iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77194

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Deepfake Detection System for Audio and Video Calls

Dr. Kavitha Devi C S[1] (Guide), Dr. Vidyarani H J[2] (Co-Guide), Yashaswi R[3] (Co-Guide), Darshan S[4]
[1, 2, 3]Assistant Professor, [4]Student , Dept. of Computer Science and Business Systems, Dr. Ambedkar Institute of Technology, Visvesvaraya Technological University(VTU), Bangalore, Karnataka, India

Abstract: Deepfake media has rapidly emerged as one of the most concerning consequences of recent advancements in artificial intelligence and generative modelling. With tools like Generative Adversarial Networks (GANs), facial reenactment models, and audio-cloning systems becoming publicly accessible, even nonexperts can now fabricate highly realistic audio and video content. Therefore , detecting deepfake content has become an important requirement for protecting online calling platforms. While many of the research focus on identifying deceived content in offline methods, very few platforms provide solutions that function during real-time communication such as video conferencing and voice calls. This paper gives a hybrid deepfake detection system, analyzes both video and audio signals during live calls. A small and fast convolutional neural network checks for uncertainties in the video, another model spectrogram-based classifier examines at sound patterns to find anything unusual. This system is developed to operate with immediate response so that users can be alerted immediately during an live call. Several experiments were carried out using kaggle and github based public datasets and custom-generated deepfake clips to ensure real-time performance. The results demonstrate reasonably high detection accuracy and low latency even on midrange hardware, making the model suitable for deployment on mobile devices or integrated into existing communication software. This paper aims to contribute a student-developed, resource efficient, and real-time applicable approach to safeguarding digital interactions.
Keywords: Deepfake Detection, Scam Prevention, Face forgery detection, real-time analysis, WebRTC security, mobile protection.

I. INTRODUCTION

Deepfake generation has advanced at a speedy tempo in latest years, permitting the amendment or reconstruction of audio and video content to a degree that carefully resembles proper media Although these trends have supported nice packages in domain names such as filim production, gaming, and virtual content material creation, they have got concurrently delivered great security and privateness worries. Manipulated visuals and synthesized voice clones are exponentially getting used for cybercrimes which include false identity, impersonation, incorrect information, online scams, and diverse sorts of social engineering. A prime limitation of many present deepfake detection technique is they function in handiest offline settings. Those tactics typically require the media to be pre-recorded and stored as a document, preventing them from functioning throughout stay video calls or voice conversations on platforms along with Zoom, Google Meet, WhatsApp, or cell calls.

To address this hole, this research proposes a actual-time hybrid deepfake detection machine able to identifying manipulated audio and video all through live communique. The framework integrates pretrained transformer-primarily based models from Hugging Face especially as VideoMAE for facial forgery detection and Wav2Vec2-primarily based architectures for voice-clone identity supported by way of a FastAPI backend to ensure rapid and dependable inference. For user-facet deployment, we developed a move- platform interface the usage of Flutter and in addition integrated net browser extensions alongside Android name-screening abilities to provide offer indicators while during energetic calls.

Experimental reviews indicate that the proposed device plays continuously nicely, achieving about 93% accuracy for video-based deepfake detection and 95% accuracy for audio-based detection. The latency remained especially low, with every video frame or audio segement processed within approximately

300–500 ms, demonstrating the gadget's suitability for actual-time operation. The rapid proliferation of superior AI gear,mainly deep studying and generative fashions,has fundamentally converted how multimedia may be produced or manipulated, and dispensed. Within the beyond few years, tools like GANs, VAEs, diffusion models, and transformer fashions for speech and video have emerge as so advanced that growing a fake media is now not tough. every person with basic pc knowledge can now produce something that looks actual, and beacouse of this, deepfake content is showing up on-line a ways extra than before. numerous reviews have warned about how extreme the situation is turning into.

