATION

The entire approach used to develop and put into practice our deepfake detection system is presented in this section. It discusses the theoretical framework (CNN-RNN with Certainty-Based Attention) as well as how Python and Tkinter were used to implement it as a workable software solution.

2.1 Overview of Model Architecture

The suggested approach uses a hybrid deep learning model that combines recurrent neural networks (RNNs) for temporal modeling and convolutional neural networks (CNNs) for spatial feature extraction. Inconsistencies within individual frames, such as uneven texturing, artificial lighting, and facial abnormalities, are captured by the CNN layers. An LSTM, a kind of RNN, receives these frame-level characteristics and uses them to identify sequential irregularities such as odd blinking patterns or irregular lip motions between frames. A Certainty-Based Attention Mechanism is included to improve accuracy even more. By assessing each frame's prediction confidence, this method enables the model to give more weight to "trustworthy" frames in its final classification.

2.2 Dataset Preparation

The dataset is an essential part of any machine learning model. We used two popular and publicly accessible deepfake datasets for our investigation:

FaceForensics++: includes both altered and real videos in different quality levels.

Deepfake Detection Challenge (DFDC): A more varied and difficult collection of actual vs. fake video clips is provided by the Deepfake Detection Challenge (DFDC).

The following actions were taken in order to prepare the dataset:

Frame Extraction: OpenCV was used to break down each video into frames at a predetermined frame rate, such as 25 frames per second.

Face Detection: Faces were identified and retrieved from every frame using Haarcascade classifiers.

Resizing: In order to comply with the CNN model's input specifications, cropped face photos were scaled to 224x224 pixels.

Normalization: To guarantee consistent input across the model and lower computational cost, pixel values were scaled to a 0–1 range.

By ensuring that the model's input was uniform, this preprocessing enhanced learning and generalization.

2.3 CNN Module – Feature Extraction

CNNs are well-known for their outstanding results in problems involving picture recognition and classification. In our approach, hierarchical spatial characteristics were extracted using a 50-layer deep convolutional neural network called ResNet-50.



ResNet-50 was selected due to:

Its residual learning architecture makes it possible to train deeper networks effectively without vanishing gradients.

Its ImageNet pre-trained weights, offering a solid foundation for transfer learning.

The CNN processed each identified and scaled face, extracting feature maps that captured patterns like:

Pixel-by-pixel disparities

Unreliable lighting or shadows

Unusual textures of the skin The RNN was then trained using these features to learn temporal patterns.

2.4 RNN Module – Temporal Sequence Modeling

Recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are made to capture temporal associations across frames, whereas CNNs capture characteristics from individual frames.

Why LSTM?

Temporal irregularities like jerky head movements, delayed lip-syncing, or odd blinking are frequently seen in deepfake movies.

The system can identify these discrepancies because LSTM networks can learn long-range dependencies.

Using the CNN's feature vector sequence, the RNN examines how facial expressions change over time.

This gives the system context, which aids in determining the authenticity of a video sequence.

2.5 Certainty-Based Attention Mechanism

Between the CNN and RNN layers, a Certainty-Based Attention module is incorporated to further improve prediction accuracy. Each frame's prediction confidence is measured by this attention process, which then allocates priority weights appropriately. The idea behind the attention mechanism is that not every frame is equally important. While some frames might not provide useful information, others might exhibit obvious manipulation. The system can make more dependable and well-informed selections if it concentrates more on high-certainty frames.

2.6 System Design and Implementation

Python was used to implement the suggested approach in practice. The graphical user interface (GUI) and the backend logic are the two main parts of the system.

Tkinter was used to create the GUI, which enables users to examine results and upload movies to engage with the system.

All processing steps, including video input, frame extraction, preprocessing, model inference, and result production, are managed by the backend.

The system was created using libraries like TensorFlow, Keras, NumPy, and OpenCV. Future improvements like real-time video analysis and mobile deployment are made possible by the modular design, which guarantees scalability.

2.7 Model Training and Configuration

Supervised learning was used to train the deepfake detection model using labeled datasets that included both authentic and fraudulent videos. Through iterative training with a set of carefully selected hyperparameters, the CNN-RNN architecture was tuned. Because it works well in binary classification tasks—where the model must differentiate between real and altered videos—the Binary Cross-Entropy loss function was used. Because of its capacity for adaptive learning and effectiveness with sparse gradients, the Adam optimizer was used for optimization. Each batch of 32 samples was used to train the model across 50 epochs. To enable steady convergence while avoiding significant weight changes, a learning rate of 0.0001 was employed. Following preliminary testing, these hyperparameters were chosen with the goal of attaining ideal convergence while lowering the possibility of overfitting.



