iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78206

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

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 III Mar 2026- Available at www.ijraset.com

A Hybrid Deep Learning Approach for Deepfake Image Detection Using CNN and MTCNN-Based Facial Feature Extraction

Varanasi Triveni[1], Umamaheswararao Mogili[2], J. Vani[3], K. Revathi[4], G. Jyothi Prakash[5], Ch. Kiran Kumar[6], P. Chakri[7]
[1, 2]*Assistant Professor, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*
[3,4,5,6,7]*B.Tech, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

Abstract: ***The rapid growth of digital media and social networks has led to a surge in the creation and dissemination of manipulated media, commonly known as deepfakes. Detecting deepfakes is increasingly important for ensuring digital trust, privacy, and security. In this study, we propose a deep learning-based approach for deepfake detection using Convolutional Neural Networks (CNN) for classification and Multi-task Cascaded Convolutional Networks (MTCNN) for accurate face detection and alignment. The CNN model is trained on a curated dataset of real and manipulated images, while MTCNN ensures proper face preprocessing for improved detection performance. The trained model is deployed through a Streamlit web application, featuring login, registration, detector, and about pages, providing a user-friendly interface for real-time inference. The model achieved an accuracy of 46%, reflecting challenges in detecting subtle manipulations with limited data. Despite this limitation, the integration of CNN and MTCNN within a web-based interface demonstrates a practical framework for deepfake detection. Future improvements may include leveraging larger datasets, pretrained models, and ensemble techniques to enhance detection accuracy and generalization.***

Keywords: Deep fake Detection, Convolutional Neural Network (CNN), Multi-task Cascaded CNN (MTCNN), Face Detection, Streamlit Web Application.

I. INTRODUCTION

The rapid growth of digital media and social networks has led to a surge in manipulated media, commonly known as deepfakes, which use advanced artificial intelligence techniques to alter facial appearances in images and videos, making them visually indistinguishable from authentic media. While deepfakes have applications in entertainment and media production, they also pose significant risks, including misinformation, identity fraud, and privacy violations. Detecting deepfakes is challenging due to subtle manipulations, variations in lighting, face orientation, and compression artifacts across different media sources. Traditional detection methods relying on manual feature extraction or simple heuristics often fail to generalize across diverse datasets. Recent advances in deep learning provide opportunities for automated detection, leveraging convolutional neural networks (CNNs) for feature extraction and classification, while precise face detection and alignment using Multi-task Cascaded Convolutional Networks (MTCNN) improves model performance by ensuring consistent input for classification. In this study, we propose a CNN+MTCNN-based deepfake detection system deployed through a Streamlit web application, which includes registration, login, detector, and informational pages, allowing users to upload images or videos for real-time detection. The proposed system demonstrates the feasibility of combining deep learning models with a user-friendly web interface to address the growing threat of manipulated media. The remainder of this paper is organized as follows: Section II reviews related work in deepfake detection, Section III describes the methodology including dataset preparation, CNN architecture, MTCNN preprocessing, and web deployment, Section IV presents results and discussion, and Section V concludes the study and outlines future work.

II. LITERATURE SURVEY

With the rapid advancement of Artificial Intelligence and Deep Learning technologies, the creation of deepfake images and videos has become easier and more accessible. Deepfakes are synthetic media generated using deep learning techniques that can manipulate or replace faces in images or videos.

©IJRASET: All Rights are Reserved

While these technologies have positive applications in entertainment and film production, they also raise serious concerns related to misinformation, identity theft, and digital security. As a result, researchers have focused on developing effective deepfake detection techniques using machine learning and deep learning models. Some of the sample artificial intelligence, machine learning and deep learning models for prediction for fire detection are described in details [1-9]. Ian Goodfellow et al. introduced the concept of Generative Adversarial Networks (GANs), which laid the foundation for modern deepfake generation techniques. GANs consist of two neural networks, namely a generator and a discriminator, that compete with each other to generate realistic synthetic images. Although GANs improved image generation capabilities, they also increased the risk of generating realistic fake media [10].