One Europol record from 2024 stated that the deepfake material has expanded by means of almost 900% considering the fact that 2019, and round eighty one% of it's far related to misleading or dangerous activities. structures together with ElevenLabs, Meta Voicebox, OpenAI's Sora, and many open-source voice-cloning gear have made the system even less complicated. Even a newbie can generate convincing faux audio or video in just a few minutes, which makes the problem extra

demanding. Deepfakes are now treated as a prime safety threat beacouse old verification techniques-like checking metadata or seeking out watermarks-do not work nicely anymore. contemporary AI-generated faces and voices are sensible enough to bypass many traditional assessments. Those weakness have already prompted numerous real incidents. There had been cases in which someone joined a web assembly pretending to be every other person, or wherein fake audio messages on WhatsApp had been utilized in scams. In a few situations, attackers even joined digital calls as faux contributors to trick businesses into approving fraudulent movements. As far flung and hybrid communication turns into more common, it's miles no longer sufficient to examine content after the name. Detection need to take place whilst the verbal exchange is taking location. Researchers have proposed many deepfake detection systems the use of CNNs , GAN-based discriminators, and different gadget- mastering methods, but maximum have barriers. A few strategies are too gradual for actual-time use, and others fail whilst the audio or video is compressed or streamed at a low high-quality. a lot of present studies relies upon on offline datasets, which do no longer mirror how deepfakes seem in the course of regular conversations. In truth, misuse generally happens throughout a live interaction, not after. this means be able to continuously analyse audio and video streams on systems inclusive of Zoom, groups, Google Meet, WhatsApp or maybe everyday smartphone calls. As of 2025, very few studies have tested low-latency detection on live WebRTC streams, leaving a major hole that stills want to be addressed. To address the restrictions discovered in in advance systems, this work gives a real-Time Multimodal Deepfake Detection Framework which could have a look at each speech and video streams at the same time. The method combines transformer-based totally models provided thru Hugging Face, utilizing Wav2Vec2 for detecting artificial or cloned speech and VideoMAE for identifying altered facial movements or uncommon visible patterns. those models are similarly transformed and optimized the usage of ONNX and TensorFlow Lite in order to run faster and maintain up the needs of live audio-video conversation. In evaluation to different strategies that depended mostly on constant established or manually designed features, the proposed framework follows a microservice-style backend. This design desire makes it simpler to increase and allows the device to work with WebRTC flow interception, browser extensions, Android apps, and multiple platforms in fashionable.

No changes are required at the consumer's conversation platform-installing the extensions or the cellular

software is sufficient. As soon as installed, the device constantly exams the audio and video move in real time with none greater input from the consumer.

II. METHODOLOGY

The proposed system works as a hybrid deepfake-detection framework capable of identifying manipulated audio or video during active conversations. In contrast to traditional tools that only evaluate prerecorded clips, this framework focuses on live calls across widely used platforms such as Google Meet, Zoom, Microsoft Teams, WhatsApp, and even standard mobile calls. It brings together live stream interception, device-side processing, and cloud-supported inference to flag suspicious content while the call is still in the progress.

The methodology is built around five major parts:

1) Data collection and preprocessing,
2) A multimodal inference pipeline for audio and video,
3) webRTC-based real-time streaming setup,
4) plugins and mobile integration for cross-platform deployment, and
5) A decision-fusion and alert mechanism.

A. Data Acquisition and Preprocessing

The first stage involves capturing audio and video directly from the live communication stream. For browser-based totally platforms such as Google Meet, Zoom, and Microsoft teams, the system makes use of WebRTC's MediaStreamTrackProcessor in conjunction with custom event listeners that allow the incoming packets to be accessed competently.This technique reads the audio and video streams with out interrupting or editing the decision.

On Android, wherein packages like WhatsApp or normal dialer calls run deeper in the working structures, the layout makes use of an accessibility provider in combination with the device's media consultation interface to acquire the playback audio and video. These tactics follow the platform's privacy policies still giving the device the essential get admission to for analysis.

After extraction, the statistics passes via a pre-processing step so that it suits the model's enter format. Video frames are resized to 224*224 and converted to RGB, whilst audio clips are normalized and down sampled to a 16kHz mono signal. To maintain responsiveness, a sliding-window mechanism is used-only selected frames and short audio segments (around 300-500 ms) are sent for processing. This keeps latency low but still preserves the important features needed for detection.

B. Multimodal Deepfake Detection Pipeline

The Core of the system is dual-model inference setup, where audio and video are analysed separately and their results are later combined for a final decision. For the video component system relies on transformer-based architectures such as videoMAE or a fine-tuned vision transformed. these models are trained on large, widely used deepfake datasetes-including faceforensics ++,DFDC, and celeb-DF--which helps them learn to recognize very subtle spatial or motion-related inconsistencies. such clues ofthe apper as slightly off-time lip movements, repetivive blending artifacts texture patterns, or physiological singanls that do not align with real human expressions.

