# Multi - Modal Deepfake Detection System

Kausthubh Peddibhotla[1], Sutrala Suvidyendra[2], A Anjani Prasad[3], Prof. K Radhika[4]
[1,2,3,4]*Department of Artificial Intelligence and Data Science, Chaitanya Bharathi Institute of Technology, Hyderabad, India*

***Abstract*—The AI Presence Detector is a robust, AI-powered web application developed to counter the increasing threat of synthetic and manipulated digital media. With the widespread use of deepfakes and AI-generated content, the authenticity of text, image, and video data has become a major concern. This project introduces a unified, multi-modal detection system that combines advanced machine learning models to analyze and classify content in real time. Users can upload various media types through a centralized interface, where videos and images are processed using CNNs and Hugging Face transformer models trained on datasets like Face Forensics++, and text content is evaluated using TF-IDF vectorization and an artificial neural network classifier. The system provides clear prediction results, manipulation scores, and interpretive feedback, all through a streamlined React.js frontend backed by a modular Flask server. Designed for accessibility, scalability, and performance, the platform supports temporary file management, concurrent processing, and a responsive user experience. It serves a wide range of users including educators, forensic professionals, journalists, and the general public by enabling them to verify content credibility quickly and reliably. The project also lays the groundwork for future enhancements such as live stream deepfake detection, audio content analysis, mobile deployment, and cloud integration. By blending state-of-the-art AI techniques with a user-friendly design, the AI Presence Detector fosters digital transparency, strengthens media integrity, and supports informed decision-making in an increasingly AI-driven world.**

***Index Terms*—Deepfake detection, Plagiarism detection, AI-generated text, CNN, Transformers, TF-IDF, ANN, Digital media authenticity**

## I. INTRODUCTION

The rise of deepfakes and AI-generated content has transformed the way we perceive digital media. Videos, images, and text that look authentic can now be fabricated within minutes, thanks to advances in Generative Adversarial Networks (GANs) and large language models. While these technologies have opened doors to creative and educational applications, they have also made it increasingly difficult to trust what we see and read online. From fake news videos and forged images to plagiarized or AI-written text, the spread of synthetic media poses risks of misinformation, academic dishonesty, and even fraud. This growing challenge demands solutions that can verify authenticity quickly, reliably, and across multiple types of content.

### A. Scope

The scope of this project is to create a unified AI-powered system that can detect manipulated or AI-generated content in videos, images, and text all within a single platform. Unlike existing systems that focus on only one type of content, the AI Presence Detector integrates multiple detection pipelines into one accessible web application. It is designed to be free, real-time, and user-friendly, ensuring that anyone from a journalist verifying a video to a teacher checking for plagia-rism can easily assess digital content. The system combines computer vision models for video and images with machine learning models for text, offering detailed authenticity scores and explanatory feedback.

### B. Challenges

Building such a system comes with several challenges. One major limitation of current approaches is fragmentation tools are scattered across platforms, each handling only one media type. Many of these systems are locked behind enterprise paywalls or designed as academic prototypes, making them inaccessible to the general public. Another challenge is real¬time detection; while many models perform well in controlled experiments, they struggle with speed and scalability when applied in real-world scenarios. Lastly, most detection tools provide only a binary decision of “real” or “fake”, without explaining the level of manipulation or offering confidence scores. This lack of

transparency makes users hesitant to trust the results.

C. Impact [1][9]

The proposed system addresses these gaps by offering a transparent, reliable, and accessible solution. Its impact spans multiple domains: journalists can verify news media before publishing, educators and students can ensure originality in academic work, and forensic experts can authenticate digital evidence with higher confidence. Most importantly, it empowers the general public with a free and user-friendly tool to check the authenticity of digital content they encounter daily. By bridging the gap between advanced AI research and practical usability, the AI Presence Detector has the potential to restore trust in digital media and act as a safeguard against the harmful spread of misinformation.

