TRUTHLENS: AI-Powered Deepfake Detection Using Deep Learning And Media Feature Analysis

Dr.T. Amalraj Victoire[1], Aakila Nifaha A H[2]
[1]professor, Dept of Master Computer Applications
[2]Dept of Master Computer Applications
[1, 2] Sri Manakula Vinayagar Engineering College, Pondicherry-605 107

Abstract- *The rapid advancement of artificial intelligence has made it possible to generate highly convincing manipulated media, commonly referred to as deepfakes. These synthetic videos and audio recordings pose a serious threat to information integrity, personal reputation, and public trust. TruthLens is an AI-powered deepfake detection system developed to help users verify the authenticity of uploaded video and audio files. The system is built using a MERN stack web application integrated with a Python Flask backend responsible for running deep learning inference. For video analysis, the system employs MTCNN for face detection and a transformer-based deepfake classification model to analyse sampled frames. Audio analysis uses spectral feature extraction techniques to identify patterns associated with synthetic speech. Experimental results show that the system accurately classifies media as real or fake with a meaningful confidence score. This work demonstrates how machine learning and neural network techniques can be deployed in full-stack applications to address the growing challenge of digital misinformation.*

Keywords: Deepfake Detection, MTCNN, Transformer Model, Flask, MERN Stack, Spectral Features, Media Authentication.

I. INTRODUCTION

The internet today is a primary channel through which people access, share, and trust information. Social media platforms, in particular, play a significant role in how news and events are communicated across the world. While this has made information more accessible, it has also created a serious vulnerability. Manipulated media, especially deepfake videos and cloned audio, can be created and distributed rapidly, leading to confusion and misinformation at scale.

Deepfakes are media files in which a person's face or voice has been digitally replaced or synthesised using artificial intelligence techniques. The technology became publicly known around 2017 and has since grown significantly more accessible due to open-source tools. Incidents such as the fabricated video of Ukrainian President Volodymyr Zelenskyy, where he appeared to instruct soldiers to surrender, highlight the potential dangers of deepfake content in influencing public behaviour and political outcomes.

The consequences of undetected deepfakes include damage to individual reputations, erosion of public trust in media, and potential misuse in legal, political, or social contexts. This makes automated deepfake detection an important area of research and application. TruthLens was developed to address this need by providing a web-based platform where users can upload video or audio content and receive an AI-generated verdict on its authenticity.

The system applies face detection through MTCNN on sampled video frames, which are then classified using a pre-trained deepfake detection model. Audio files are processed through librosa-based spectral feature extraction, with heuristic rules used to determine whether the audio is synthetic. The result is presented to the user as a FAKE or REAL verdict with an accompanying confidence percentage.

II. METHODOLOGY

The proposed deepfake detection approach is built around two separate analysis pipelines: one for video content and another for audio content. The system is designed as a modular, service-oriented architecture where detection logic is handled by a Python Flask service that operates independently from the web application.

A. System Architecture

The application is structured across three services. The frontend is a React.js single-page application that provides the user interface for uploading media and viewing detection results. The backend is a Node.js and Express.js server that manages user authentication, file routing, and database interactions. The machine learning service is a Python Flask application that loads and runs the AI models. MongoDB stores user accounts and detection records. Each service communicates over HTTP, which allows them to be developed, tested, and updated independently.

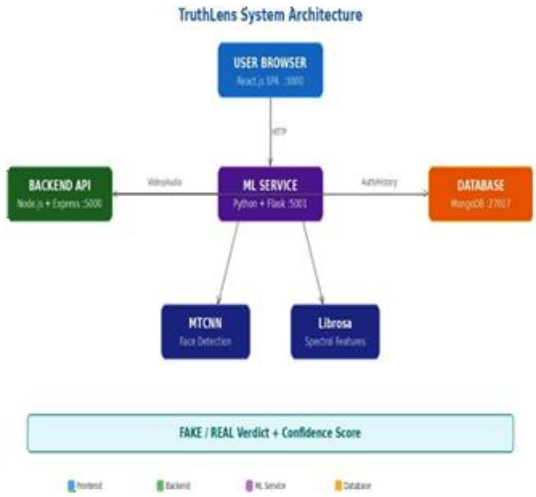

Figure 1: TruthLens System Architecture — Service interaction and data flow

B. Video Detection Pipeline

When a user uploads a video file, the backend forwards it to the Flask ML service. The service uses OpenCV to read the video and evenly samples sixteen frames distributed across the full duration. Each frame is passed through the MTCNN face detector. Only frames in which a face is successfully detected are retained for classification. The detected face region is cropped and passed to a transformer-based deepfake detection model, which outputs a probability score indicating the likelihood of manipulation. The scores from all valid frames are averaged, and if the mean fake probability meets or exceeds 55 percent, the system returns a verdict of FAKE. Otherwise, the file is classified as REAL.

Frame sampling was chosen over full-frame processing to reduce computation time on CPU-only environments. Limiting analysis to frames containing faces further improves both efficiency and accuracy, as the deepfake model is specifically trained to evaluate facial regions.