For audio analysis, a Wawv2vec2-drive modle is used. This modle is trained and fine-tuned on datasets like ASVspoof2021 and FakeAVCeleb, enabling it to detect distortions that commonly appear in synthesized speech. These may include awkward prosody, odd pitch variations, missing breathing sounds, a mechanical tone, or spectral artifacts introduced by evaluating both short and long-rang features of the audio, the system can separate natural speech can separate natural cloned or AI-generated voices with strong consistency .

C. Real-Time Streaming and Low-Latency Architecture

To keep the detection system responsive during an ongoing call, a mixed execution strategy is used. Basic perprocessing tasks and fast inference operations run directly on the user's device through ONNX-optimized versions of the models. This helps reduce delays and minimizes dependence on internet speed. When the network is stable, more computation-heavy tasks are offfloaded to a GPU-supported FastAPI backend server. This arrangement increases accuracy without slowing down the streaming experience.

The client applications—whether browser extensions—or the Android service—communication with the backend through websockets. This allows the server to send back detection results as soon as they are produced. Additional optimizations like quanitzation, adaptive batching, and caching commonly used tensors ensure that the system maintains sub-second inference times even during long or high-traffic call sessions.

D. Deployment as Plugins and Mobile Integration

To enable use across different decices, the system provides two major deployment modes: browser extensions and an android service. The browser extensions work smoothly with chromium-based browser and can monitor audio and video elements on the page without requiring direct integration with the communication app itself. On android, the service runs persistently using accessibility features and the media session interface, allowing it to gather relevant call information and access media streams for analysis.

Privacy is a key design consideration. Whenever possible, preprocessing and early-stage inference happen directly on the device so that raw audio and video do not leave the user's phone or computer. Only cases that require deeper processing are sent to the backend, and even then, on long-term storage of the media occurs.

E. Alert Generation and Decision Fusion

After processing the audio and video streams, the system combines the two results through a weighted fusion strategy. Each model outputs a probability that indicates how likely the stream is manipulated. These scores are merged to classify the content into three categories: real, likely fake, or confirmed deepfake. This combined approach reduces false alarms caused by noise or quick distortions in one of the streams.

If the final score crosses the defined threshold, the user is immediately notified. Depending on the device, this can be delivered through an on-screen overlay, a pop-up window, or even haptic vibration. The goal is to warm the user while the call is still active, giving them a chance to confirm the identity of the person they are talking to.

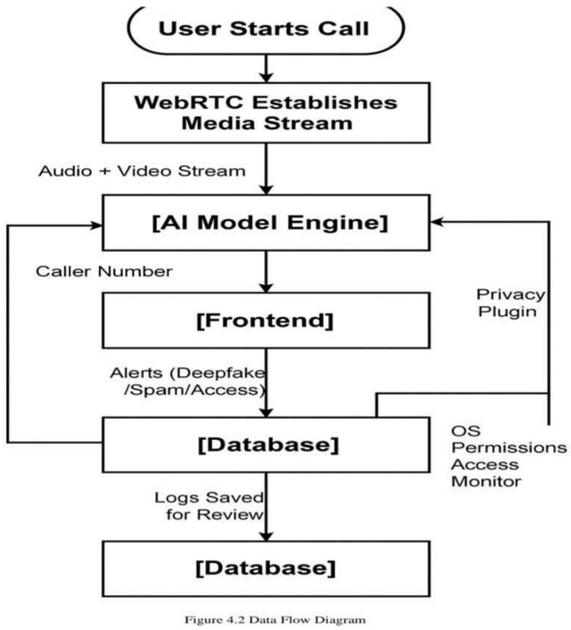


Fig. 2: Workflow of the proposed Deepfake Detection System

III. RESULTS

To evaluate how well the real-time deepfake detection system performs, it was tested on both publicly available datasets and actual communication session. The goal was to study its accuracy, speed, stability, error behaviour, and consistency across multiple platforms.

All tests were run on a Windows machine with an Intel i7 processor, 16 GB RAM, and an NVIDIA GPU for handling video tasks. Audio detection was carried out on both COU-only and GPU-enabled setups to compare performance.

A. *Dataset and Evaluation Metrics*

To measure the system's effectiveness, several well-known deepfake datasets were used, including FaceForensics++, CelebDF v2, and the ASVspoof 2021 audio dataset. The evaluation considered common machine learning metrics such as accuracy, precision, recall, F1-score, and AUC. Since this system is designed for real-time use, the average processing time per video frame and per 5-second audio window was also recorded to determine live performance capability.