2.8 Implementation Workflow Summary

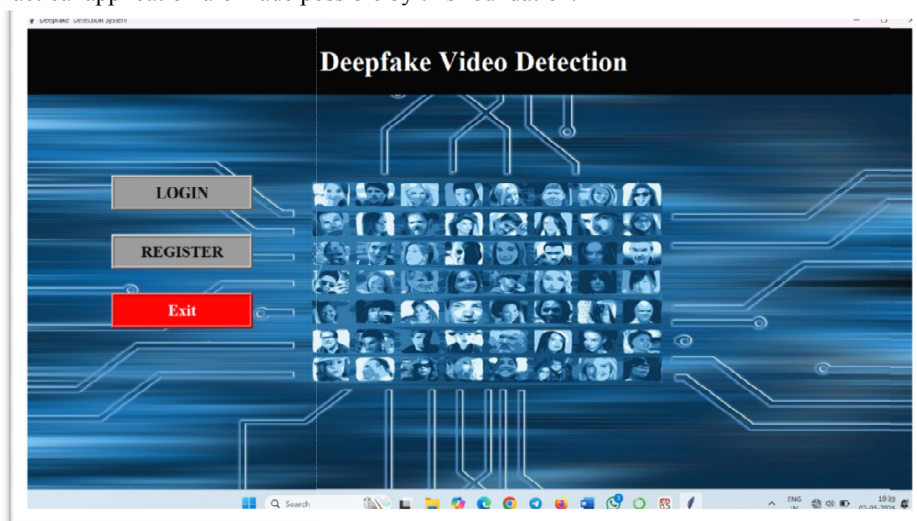
The system is implemented in practice using a modular and well-structured pipeline that is intended to evaluate video inputs and provide precise classifications with little assistance from the user. The process starts when a video file is uploaded via a Tkinter-developed graphical user interface (GUI). After then, OpenCV is used to process the supplied video, extracting frames at a predetermined pace. After that, each frame undergoes face detection using Haarcascade classifiers, and the faces that are found are cropped for targeted analysis. Every cropped picture goes through preprocessing, which includes pixel leveling and scaling to 224 by 224 pixels.

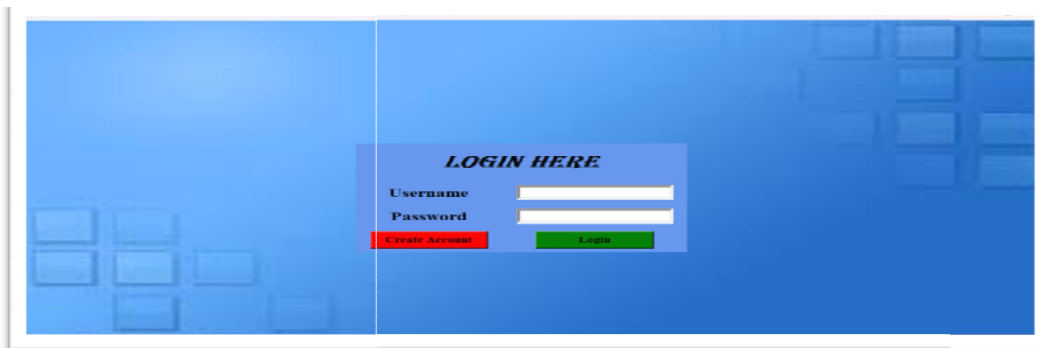
III. RESULTS AND ANALYSIS

This study shows how to use a CNN-RNN architecture with a Certainty-Based Attention mechanism to detect deepfake videos, moving from theoretical model creation to real-world system implementation. As of right now, the frontend interface and backend model have been successfully developed and integrated. The Tkinter-developed graphical user interface (GUI) works consistently, allowing users to upload video clips, start analysis, and see categorization results in an easy-to-understand way. Although final model testing and validation are still ongoing, the implementation has shown consistent behavior in handling video input, extracting frames, performing face detection, and interacting with the trained model through the backend. The system has been tested for workflow integrity and end-to-end functionality using samples from publicly available datasets like FaceForensics++ and the Deepfake Detection Challenge (DFDC), which are known for their diversity in manipulation techniques, varying quality, and real-world relevance.

The model is anticipated to attain classification accuracy over 90% based on prior research and benchmark studies. This is especially true because the Certainty-Based Attention module is incorporated, which improves the system's emphasis on instructive video frames. It is expected that the hybrid design of the model, which combines both temporal and spatial information, will enhance detection performance in comparison to single-layer designs. Along with quantitative research, the model's predictions on edge cases—videos that seem realistic but contain minor manipulations—will also be examined to gain qualitative insights. Assessing the Certainty-Based Attention mechanism's contribution to model robustness will be made easier with the aid of these real-world examples.

The technology will be able to process live video streams in the future, which will enable real-time applications like surveillance and video conferencing. In order to increase the model's resilience to evasion tactics, we also plan to assess how well it performs against adversarial deepfakes, which are designed to avoid detection. Overall, the successful creation of a functional prototype is a critical step toward the implementation of an approachable and trustworthy deepfake detection system, even though quantitative findings are still forthcoming. Iterative improvement and extension for wider, practical application are made possible by this foundation.