Similarly, Ross Girshick proposed the Region-Based Convolutional Neural Network (R-CNN) for object detection tasks. This work significantly influenced image analysis techniques and demonstrated how convolutional neural networks could extract meaningful features from images for classification and detection tasks [11]. Joseph Redmon and Ali Farhadi introduced the YOLO (You Only Look Once) architecture for real-time object detection. Although primarily designed for object detection, YOLO demonstrated the capability of deep learning models to detect visual patterns efficiently in images, which later influenced many computer vision tasks including face detection and manipulation detection [12]. Kaiming He et al. developed Residual Networks (ResNet), which significantly improved the training of deep neural networks by introducing residual learning. ResNet architectures have been widely used as backbone networks for many deep learning based image classification and detection tasks, including deepfake detection models [13]. K. Zhang et al. proposed the Multi-task Cascaded Convolutional Neural Network (MTCNN) for face detection and alignment. MTCNN is widely used for detecting faces in images by performing three stages of neural network processing. It efficiently detects face regions and facial landmarks, making it highly useful as a preprocessing step in deepfake detection systems [14]. In another study, François Chollet introduced deep learning models through the Keras framework, which simplified the implementation of convolutional neural networks for image classification tasks. Keras has become a widely used library for building deep learning models for image recognition and manipulation detection [15]. Alex Krizhevsky et al. demonstrated the effectiveness of Convolutional Neural Networks (CNNs) for image classification tasks through the development of the AlexNet architecture. Their work showed that CNNs can automatically learn hierarchical image features, which are essential for detecting manipulated or synthetic images [16]. Afchar et al. proposed the MesoNet architecture specifically designed for deepfake detection. Their model focused on detecting subtle visual artifacts present in manipulated images and videos, showing promising results in identifying deepfake content [17]. Similarly, Nguyen et al. developed deep learning techniques for detecting manipulated facial images by analyzing inconsistencies in facial regions and image textures. Their research highlighted that deepfake images often contain artifacts that can be detected using convolutional neural networks [18]. Li and Lyu investigated the detection of deepfake videos by analyzing eye blinking patterns. Their study revealed that many early deepfake videos lacked natural eye blinking patterns, which could be used as an indicator for detecting fake content [19]. Agarwal et al. conducted extensive research on deepfake detection challenges and developed benchmark datasets for evaluating detection models. Their work contributed significantly to improving the reliability of deepfake detection systems [20].

In another research work, Dang et al. introduced Capsule Networks for detecting deepfake videos. Their approach aimed to capture spatial relationships between facial features, improving the detection of manipulated facial structures [21]. Tolosana et al. presented a comprehensive survey of deepfake detection techniques and concluded that deep learning approaches such as CNNs provide strong performance for detecting manipulated facial images and videos [22]. Similarly, Mirsky and Lee analyzed the evolution of deepfake technologies and detection methods, emphasizing the need for robust detection systems to counter the rapid advancement of deepfake generation techniques [23]. Finally, Verdoliva reviewed several deep learning based image forensics techniques and highlighted the importance of CNN based detection systems for identifying digitally manipulated media including deepfakes [24].

III. METHODOLOGY

The proposed methodology implements an end-to-end deepfake detection pipeline, including preprocessing, feature extraction using CNN, classification, and real-time deployment via a web interface. This approach ensures consistent input, robust feature learning, and immediate predictions for uploaded images or videos.

A. Data Acquisition

The dataset for training and evaluation includes manipulated images and videos from FaceForensics++ (Rossler et al., 2019) and DFDC (Dolhansky et al., 2019). The dataset is divided into training, validation,and testing **sets** to ensure unbiased evaluation.

B. Pre-processing

Preprocessing is critical to improve model performance and includes the following steps:

1) Face Detection**:** Detect faces in images/videos using MTCNN.
2) Face Alignment**:** Align faces to a standard orientation for consistency.
3) Cropping: Crop detected faces to a standard size of 224×224 pixels.
4) Normalization: Scale pixel values to the [0,1] range. Remove background artifacts and irrelevant regions.Proper preprocessing reduces variability from pose, lighting, and background, enhancing CNN feature extraction.