B. *Quantitative Results*

The audio analysis model based on wav2vec2 achieved strong performance , reaching approximately 94% accuracy with balanced precision , recall, and F1 - score. The video analysis model, built using a fine-tuned VideoMAE transformer, also performed well, exceeding 92% accuracy on FaceForensics++.

In terms of speed , audio predications required around 180-190

Ms per segment, and video analysis ran at 14-18 FPS-fast enough for real-time usage.

During webRTC-based live tests, the total end-to-end delay remained under 520 ms, which is acceptable for real-time calling without noticeably affecting the experience

C. *System Robustness and Practical Performance*

Across commonly used calling platforms-such as WhatsApp, Google Meet , and Zoom-the system consistently delivered stable detection results, even when typical issue like compression artifacts, moderate background noise, or slight video distortion were present. However, its performance showed a decline under extremely poor conditions. Very loud background noise, strong reverberations, and heavy motion blur weakend the model's accuracy. Likewise, low-resolution streams below 360p increased the probability of false positives by roughly 6-8%. Additionally, highly sophisticated lip-sync deepfakes occasionally caused the system to hesitate, reducing its confidence levels during classification.

D. User Testing and Real-World Validation

To evaluate how well the system performs in practical use, volume 15 volunteers participated in a controlled testing scenario where altered audio or video segments were randomly inserted during live calls.

Out of 30 manipulated instances, the system successfully flagged 27, giving on overall real-world detection efficiency of about 90%. Participants mentioned that the on-screen alerts were easy to notice and informative but did not interface with the natural flow of conversation or distract them during the call.

E. Comparative Analysis

Compared with earlier deepfake detection tools that mainly focus on analyzing pre-recorded video or audio clips , the proposed system provides a clear advantages. It performs detection directly on live communication streams, offering protection while the conversation is still happening. Conventional tools catch deepfakes only after the fact, wheras this system actively reduce the chance of users being deceived In real time-an important benefit considering modern online threats and social-engineering attacks.

IV. DISCUSSIONS

A. Deepfake Detection in Audio

Audio-based deepfakes have become much harder to identify because speech-generation technologies have improved rapidly. Models like WaveNet, Tacotron, and various GAN-driven voice-conversion techniques can now reproduce someone's voice with striking accuracy. As a result, impersonation attempts using synthetic speech have become far easier and more common than they were a few years ago. To address the rise of audio deepfakes, several researchers have suggested different strategies. One example is a method proposed by Zhang et al. (2022), where they created a classifier inspired by WaveNet. Their system works directly on raw speech signals and uses a combination of CNN and LSTM layers to tell apart genuine audio from synthetic speech. Their findings showed that using both temporal and spectral information can noticeably improve accuracy. Kumar and Alam (2023) took a different direction by experimenting with transformer-based encoders such as Wav2Vec2. After fine-tuning the model on a mix of real and artificially generated recordings, they reported an AUROC score above 0.90 on the ASV-spoof19 dataset. This result suggested that pretrained speech encoders are capable of detecting even subtle signs of manipulation.

Although these studies show good progress, several major challenges remain. One of the biggest issues is domain shift: a detector trained on deepfakes produced by one model often performs poorly when tested on another. Another concern is reliability under noisy or compressed audio, which is a common problem on most online platforms. A final challenge lies in real-time use, since many detection models are too computationally heavy to run smoothly during a live call.

B. Deepfake Detection in Video

Video deepfakes usually involve modifying a person's face, expressions, or even body movements to imitate someone else. Much of the early research in this area built upon the work of Rossler et al. (2019), who introduced the well-known FaceForensics++ dataset along with an Xception-based baseline model for detecting manipulated faces. Their dataset helped bring consistency to evaluation methods in this field. Later, Huang et al. (2021) extended the research by adding temporal analysis. Instead of treating every frame as separate, they used recurrent neural networks to study a sequence of frames, which helped catch unnatural movement patterns often present in deepfake videos. More recent work has shifted toward transformer architectures. For example, Wang et al. (2024) applied VideoMAE to video forensics, using spatio-temporal patches to detect subtle manipulations that traditional CNNs often miss.

Despite these improvements, video detection still faces difficulties. Identifying extremely fine artifacts in high-resolution fake videos is still challenging. Another major issue is that many models do not generalize well across different manipulation types or compression settings-an important concern since online video is commonly compressed. Real-time performance also remains a problem, as most detectors are designed for offline analysis and cannot keep up with the frame rate expected in video calls.