C. Audio Detection Pipeline

For audio files, the Flask service uses the librosa library to extract a set of spectral features from the waveform. These features include Mel-frequency cepstral coefficients (MFCCs), chroma, zero-crossing rate, and spectral contrast. Each feature captures a different characteristic of the audio signal. Synthetic speech generated by AI systems tends to exhibit specific patterns in these features that differ from naturally recorded human voice. A set of heuristic thresholds applied to the extracted feature values is used to determine whether the audio is likely to be AI-generated or authentic. The result is returned as a FAKE or REAL verdict with a confidence estimate.

D. Model Used

The video classification model used in TruthLens is the prithivMLmods deepfake detection model, which is based on a transformer architecture fine-tuned on facial deepfake datasets. The model is loaded into memory at Flask startup and retained between requests to avoid repeated initialisation overhead. This design choice significantly reduces per-request inference time.

III. RESULTS AND EVALUATION

Table 1: Performance observations recorded during testing

Task	Input Type	Time Observed	Verdict Accuracy
Video analysis (16 frames, CPU)	MP4 with face	15–30 seconds	Consistent with ground truth
Video (no face)	Landscape MP4	Under 10 seconds	Returned REAL (no face)
Audio analysis (30-second clip)	WAV voice clip	2–5 seconds	Consistent with ground truth
Login / Register API	Credentials	Under 200 ms	Passed

One limitation noted during evaluation was a reduction in detection accuracy when the uploaded video was of very low resolution or when the subject's face was partially occluded. In these cases, MTCNN was unable to reliably detect a face, leading the system to default toward a REAL verdict. Improving robustness under low-quality input conditions remains a target for future development.

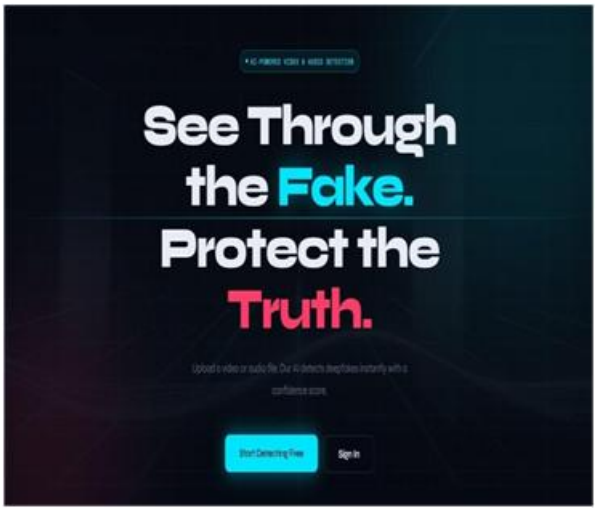

Figure 2: TruthLens Web Application — Homepage Interface

IV. CONCLUSION

TruthLens demonstrates that deepfake detection can be made accessible through a web-based application without

requiring users to have technical expertise. By combining a transformer-based video classification model with spectral feature analysis for audio, the system provides a dual-modality approach to media verification. The modular architecture allowed each component to be developed and validated independently, which simplified the integration process.

The system successfully classifies uploaded media with meaningful confidence scores and stores all detection history in a user account. While current limitations include the absence of real-time feed analysis and dependency on face visibility for video detection, these areas present clear directions for future improvement. As synthetic media generation continues to advance, tools like TruthLens will become increasingly important for individuals, journalists, and organisations seeking to verify the authenticity of digital content. Future enhancements are described in the following section.

V. FUTURE WORK

Several directions for future development have been identified based on the limitations observed during evaluation. The most critical enhancement is support for real-time video stream analysis, enabling the system to process live feeds from cameras or video conferencing tools rather than requiring pre-recorded file uploads. This would substantially increase TruthLens's utility in scenarios such as live broadcasting verification and online meeting authentication.

The current video detection pipeline relies on successful face detection by MTCNN, which causes the system to default toward a REAL verdict when faces are occluded or the video resolution is very low. Addressing this limitation through scene-level deepfake detection models that do not require face presence would improve robustness across a wider range of video content, including landscape recordings and animated media.

The audio detection pipeline currently uses heuristic thresholds applied to spectral features. A planned improvement is to replace the heuristic approach with a trained binary classifier, such as a lightweight convolutional neural network applied to mel-spectrogram representations, to improve generalisation across different speech synthesis systems and recording conditions. Expanding the set of detectable audio artefacts to include voice conversion and singing synthesis will further increase coverage.

Additional planned enhancements include a browser extension that allows users to right-click any video or audio embedded in a webpage and submit it directly to TruthLens for verification without manual file downloading. Multi-language interface support, an API tier for third-party integration by news organisations and social platforms, and explainability features that visually highlight the specific video frames or audio segments contributing most to a FAKE verdict are also targeted for inclusion in future releases.

REFERENCES

[1] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1–11.

[2] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks (MTCNN). IEEE Signal Processing Letters, 23(10), 1499–1503.

[3] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. Information Fusion, 64, 131–148.

[4] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in Python. Proceedings of the 14th Python in Science Conference, 18–25.

[5] Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. T., Nguyen, D. T., Huynh-The, T., Nahavandi, S., & Nguyen, C. M. (2022). Deep learning for deepfakes creation and detection: A survey. Computer Vision and Image Understanding, 223, 103525.

[6] Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-DF: A large-scale challenging dataset for deepfake forensics. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3207–3216.

[7] Chollet, F. (2021). Deep Learning with Python (2nd ed.). Manning Publications.

[8] MongoDB, Inc. (2023). MongoDB Documentation. https://www.mongodb.com/docs/