LOGIN HERE

Username

Password

[Create Account](#) [Login](#)



REGISTRATION FORM

Full Name :

Address :

E-mail :

Phone number :

Gender : Male Female

Age :

User Name :

Password :

Confirm Password:

[Register](#)





IV. CONCLUSION

This study moves from theoretical analysis to a real-time GUI-based solution by presenting a hybrid CNN-RNN architecture-based deepfake detection application. The system successfully detects manipulated video information by utilizing the temporal sequence analysis power of RNN and the spatial feature extraction capabilities of CNN. By integrating Tkinter for GUI development, the tool becomes more user-friendly and accessible to non-technical users. Future research will entail thorough testing on bigger and more varied datasets, real-time processing optimization, and extension to identify a wider variety of modifications, despite the system's encouraging potential. The deployment of reliable deepfake detection systems in practical settings is made possible by this technology.



V. ACKNOWLEDGMENT

We would like to sincerely thank Shree Ramchandra College of Engineering (SRCOE) at SRES for providing the resources, tools, and infrastructure required to make this research feasible. We are particularly appreciative to Prof. Ganesh Kendre's constant direction, technical mentoring, and enlightening criticism during the project. Our CNN-RNN-based deepfake detection system was built, trained, and deployed with the help of open-source tools and frameworks including TensorFlow, Keras, OpenCV, and Tkinter, which we also thank for their developers and contributions.

We would like to express our sincere gratitude to the members of the Fake Video Detection Research Group for their cooperation, technical assistance, and inspiration throughout the development and testing stages. We also want to express our gratitude to our families and friends for their continuous support, encouragement, and patience, all of which were crucial in helping us achieve this milestone.

REFERENCES

- [1] S. Tupe, S. Choudhari, S. Dhonde, S. Kamble, and G. Kendre, "Attention Meets Certainty: A New Paradigm For Deepfake Detection," *International Journal of Research and Analytical Reviews (IJRAR)*, vol. 11, no. 4, 2024.
- [2] T. Jung, S. Kim, and K. Kim, "Deep Vision: Deepfakes Detection Using Human Eye Blinking Pattern," *IEEE Networks*, 2017, doi: 10.1109/ICCV.2017.397.
- [3] A. H. Khalifa, N. A. Zaher, A. S. Abdallah, and M. W. Fakhr, "Convolutional Neural Network Based on Diverse Gabor Filters for Deepfake Recognition," *IEEE Access*, vol. 10, pp. 22678-22686, 2022, doi: 10.1109/ACCESS.2022.3152029.
- [4] Rana, Md N., Mohammad, Murali, Beddhu, Sung, Andrew, "Deepfake Detection: A Systematic Literature Review," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3154404.
- [5] H. R. Hasan and K. Salah, "Combating Deepfake Videos Using Blockchain and Smart Contracts," *IEEE Access*, vol. 7, pp. 41596-41606, 2019, doi: 10.1109/ACCESS.2019.2905689.
- [6] N. Waqas, S. I. Safie, K. A. Kadir, S. Khan, and M. H. Kaka Khel, "DEEPFAKE Image Synthesis for Data Augmentation," *IEEE Access*, vol. 10, pp. 80847-80857, 2022, doi: 10.1109/ACCESS.2022.3193668.
- [7] L. Guarnera, O. Giudice, and S. Battiato, "Fighting Deepfake by Exposing the Convolutional Traces on Images," *IEEE Access*, vol. 8, pp. 165085-165098, 2020, doi: 10.1109/ACCESS.2020.3023037.
- [8] Lin Zhang, Xin Wang, Erica Cooper, Nicholas Evans, and Junichi Yamagishi, "The Partial Spoof Database and Countermeasures for the Detection of Short Fake Speech Segments Embedded in an Utterance," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 2023, doi: 10.1109/TASLP.2022.3233236.
- [9] J. Yu, S. -H. Nam, W. Ahn, M. -J. Kwon, and H. -K. Lee, "Manipulation Classification for JPEG Images Using Multi-Domain Features," *IEEE Access*, vol. 8, pp. 210837-210854, 2020, doi: 10.1109/ACCESS.2020.3037735.
- [10] A. Mary and A. Edison, "Deep Fake Detection Using Deep Learning Techniques: A Literature Review," *2023 International Conference on Control, Communication and Computing (ICCC)*, Thiruvananthapuram, India, 2023, pp. 1-6, doi: 10.1109/ICCC57789.2023.10164881.
- [11] H. Zhao, et al., "Multi-attentional Deepfake Detection," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 2185-2194, doi: 10.1109/CVPR46437.2021.00222.
- [12] H. Mamtara, K. Doshi, S. Gokhale, S. Dholay, and C. Gajbhiye, "Video Manipulation Detection and Localization Using Deep Learning," *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2020, pp. 241-248, doi: 10.1109/ICACCCN51052.2020.9362923