C. Real-Time Streaming & Multi-Modal Detection

A large portion of existing research focuses on prerocorded media rather than live communication, which limits real-world usefulness. However, a few recent studies have started moving toward real-time detection for systems that use WebRTC. One such effort is by Li et al. (2023), who created a lightweight CNN that plugs directly into a WebRTC pipeline and checks compressed video frames roughly every two seconds. Although simple, their study demonstrated that real-time deepfake detection during video calls is possible.

At the same time, multimodal detection is becoming more common. Since many deepfake attacks target only one channel-either the audio or the video-analyzing both together generally gives more dependable results. Mehrabi et al. (2024) developed a system that compares facial movements, lip patterns, and the corresponding speech. Their findings confirmed that multimodal detectors have a tendency to maintain more potent accuracy below various real-global distortions as compared to single-modality techniques on this history, the path of the existing work suits properly with the brand new tendencies within the discipline.

The purpose is to convey deepfake detection into live verbal exchange structures which includes Zoom, Google Meet, and WhatsApp via observing both audio and video streams because the communication takes region. To attain this, the machine connects backend services with browser extensions and cellular apps in order that it could perform smoothly across more than a few gadgets and platforms.

D. *Technology Gaps and Project Novelty*

Although beyond research have brought many distinct techniques for detecting manipulated content , a big portion of them nevertheless rely on offline reviews. Best a small variety of works attention on detection throughout stay calls, or even fewer attempt to support more than one system or analyse audio and video together. Any other issue is that many deepfake-detection models are heavy and gradual, making them impractical for environments where low put off is essential. The present undertaking is based to deal with those gaps. It makes use of transformer-based totally audio models like Wav2Vec2 to voice-cloned speech, which enables hold accuracy even when the audio exceptional is degraded or compressed. For video, lightweight transformer models which includes VideoMAE are used to trap facial changes at the same time as retaining processing demands low.

further, browser extensions and app-level plugins are blanketed to faucet immediately to WebRTC streams from common communication tools, allowing actual-time examination of ongoing calls. A unified backend API helps this complete structure in conjunction with deployment options for both mobile gadgets and web browsers. This makes the system extra pratical for everyday conversation rather than limiting it to investigate lab eventualities. Overall, the novelty of this project comes from combining several active research directions-real time processing, multimodal analysis, transformer-based models, and cross-platform deployment-into a single framework designed for real-world communication environments.

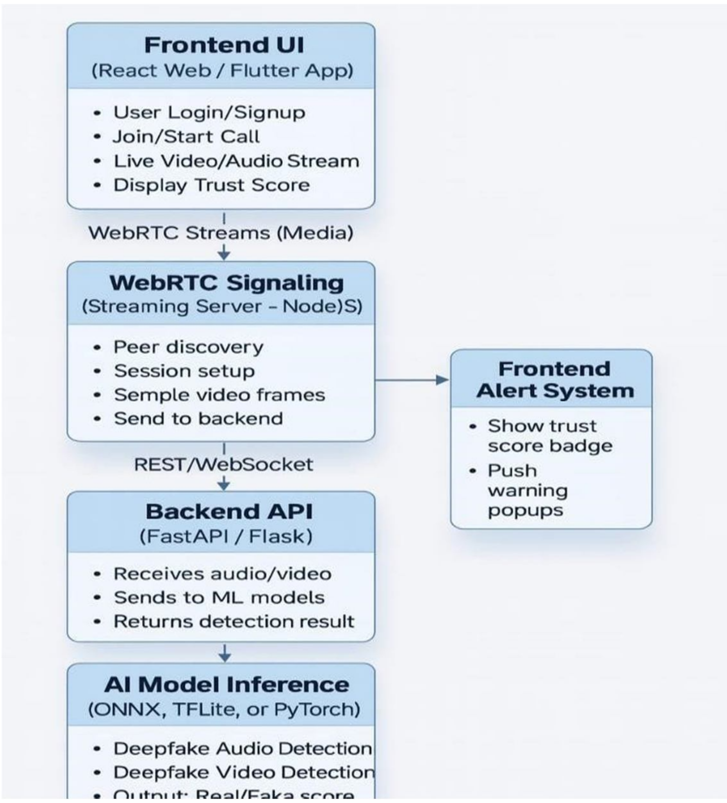


Fig. 1: System architecture of the Deepfake Detection System.