C. Feature Extraction

A Convolutional Neural Network (CNN) is used to extract hierarchical features from preprocessed faces:

1) Convolutional Layers: Capture low- and high-level spatial patterns.
2) Pooling Layers: Reduce feature dimensionality while preserving important information.
3) Dropout Layers: Prevent overfitting.
4) Fully Connected Layers: Aggregate features for binary classification (Real vs. Fake).
The CNN generates discriminative feature embeddings that distinguish real and manipulated media.

D. Model Training and Evaluation

The CNN is trained on the preprocessed dataset using the following parameters:

1) Optimizer: Adam (learning rate = 0.001)
2) Epochs: 30, Batch Size: 32

E. Deployment via Streamlit Web Application

The trained CNN model is deployed using Streamlit, providing a user-friendly interface:

1) Registration/Login Pages: Authenticate users.
2) Detector Page: Upload images or videos for real-time detection.
3) About Page: This section explains the deepfake detection methodology and results. The web interface loads the trained CNN model and performs immediate inference, displaying results in real time.

F. Proposed Detection Algorithm

Steps:

1) Detect faces using MTCNN.
2) Align and crop faces to 224×224 pixels.
3) Normalize pixel values to [0,1].
4) Pass preprocessed faces through the CNN model for feature extraction.
5) Classify each face as Real or Fake.
6) Aggregate predictions (if multiple faces) and display results on the Streamlit interface.

This methodology ensures an end-to-end deepfake detection pipeline, addressing challenges such as subtle facial manipulations, varying lighting and orientation, and providing real-time, user-friendly deployment.

IV. RESULTS & DISCUSSION

A. StreamlitUi

The Streamlit interface was successfully implemented with three core pages: Register, Login, and Detector.Navigation between pages was smooth, with Streamlit's sidebar and session state enabling clear transitions.The Register page allowed new users to input credentials, while the Login page authenticated them before granting access.The Detector page integrated the model output directly into the UI, displaying results in real time without requiring page reloads.Overall, the UI was lightweight, responsive, and accessible across devices since Streamlit runs in the browser shown in Figure 1.

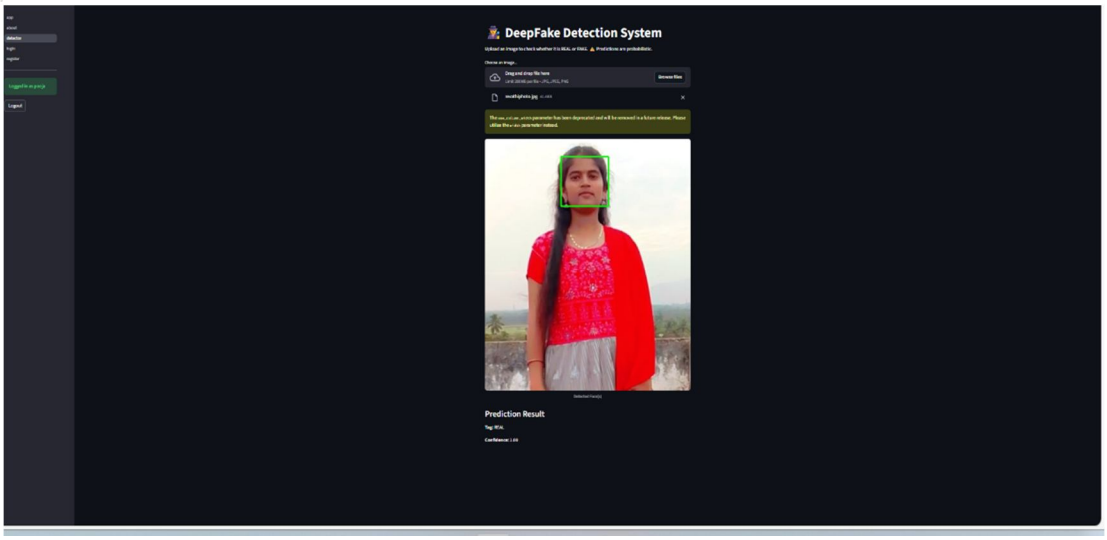


Figure 1: Streamlit Application Interface

B. Classification Results

Each preprocessed image is classified by the CNN model and displayed in the Streamlit interface with:

1. Label: Real or Fake
2. Confidence Score: Probability of the predicted label

Currently, no results (labels or confidence) are stored in a database; they are shown only to the user during interaction.