## II. LITERATURE REVIEW

[1]- [6] The following table summarizes existing research in the field.

| Ref. | Models Used / Methodology | Performance Metrics & Scores | Datasets Used | Key Contributions | Limitations |
|---|---|---|---|---|---|
| [1] | * DL: CNNs (Xception-Net, ResNet, VGG), RNNs (LSTM, BiRNN)<br>* ML: SVM, LR, k-MN, MLP | * DL Methods: Mean Acc: 89.73%, Mean AUC: 0.917<br>* ML Methods: Mean Acc: 86.86%, Mean AUC: 0.909 | * FaceFor en-sics++ (FF++)<br>* Celeb-DF<br>*DFDC dataset | * Provides a taxonomy of detection methods.<br>* Confirms CNNs are most effective.<br>* Identifies lack of standardized comparison framework. | * Limited by unstandard-ized metrics and datasets in studies. |
| [2] | * Generation Models: Autoencoders, GANs<br>* Detection Models: CNNs (XceptionNet), GMMs, DNNs, RNNs, C-LSTMs | * XceptionNet (cited): 99.7% Acc (FF-DF), 65.3% Acc (Celeb-DF)<br>* IBMM (Case Study): 95.87% A U C ( DFDC), 98.1% Acc (DF-TIMIT LQ) | * FF++<br>* Celeb-DF<br>*DFDC dataset<br>*Deeper-Forensics-1.0<br>* ForgeryNet WildDeepfake | * Bridges gap by reviewing deepfake generation techniques.<br>* Highlights challenges: generalizability, robustness, interpretability. | * Does not cover signal-level or transfer learning approaches.<br>* May fail on unseen deepfake types. |
| [3] | * Model: Ensemble CNN (ECN-MF) using soft voting.<br>* Methodology: Combines RNN, 1D CNN, LSTM, ConvLSTM. | for-original: 99.5% Acc<br>* for-norm: 98% Acc<br>* for-2sec: 96.9% Acc<br>* for-rerec: 92.8% Acc<br>* for-merged: 98% Acc | * Fake-or-Real (FoR) dataset & its 4 sub-datasets | * Novel deep ensemble model for audio deepfake detection.<br>* Outperforms individual models. | * High computational cost and latency. |
| [4] | * Model: BMNet<br>* Methodology: BiLSTM + Multi-Head Self-Attention (MHSA) | * FF++: 95.54% Acc, 98.60% AUC<br>* Celeb-DF: 80.20% Acc, 75.72% AUC<br>* DFDC: 84.72% Acc, 88.51% AUC | * FF++<br>* Celeb-DF<br>*DFDC dataset | * Novel BMNet for temporal inconsistencies.<br>* High performance using only facial landmarks. | * Model reliant on BiL-STM; replacing with simple RNN causes 6.3% drop.<br>* Heavy data augmentation may reduce accuracy. |
| [5] | * Spatial Domain Models: Forensic-based (PRNU, ELA), Visual Artifact-based (CNN, XceptionNet, MesoNet).<br>* Temporal Domain Models: RNNs, LSTMs, biological signal-based (SVM). | * FF++: 91% Acc, 95% AUC<br>* Celeb-DF: 81.23% Acc, 73.23% AUC<br>* DFDC: 87.68% Acc, 84.51% AUC | * FaceForen sics (FF/FF++)<br>* Celeb-DF<br>*DFDC dataset | * Offers taxonomy of feature domains.<br>* Analyzes pros/cons of various models. | * Many techniques fail on real-world low-quality data.<br>* Compression loss remains a major issue. |

| | | | | | |
|---|---|---|---|---|---|
| [6] | * Supervised ML: SVM, Random Forest, BiLSTM, CNN, Logistic Regression<br>* Unsupervised ML: K-Means, DBSCAN | * BiLSTM: 88% Acc<br>* Random Forest: 0.86 F1-score, 85% Recall | *PAN datasets (2009–2018)<br>* Corpus of English Novels | * First SLR focused on Intrinsic Plagiarism Detection (IPD).<br>* Highlights challenges in low-resource settings. | * IPD lacks reference collection for comparison.<br>* Difficult to separate genuine style shifts from plagiarism. |

Table I: Critical Literature Review Summary

## III. METHODOLOGY

### A. Research Process

Problem Identification: Studied fragmentation in existing systems (video-only or text-only) and lack of real-time verification.

Dataset Collection: Selected benchmark datasets: FaceForen-sics++ (manipulated & original videos), Celeb-DF (realistic celebrity deepfakes), AI vs Human Text Dataset (GPT-2/3 vs human-authored).

Preprocessing:

- Video: Frame extraction, normalization. [7]
- Image: Resizing, normalization, augmentation.
- Text: Cleaning, tokenization, TF-IDF vectorization.

Model Selection: CNNs for video/image detection, Hugging Face Transformers for embeddings, ANN for text classification.

Evaluation Metrics: Accuracy, precision, recall, F1 score, confusion matrix.

### B. Detection Pipelines

1) Video Detection Pipeline: [12] User uploads a video file.
a) Extract frames using OpenCV.
b) Pass each frame to a CNN model trained on FaceForensics++ and Celeb-DF datasets.
c) Extract embeddings using Hugging Face pretrained deepfake detection models.
d) Apply majority voting across frames:

$$\text{Final\_Label} = \text{argmax} \sum_{i=1}^{n} \text{Prediction (Frame}_i\text{)} \quad (1)$$

e) Output: Classification (Real/Fake) with probability score.

2) Image Detection Pipeline: [8] User uploads an image.
a) Image normalized and resized.
b) Forwarded through CNN binary classifier.
c) Output: Real/fake classification with manipulation probability:

$$P(y = \text{Fake}|X) = \sigma(W \cdot X + b) \quad (2)$$

where σ is sigmoid activation, Wa re weights, X input features, and b is bias.

3) Text Detection Pipeline: [5] User enters/pastes text.
a) Text cleaning (stop-word removal, tokenization).
b) Convert text into numerical features using TF-IDF:

$$\text{TFIDF}(t, d) = \text{TF}(t, d) \times \log \frac{N}{\text{DF}(t)} \quad (3)$$

where TF(t,d) = frequency of term t in document d, DF(t) = number of docs containing t, and N = total docs.
c) Features passed into an ANN with input, hidden, and output layers.
d) ANN classifies text as human-written, AI-generated, or plagiarized.

### C. Implementation Methodology [2], [4]

Frontend (React.js) The frontend of the system is designed using React.js, enabling users to upload media easily through a drag-and-drop interface. A separate text input field allows users to check plagiarism or AI-generated content instantly. Real-time outputs such as result labels, confidence scores, and feedback are displayed through a dynamic dashboard.

Backend (Flask) The backend, developed using Flask, executes the trained models, processes input data, and returns the analyzed results. It also manages file privacy and automatically deletes temporary data after each session, ensuring both security and efficiency.

1) Machine Learning Methods Used: The following machine learning models were employed for content analysis and classification.

- Support Vector Machine (SVM): In this architecture, SVM acts as a robust binary classifier for high-dimensional feature vectors. It is used to draw an optimal decision boundary between" authentic" and" manipulated" samples, particularly when processing the refined feature embeddings extracted

from the Video and Image pipelines.

- Naive Bayes: This algorithm is integrated into the Text Detection Pipeline to handle rapid, probabilistic classification. By calculating the likelihood of specific linguistic patterns appearing in AI-generated vs. human-written text, it provides a fast baseline for identifying structural anomalies in textual input.

- KNN (K-Nearest Neighbors): KNN is employed as a similarity-based validator. By comparing the feature distance of a new input against a pre-labeled dataset, KNN helps identify content that closely matches known patterns of forgery or plagiarism, providing an additional layer of evidence for the final ensemble result.

2) Deep Learning Methods Used: The following deep learning architectures are integrated to improve detection ac¬curacy across images, videos, and text.

- CNN (Convolutional Neural Network): [14] CNNs are the primary engines for the Image and Video Pipelines. They are used to perform hierarchical spatial analysis, specifically designed to detect" micro-anomalies" such as inconsistent lighting, pixel-level artifacts, or unnatural facial movements (like abnormal blinking) that signify deepfake manipulation.

- Hugging Face Transformers: [9] These models are used to generate contextual embeddings. In the Text Pipeline, they capture long-range semantic dependencies to identify the" mechanical" flow of AI-generated prose. In the Video Pipeline, they work alongside CNNs to ana¬lyze the temporal relationships between frames, ensuring consistency across the entire sequence

- ANN (Artificial Neural Network): The ANN serves as the primary classifier within the Text Detection Pipeline. It receives the numerical feature vectors (TF-IDF) as input and processes them through multiple hidden layers to categorize the text into three distinct classes: Human-written, AI-generated, or Plagiarized.

- Term Frequency–Inverse Document Frequency (TF-IDF): [4] This is the preprocessing backbone for text analysis. It transforms raw text into a weighted numerical format that highlights unique," signature" keywords. These vectors are then fed into the ANN and SVM to allow the models to focus on the most statistically significant terms in the document.

- Majority Voting (Ensemble Method): [5] This serves as the decision logic for the Video Pipeline. Since a single video contains thousands of frames that may vary in quality, the system records a prediction for every individual frame. Majority Voting aggregates these labels, ensuring the final" Real" or" Fake" output is based on the most consistent result across the entire duration, reducing the risk of false positives from motion blur.

## IV. DATASETS AND METRICS

To build a robust system that can detect deepfakes and AI-generated text across multiple formats, it was essential to use diverse and well-established datasets. Each dataset was carefully chosen to cover a specific type of content videos, images, or text and to ensure that the models trained on them would generalize well to real-world scenarios. Once the datasets were finalized, appropriate metrics were selected to measure the performance of the system in a transparent and reliable way.

- FaceForensics++: [7] This dataset is a widely used benchmark for deepfake detection, containing over 500,000 manipulated videos and 1,000 original videos. It includes both high-quality (HQ) and low-quality (LQ) versions, which helped train our CNN models to be robust against real-world video compression and degradation. The variety of forgery techniques represented in this dataset was invaluable for building a generalized detec¬tion model.

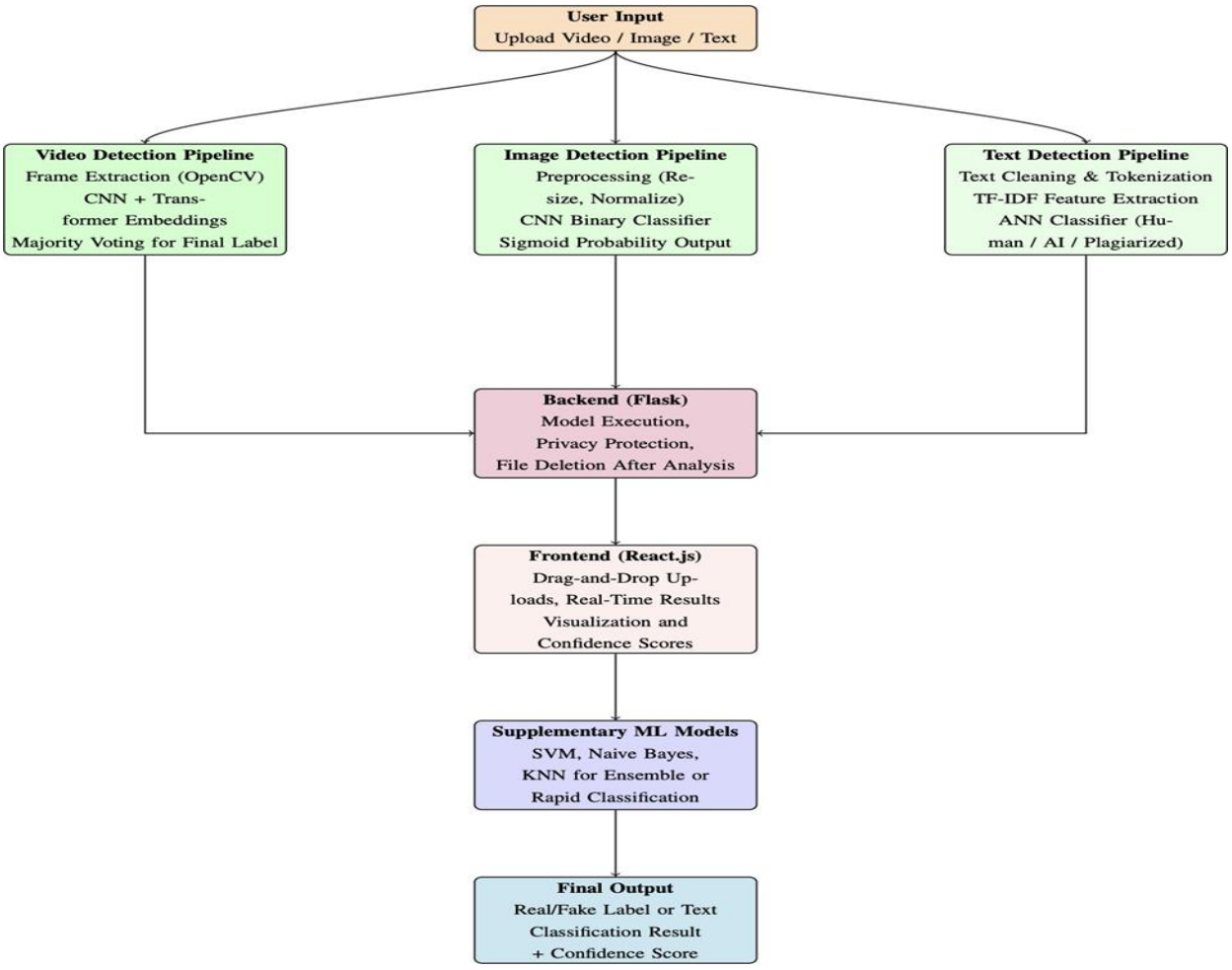

Fig. 1: Methodology flowchart

- Celeb-DF: [2] To test our models on more subtle and realistic manipulations, we used the Celeb-DF dataset. It consists of 5,639 deepfake videos of celebrities, capturing fine details like facial expressions and blinking patterns that are often found in deepfakes on social media. While highly realistic, a limitation of this dataset is its focus on celebrity faces, which may not fully represent manipulations of the general public.
- AI vs Human Text Dataset: [4] For text analysis, we created a custom dataset for detecting AI-generated content and plagiarism. This hybrid dataset contains thousands of samples, including human-written essays and articles, alongside text generated by models like GPT-2, GPT-3, and ChatGPT. All text samples were preprocessed through tokenization, cleaning, and TF-IDF vectorization to prepare them for our ANN classifier.

All datasets were split into a 70% training set, a 20% validation set, and a 10% testing set to ensure a fair and robust evaluation of our models.

Metrics:

To evaluate the performance of the system, standard machine learning metrics were applied. These metrics not only measure the correctness of predictions but also help understand how well the model balances between false positives and false negatives.

A. Accuracy

Accuracy measures the percentage of correctly predicted samples across all classes. It is the simplest metric but may not be sufficient when classes are imbalanced.

B. Confusion Matrix

The confusion matrix provides a complete picture of classification results, showing the counts of:

- TP - True Positive: Count, when actual label is

positive and your model predicts it positive
- TN - True Negative: Count, when actual label is negative and your model predicts negative
- FP - False Positive: Count, when actual label is negative and model predicts positive
- FN - False Negative: Count, when actual label is positive and model predicts negative

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall/Sensitivity/True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$

$$\text{F1 score} = \frac{2TP}{2TP + FP + FN}$$

$$\text{False Positive rate (FPR)} = \frac{FP}{FP + TN}$$

The F1 Score balances Precision and Recall into a single number, making it useful when both false positives and false negatives matter.

C. Model Loss and Model Accuracy Curves

For neural networks (CNNs and ANNs), we also track training vs. validation loss and accuracy curves. A well-trained model should show decreasing loss and stable accuracy across both sets without overfitting.

D. Frames Per Second (FPS) for video

[19] Since real-time performance is a key goal, we also measured frames per second (FPS) while processing videos. Higher FPS ensures that videos can be analyzed quickly without long delays, making the system practical for real-world use.

## V. CONCLUSION

The AI Presence Detector was developed to tackle one of today's most pressing digital challenges: verifying the authenticity of online content. By combining CNNs for visual detection, Hugging Face transformers for embeddings, and TF-IDF with ANN for text analysis, the system delivers a unified, real-time solution for detecting manipulated videos, images, and text. Unlike fragmented tools that only handle one medium or provide limited results, this project emphasizes accessibility, transparency, and usability. With its ability to provide confidence scores and detailed feedback, the system not only identifies fake content but also helps users understand the reasoning behind the results. Ultimately, it serves as a step towards restoring trust in digital media and combating misinformation in society.

## VI. FUTURE POTENTIAL DEVELOPMENT

While the system provides a strong foundation, there are several ways it can be extended and improved in the future: Audio Deepfake Detection – Extend pipelines to analyze speech and detect voice manipulation using spectrograms and audio-based CNN/RNN models. Real-Time Live Stream Verification – Enable real-time monitoring of video streams (e.g., news broadcasts or online meetings) to flag manipulated content instantly. Cloud Deployment and Scalability –Deploy the platform on AWS, GCP, or Azure with Docker and Kubernetes, allowing large-scale usage by thousands of users simultaneously. Mobile Applications – Build lightweight mobile apps using React Native or Flutter, making detection tools accessible on smartphones.

## ACKNOWLEDGMENT

The above-used figures and the information is the best to the knowledge and has been sourced from the online resources and the references mentioned.

## REFERENCES

[1] N. U. R. Ahmed, A. B. A. Daud, T. Alsahfi, H. Adeel, and A. Tajammul, "Visual deepfake detection: Review of techniques, tools, limitations, and future prospects."

[2] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake generation and detection: Case study and challenges."

[3] Xiong, Z. Wen, C. Zhang, D. Ren, and W. Li, "BMNet: Enhancing deepfake detection through BiLSTM and multi-head self-attention mechanism."

[4] M. F. Manzoor, M. S. Farooq, M. Haseeb, U. Farooq, S. Khalid, and A. Abid, "Exploring the landscape of intrinsic plagiarism detection: Benchmarks, techniques, evolution, and challenges."

[5] Ali, J. Rashid, M. R. U. Hussnain, M. U. Tariq, A. Ghani, and D. Kwak, “Beyond the illusion: Ensemble learning for effective voice deepfake detection.”
[6] M. U. T. Gujjar, K. Munir, M. Amjad, A. U. Rehman, and A. Bermak, “Unmasking the fake: Machine learning approach for deepfake voice detection.”
[7] Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to detect forged facial images.”
[8] J. Goodfellow et al., “Generative adversarial nets.”
[9] L. Verdoliva, “Media forensics and deepfakes: An overview.”
[10] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks.”
[11] Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: A compact facial video forgery detection network.”
[12] Y. Li, M.-C. Chang, and S. Lyu, “Exposing deepfake videos by detecting face warping artifacts.”
[13] X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses.”
[14] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, “Exposing AI-created fake videos by detecting eye blinking.”
[15] Masi, A. Killekar, R. Rodrigues, S. Pnaso, and W. AbdAlmageed, “Two-branch recurrent network for isolating deepfakes in videos.”
[16] Cozzolino, K. Nagano, and L. Verdoliva, “ID-Reveal: Identity-aware deepfake video detection.”
[17] Lee, K.-S. Kim, S.-J. Son, and J.-M. Seo, “Deepfake audio detection using audio-visual cues.”
[18] X.-F. Wang, S. H. G. Chan, W.-Y. Wang, and W. Wang, “Differential and channel-invariant attention for deepfake voice detection.”
[19] J.-C. Chen, W.-C. Tseng, H.-M. Wang, W.-C. Huang, and T. Toda, “General-purpose deepfake voice detection.”
[20] N. Agarwal, H. K. Triath, A. K. Vatsa, and T. Singh, “Deep-in-the-sky: A deep learning approach for detecting deepfake in satellite imagery.”