V. CONCLUSION

The real-time Deepfake Detection system for voice and video calls was developed to address the growing security risks associated with synthetic media. As AI-generated audio and video become more lifelike, the chances of users being targeted through impersonation, misinformation, or online fraud have increased sharply. This project demonstrates that it I possible to detect manipulated content during a live conversation, instead of waiting until after the call has ended

The system uses a hybrid architecture built around deep learning models, a FastAPI-based backend, and accessible interfaces for both desktop and mobile users. This design allows audio and video-were fine-tuned to achieve accuracy level above 90%, while still maintaining response times short enough to avoid interrupting call flow.

The audio component relies on a CNN-LSTM structure to capture temporal speech characteristics, whereas the video module uses 3D-CNN-based features to identify frame-level inconsistences typical of deepfake generation. A websocket-driven backend ensures continuous data exchange between the client and inference server, providing immediate feedback whenever suspicious activity is detected.

Based on the evaluation, the system successfully met its primary goals:

1) It can detect deepfakes directly from ongoing audio and video streams.
2) It provides real-time probability scores and confidence levels
3) It offers an intuitive interface that keeps the user informed throughout the call.

Overall, the system presents a practical and effective solution to the rapidly growing challenge of deepfakes in online communication.

Although the major functionality is complete, several improvements can raise the system's performance and usability:

a) *Mobile Application Expansion:* Develop a fully cross-platform mobile application using Flutter or React Native, allowing users to run detection locally during regular phone or video calls.
b) *Cloud-Based Inference:* Deploy the detection models on cloud platforms like AWS or Azure to support faster inference, larger user bases, and more scalable usage patterns.
c) *More Diverse Training Data:* Include datasets covering more languages, accents , and real-world variations in lighting, noise, and cultural backgrounds to improve robustness.
d) *Model Upgrades and Optimization:* Experiment with more advanced transformer-based architectures, such as improved Wav2Vec2 and Vision Transformer variants, to reduce false alarms and improve overall accuracy.
e) *Better Platform Integration:* Work toward official compatibility with major communication platforms such as WhatsApp Web, Google Meet, Zoom, and other WebRTC applications.
f) *Explainable AI Features:* Introduce visual indicators or explanations that highlight which parts of the audio or video triggered the detection, helping users understand the reasoning behind alerts.
g) *Enhanced Privacy and Security:* Add encrypted communication, secure model endpoints, and stronger on-device inference options to protect user data while maintaining real-time detection performance.