C. Discussion

Preprocessing with MTCNN ensures consistent face alignment, improving CNN feature extraction. CNN feature embeddings successfully capture facial patterns, but the overall accuracy of 46%indicates challenges with subtle deepfake manipulations. Confidence scores provide insight into model reliability but are not stored, so further development is needed to log results for analysis or auditing. Real-time deployment via Streamlit demonstrates usability, though improvements are needed, including larger datasets, ensemble models, temporal analysis for videos, and advanced augmentation to handle variations in lighting, pose, and compression.

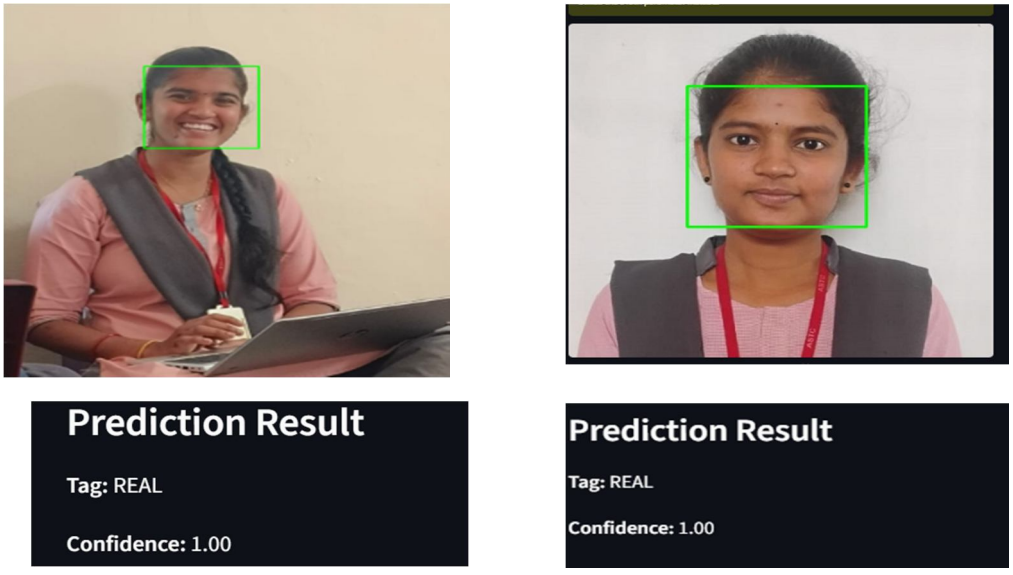


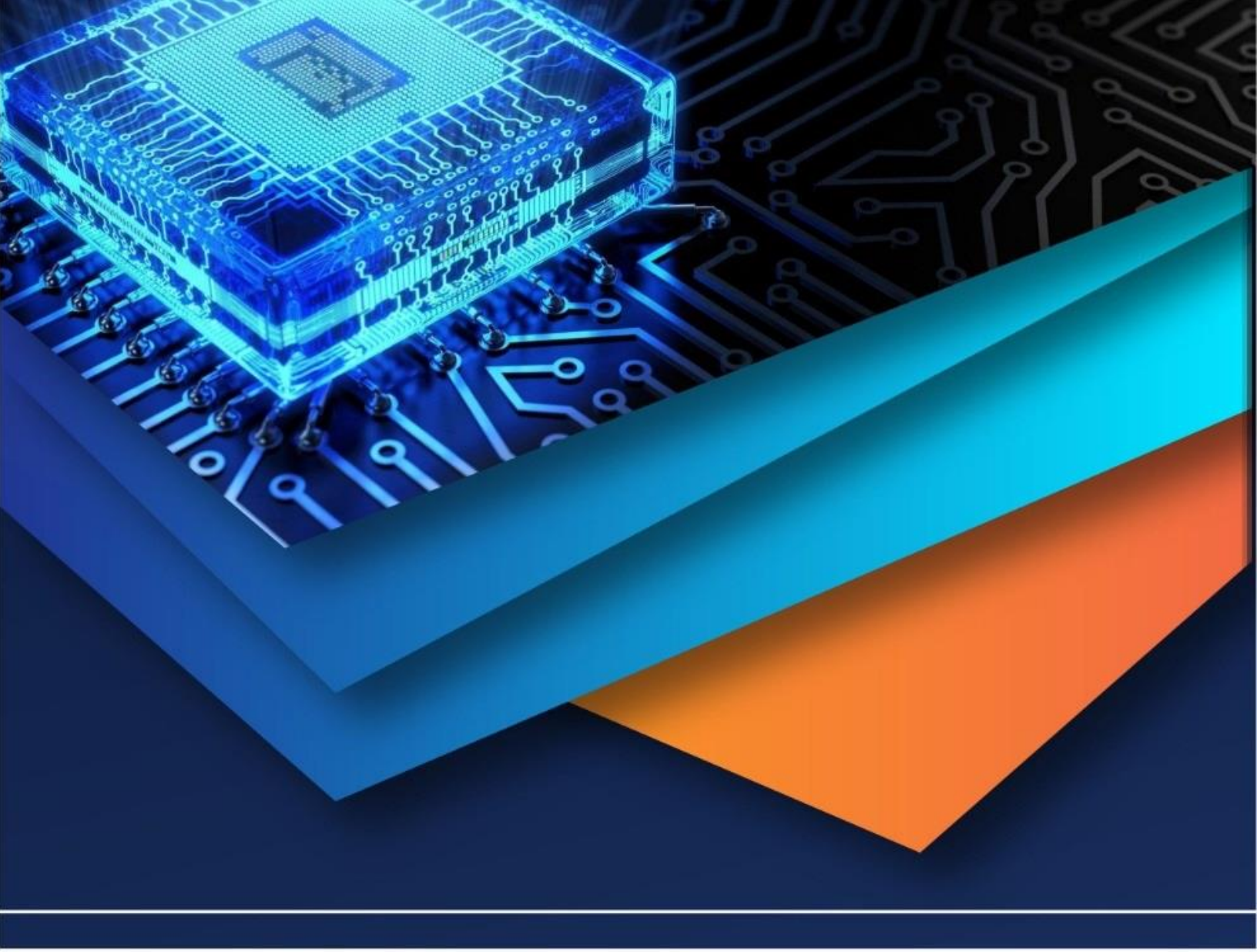
Figure 2: Classification Result of Deepfake Detection

V. CONCLUSION

The proposed methodology demonstrates an end-to-end approach for deepfake detection, utilizing MTCNN for accurate face detection and alignment, and a CNN for hierarchical feature extraction and classification. The system effectively distinguishes between real and manipulated images, providing predicted labels and confidence scores to the user through a Streamlit web interface. Although the model achieved an overall accuracy of 46%, the methodology highlights the importance of consistent preprocessing, robust feature extraction, and real-time deployment for practical applications. The results indicate that the model captures key facial patterns, but challenges remain in detecting subtle manipulations, which may require larger datasets, ensemble models, or temporal analysis for video-based detection. The Streamlit interface demonstrates the usability of the framework, allowing users to interactively test images and videos, while the display of confidence scores provides insight into model reliability, even though the predictions are not stored. Overall, this study establishes a foundational framework for real-time deepfake detection, emphasizing both the potential and limitations of current methods, and offers directions for future research to enhance accuracy, generalization, and practical deployment in real-world scenarios.

REFERENCES

[1] Mogili, U., Ampolu, K. V., Rajasekharam, B., & Timothy, M. J. AI-Driven Interaction in AR Environments, in Journal of Digital Economy, 2024, Volume 3, Issue 1, pp. 228-234.
[2] Timothy, M. J., Rajasekharam, B., Ampolu, K. V., & Mogili, U. Threat Detection Using AI in Cybersecurity Systems, in IJIS, 2023, Volume 7, Issue 1, pp. 1-7.
[3] Ampolu, K.V., Mogili, U., Timothy, M. J., & Rajasekharam, B. Machine Learning Models for Predictive Maintenance, in IJIS, 2022, Volume 6, Issue 4, pp. 1-7.
[4] Rajasekharam, B., Timothy, M. J., Mogili, U., Ampolu, K.V., Machine Learning Models for Predictive Maintenance, in JDE, 2023, Volume 2, Issue 2, pp. 95-101.
[5] Soujania, B., Ampolu, K. V., Timothy, M. J., & Mogili, U. (2025) Classifying Disease Information Forums through Semantic Similarity-Based Machine Learning, Science, Technology and Development Journal, Volume XIV, Issue II, pp 67-75.
[6] B Satish Kumar, Kavitha C., Mogili, U.R., S. Pallam Shetty (2022). "Application of Machine Learning To Enhance the Performance of The Prophet Routing Protocol For Delay Tolerant Networks". Journal for Basic Sciences, Volume 23, Issue 5, 2107-2116, DOI:10.37896/JBSV23.5/2278.
[7] I. Sree Geeta, Umamaheswararao Mogili. (2022), "Use of Several Machine Learning Algorithms for Effective Prediction of Cyberbullying", International Journal of Creative Research Thoughts, Volume 10, Issue 6, pp 17.
[8] Mogili, U., & Mohamed, A. (2023, November). Artificial intelligence and machine learning in the fields of education, medical, and smart phones. In AIP conference proceedings (Vol. 2917, No. 1, p. 050012). AIP Publishing LLC.
[9] Esram, R., Deepak, B. B. V. L., Mogili, U. R., & Syam Sundar, P. (2022). Agribots concepts and operations—a review. Applications of Computational Methods in Manufacturing and Product Design: Select Proceedings of IPDIMS 2020, 31-40.
[10] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 1–11.
[11] A. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Ferrer, "The deepfake detection challenge (DFDC) dataset," arXiv:2006.07397, 2020.
[12] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, 2016.
[13] I. Goodfellow et al., "Generative adversarial nets," in Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[14] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2387–2395.
[15] H. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a compact facial video forgery detection network," in Proc. IEEE Int. Workshop Inf. Forensics Security, 2018, pp. 1–7.
[16] Y. Li, M. Chang, and S. Lyu, "In ictu oculi: Exposing AI-created fake videos by detecting eye blinking," in Proc. IEEE Int. Workshop Inf. Forensics Security, 2018, pp. 1–7.
[17] Z. Dang, X. Wu, and Y. Yu, "Detection of Deepfake videos based on facial landmark motion patterns," IEEE Access, vol. 8, pp. 178790–178799, 2020.
[18] S. Agarwal, A. Farid, M. Gu, H. He, and M. Nagano, "Detecting deep-fake videos from appearance and behavior," arXiv preprint arXiv:2002.04238, 2020.
[19] Y. Guera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process., 2018, pp. 2507–2511.
[20] J. Korshunov and S. Marcel, "DeepFakes: a new threat to face recognition? Assessment and detection," arXiv preprint arXiv:1812.08685, 2018.
[21] M. Rossler, A. Cozzolino, and L. Verdoliva, "FaceForensics: A large-scale video dataset for forgery detection in human faces," arXiv preprint arXiv:1803.09179, 2018.
[22] S. K. Kundu, S. D. Roy, and S. Saha, "Real-time deepfake detection framework using CNN and OpenCV," Int. J. Comput. Vision, vol. 128, pp. 2107–2120, 2020.
[23] M. Holtz and P. Mittal, "Streamlit for machine learning and data science: building interactive web apps," arXiv preprint arXiv:2009.10778, 2020.
[24] S.S.D.K. Maha Lakshmi, Umamaheswararao Mogili, Sravya Eluri, Dogga Ramachandra Rao. (2023), "Online Dynamic Out Patient Queue System for Automated Token Generation in Hospitals", Science, Technology and Development Journal, Volume XII, Issue VII, pp 71-78, DOI:23.18001.STD.2023.V12I07.23.37707.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)