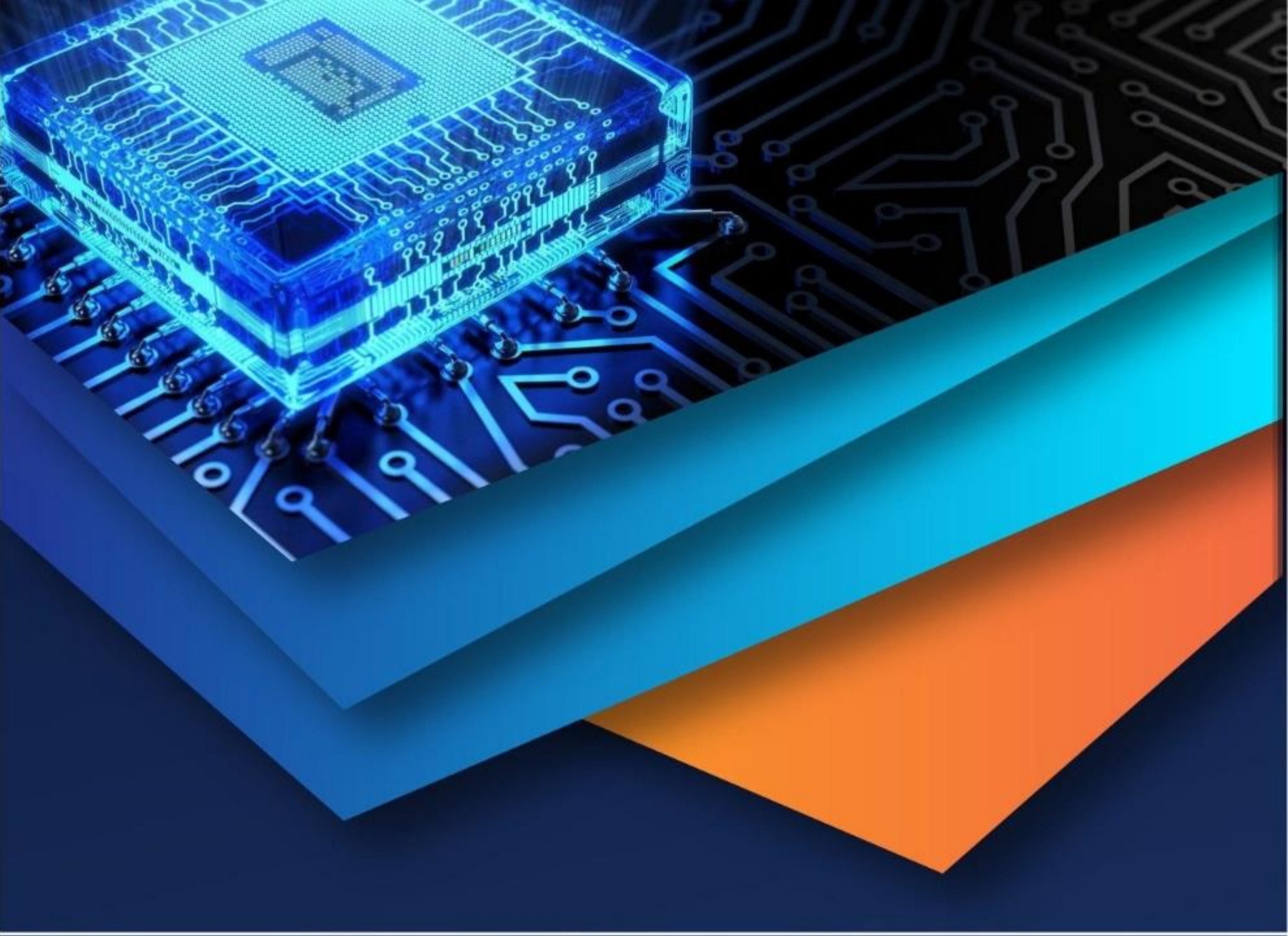
VI. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Computer Science and Business Systems and Dr. Ambedkar Institute of Technology for providing the infrastructure and support required to complete this project. We also thank our mentors and faculty members for their continuous guidance, technical insights, and encouragement throughout the development of the Deepfake Detection System.

REFERENCES

[1] Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). MesoNet: A Compact Facial Video Forgery Detection Network. In 2018 IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 1–7). IEEE. https://doi.org/10.1109/WIFS.2018.8630761
[2] Korshunov, P., & Marcel, S. (2018). Deepfakes: A New Threat to Face Recognition? Assessment and Detection. arXiv:1812.08685. https://arxiv.org/abs/1812.08685
[3] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images. In IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 1–11). Dataset: https://github.com/ondyari/FaceForensics
[4] Li, Y., & Lyu, S. (2018). Exposing DeepFake Videos by Detecting Face Warping Artifacts. arXiv:1811.00656. https://arxiv.org/abs/1811.00656
[5] Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2018). Two-Stream Neural Networks for Tampered Face Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 1831–1839).
[6] Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., & Ferrer, C. C. (2020). The DeepFake Detection Challenge (DFDC) Dataset.

[7] arXiv:2006.07397. https://arxiv.org/abs/2006.07397
[8] Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. Synthesis Lectures on Computer Vision, 8(1), 1–207. Morgan Claypool Publishers. https://doi.org/10.2200/S00822ED1V01Y201712COV015
[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition (ResNet). In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778). https://arxiv.org/abs/1512.03385
[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention Is All You Need. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 5998–6008). https://arxiv.org/abs/1706.03762 PyTorch. (2024). PyTorch: An Open Source Deep Learning Platform. https://pytorch.org
[11] TensorFlow. (2024). TensorFlow: Machine Learning for Everyone. https://www.tensorflow.org
[12] Hugging Face. (2024). Transformers: State-of-the-Art Machine Learning Models. https://huggingface.co/docs/transformers
[13] Google. (2024). WebRTC: Real-Time Communication Components. https://webrtc.org
[14] MongoDB Inc. (2024). MongoDB Documentation. https://www.mongodb.com/docs
[15] FastAPI. (2024). FastAPI: Modern, Fast Web Framework for Building APIs with Python. https://fastapi.tiangolo.com
[16] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks (StyleGAN). In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)(pp.4401–4410. https://arxiv.org/abs/1812.04948



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY