MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 890-892

Detection of Deepfakes with the help of Machine learning

[1]Gokila Kannan, [2]Sivakumar Dhandapani [3,4,5] Kolle sagar, Manam Trinadh,Shaik Muhammad Abubakar Siddiq, and [6] Jothimani Ponnusamy
[1]Assistant Professor
Department of Computer Science and Engineering,
Academy of Maritime Education and Training (AMET) Deemed to be University,
135, East Coast Road, Kanathur, Chennai – 603112, Tamil Nadu, India
[2] Professor, Department of CSE-Cyber Security,
Rajalakshmi Engineering College, Thandalam, Chennai - 602 105, Tamil Nadu, India
[3,4,5]. Undergraduate Students and [6] Professor of Practice,
Department of Computer Science and Engineering,
Academy of Maritime Education and Training (AMET) Deemed to be University,
135, East Coast Road, Kanathur, Chennai – 603112, Tamil Nadu, India
gokilakannan17@gmail.com, sivakumar.d@rajalakshmi. edu. in
, kollesagar4@gmail.com and jothi58@gmail.com

Abstract
Deepfake technology has advanced rapidly in recent years due to improvements in artificial intelligence. Today, it is possible to create highly realistic fake videos and images that are often difficult to distinguish from real ones. While this technology has useful applications in areas like entertainment, it also raises serious concerns such as misinformation, identity theft, and threats to digital security.Detecting deepfakes has become increasingly challenging, especially when models are trained on limited datasets and struggle to perform well on new types of manipulations. In this work, we propose a deepfake detection system that combines Convolutional Neural Networks (CNNs), Vision Transformers (ViT), and meta-learning techniques to improve both accuracy and adaptability. The system is designed as a modular pipeline and includes a user-friendly web interface built using FastAPI and React. By focusing on better generalization and improved learning strategies, the model performs effectively across datasets like FaceForensics++ and CelebDF-v2. Overall, the proposed system offers a practical and efficient solution for real-time deepfake detection.
Keywords: CNNs,Visition transformers, Face Forensics,Machine learning and Deepfakes

1. Introduction

Artificial intelligence has made it easier than ever to generate realistic fake images and videos, commonly known as deepfakes. These are created using advanced methods such as Generative Adversarial Networks (GANs), which can replicate human faces, expressions, and even speech with impressive accuracy.
Although deepfakes have useful applications, they also pose serious risks. They can be used to spread misinformation, manipulate public opinion, or commit fraud. Because of this, detecting deepfakes has become an important area of research.
However, detecting deepfakes is not simple. Modern techniques are designed to hide visible flaws, making them difficult to identify using traditional methods. Many existing approaches focus only on analyzing individual frames, which often fails to capture subtle manipulations.
To address these challenges, this work combines multiple techniques:

- CNNs for capturing fine visual details
- Vision Transformers for understanding global patterns
- Meta-learning to improve performance on unseen data

The aim is to develop a system that is accurate, adaptable, and easy to use in real-world scenarios.

2. Literature Background

Deepfake detection has received significant attention in recent years. Researchers have explored various approaches to improve accuracy and ensure models work well across different types of manipulations. This section highlights key contributions in scalability, spatio-temporal analysis, and generalization.

2.1 Discrepancy-Based Learning for Scalable Detection

The study on discrepancy-based learning focuses on improving the scalability of deepfake detection systems. Traditional models often fail when trained on one type of manipulation and tested on others.
To solve this, the proposed method uses a dual-branch architecture. One branch processes the original image, while the other processes a modified version of the same image. A self-attention mechanism is then used to identify common patterns between them.
This approach helps the model focus on general forgery characteristics rather than specific dataset patterns. As a result, it performs better across different datasets and remains robust even when the input is slightly distorted.

2.2 Spatio-Temporal Attention-Based Detection

Another important approach focuses on combining spatial and temporal information. Many earlier systems only analyze images frame by frame and miss important motion-based inconsistencies.
This method introduces attention mechanisms that analyze both image details and motion patterns. Spatial attention detects small visual changes within frames, while temporal attention identifies unusual movements across frames.
By combining these features, the model can detect even subtle deepfake manipulations more effectively and with better consistency.

2.3 Meta-Learning with Vision Transformers (MEViT)

A major challenge in deepfake detection is the inability of models to handle new and unseen manipulation techniques. The MEViT approach addresses this by combining EfficientNet, Vision Transformers, and meta-learning.
The model learns not just from data but also how to adapt to new data. Additional techniques like Pair-Discrimination Loss and Domain Adjustment Loss help improve the separation between real and fake data and reduce differences between datasets.
This results in better performance, especially when tested on unfamiliar datasets.

2.4 Summary of Literature

Overall, modern approaches are moving beyond traditional CNN-based methods. Techniques like attention mechanisms, transformers, and meta-learning have significantly improved performance. However, combining these methods into a single practical system remains a challenge, which this work aims to address.

3. System Analysis

3.1 Existing System

Most traditional deepfake detection systems rely on CNN models trained on specific datasets. While they perform well on familiar data, they struggle when exposed to new types of deepfakes.
Some common limitations include:

- Poor performance on unseen datasets
- Weak preprocessing, especially in face detection
- Lack of real-time capability
- Absence of user-friendly interfaces
- Limited explanation of results

3.2 Proposed System

To overcome these limitations, a web-based deepfake detection system is proposed. It combines multiple technologies such as React for the frontend, FastAPI for the backend, OpenCV for face detection, and TensorFlow for model inference.
The system is designed to be efficient, accurate, and easy to use. It supports real-time analysis and provides clear outputs through a structured interface.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 890-892

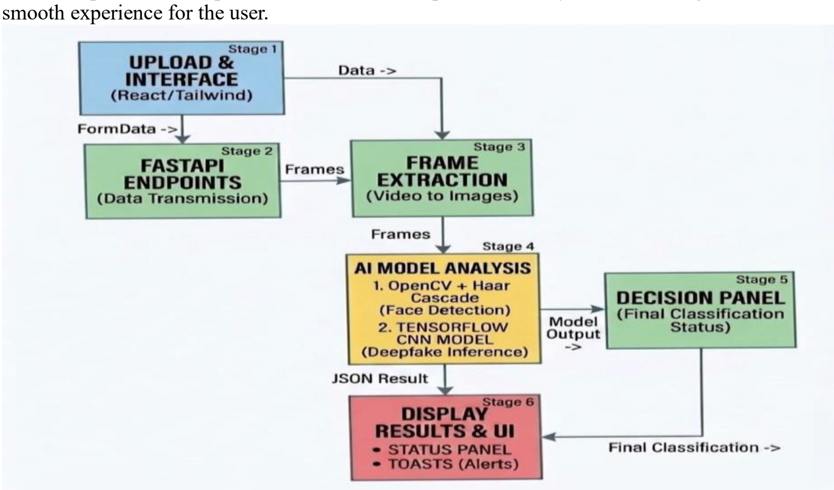
Vision Transformer Integration
The system uses Vision Transformers to capture global relationships in images. This helps in identifying deeper inconsistencies that traditional CNNs might miss.
Meta-Learning for Generalization
Meta-learning is used to improve the system's ability to handle different datasets. It allows the model to adapt to new manipulation techniques more effectively.
Advantages

- Higher detection accuracy
- Better generalization
- Real-time processing
- User-friendly interface
- Clear and interpretable results

3.3 System Architecture
The system follows a modular design that includes frontend, backend, and machine learning components. The workflow involves data input, preprocessing, feature extraction, model prediction, and result display.
3.4 Working Modules

- Data Training: Involves dataset preparation, preprocessing, and model training using standard datasets.
- Feature Extraction & Testing: Extracts frames, detects faces, and performs predictions.
- Frontend Design: Provides a clean interface for uploading files and viewing results.
- Backend Development: Handles processing, model inference, and communication.
- API Integration: Ensures smooth data flow between system components.

4. Requirement Specification
4.1 Hardware Requirements
The system requires a reliable computing setup for efficient processing. A basic system with a multi-core processor, 8 GB RAM, and SSD storage is sufficient for testing. However, higher configurations with more RAM and a dedicated GPU are recommended for real-time performance.
GPUs significantly speed up deep learning operations, making them important for large-scale or real-time applications. Additional components such as cameras, storage devices, and network infrastructure further support system functionality.
4.2 Software Requirements
The system is developed using a combination of modern software tools. Python is used for backend processing and model execution, while React and Tailwind CSS are used for building the frontend interface.
FastAPI handles backend communication, and TensorFlow is used for deep learning tasks. OpenCV is used for image processing and face detection. Additional tools and libraries support development, testing, and deployment.
5. Implementation
5.1 Pre-Processing and Face Extraction
Before performing deepfake detection, the input image or video needs to be prepared properly. This step ensures that the data is clean, consistent, and suitable for the model to process.
In this system, OpenCV along with the Haar Cascade classifier is used to detect faces efficiently. Instead of processing every single frame (which can be time-consuming), the system selects frames at regular intervals to reduce computation while still maintaining accuracy.
Once a face is detected, it is cropped and resized to a fixed size of 224 × 224 pixels, which matches the input requirements of the model. In addition, pixel values are normalized to a range between 0 and 1. This helps the model perform more stable and reliable predictions.
Overall, this step plays a crucial role in improving both the speed and accuracy of the system.
5.2 Backend API Implementation
The backend of the system is built using FastAPI, which acts as the connection between the user interface and the deep learning model.
When a user uploads an image or video, the backend receives the file, processes it, and sends it through the detection pipeline. After analysis, it returns the result in a simple JSON format, including whether the content is real or fake along with a confidence score.
FastAPI is particularly useful because it can handle multiple requests at the same time. This makes the system faster and suitable for real-time applications.
5.3 Frontend Interface Implementation
The frontend is designed using HTML, CSS, and JavaScript to provide a clean and easy-to-use interface.
Users can upload either an image or a video, preview it before processing, and then view the detection result. A toggle option is provided to switch between image and video modes, making the interface flexible.
The results are displayed clearly along with confidence values, and previous detections are saved in a history section. Color-coded outputs make it easier for users to quickly understand whether the content is real or fake.
5.4 Client-Side Scripting and Interaction
JavaScript is used to handle all user interactions on the frontend and to communicate with the backend.
It manages file uploads, shows previews, and sends the data to the backend for processing. Once the response is received, the interface is updated instantly with the result and confidence score.
It also keeps a record of past detections, which improves usability. Error handling is included to deal with issues like invalid files or server problems, ensuring a smooth experience for the user.



5.5 System Integration
All components of the system—preprocessing, model inference, backend processing, and frontend interaction—are combined into a single workflow.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 890-892

The process starts with user input, followed by data preparation, prediction, and finally displaying the result. This modular design makes the system flexible and easy to upgrade in the future.
It also ensures efficient data flow and quick response times, making the system reliable for real-time deepfake detection.

6. Experimental Results and Performance Analysis
6.1 Dataset Description
To evaluate the system, two well-known datasets were used.
The first dataset, FaceForensics++, contains a large number of videos that have been manipulated using different deepfake techniques. It provides a good base for training and testing.
The second dataset, Celeb-DF (v2), is more challenging because it contains high-quality and realistic deepfake videos of celebrities. This dataset helps test how well the model performs on more advanced manipulations.

6.2 Performance Evaluation
The system's performance is measured using standard metrics such as accuracy and AUC.
When compared with other models, the proposed approach shows clear improvement:

- EfficientNet-B0: 87.4% accuracy
- Vision Transformer (ViT): 91.2% accuracy
- Proposed model (ViT + Meta-learning): 94.8% accuracy

These results show that combining Vision Transformers with meta-learning significantly improves performance, especially when dealing with new or unseen data.

6.3 Explainability and Transparency
Understanding how the model makes decisions is important, especially in sensitive applications.
In future versions, Explainable AI techniques like Grad-CAM will be added. These methods highlight specific regions in an image (such as the eyes or mouth) that influenced the model's decision.
This makes the system more transparent and trustworthy, especially for applications like digital forensics or security.

6.4 Temporal Analysis Enhancement
Currently, the system focuses mainly on analyzing individual frames. However, deepfake videos often show inconsistencies across frames, such as flickering or unnatural movements.
To improve this, future updates may include models like LSTM or 3D CNNs, which can analyze sequences of frames and detect motion-based irregularities.

7. Conclusion
This project demonstrates a practical and effective approach to detecting deepfake images and videos using machine learning.
With the growing use of AI-generated media, such systems are becoming essential for maintaining trust and preventing misuse. The proposed system combines computer vision and deep learning techniques within a modular and user-friendly framework.
Key achievements include:

- A web-based platform for easy media upload and analysis
- Efficient frame extraction and preprocessing
- Accurate deepfake classification
- Clear and structured output with confidence scores
- Fast and responsive backend processing

Overall, the system shows how machine learning can be applied to solve real-world problems related to digital security.

8. Future Scope
While the current system performs well, there are many opportunities for improvement.
Future work can include integrating more advanced models like Vision Transformers and 3D CNNs to further improve accuracy. Real-time detection can be enhanced using GPU optimization, allowing the system to analyze live video streams.
The system can also be expanded to include audio analysis and lip-sync detection, making it capable of identifying more complex deepfakes. Deployment on cloud platforms and mobile applications can make the system more accessible to a wider audience.
Additionally, using larger and more diverse datasets will help the model stay effective against new types of deepfake techniques. Incorporating explainable AI will further improve user trust and transparency.

REFERENCES

[1] VAN-NHAN TRAN SUK-HWAN LEE - Generalization of Deepfake Detection With Meta-Learning EfficientNet Vision Transformer, 17 March 2025
[2] Yongqi Yang1- D3: Scaling Up Deepfake Detection by Learning from Discrepancy, 23 Mar 2025
[3] Yunzhuo Chen- Deepfake Detection with Spatio-Temporal Consistency and Attention, 12 Feb 2025
[4] Chenqi Kong - Enhancing General Face Forgery Detection via Vision Transformer with Low-Rank Adaptation[5] Chuangchuang Tan1- Frequency-Aware Deepfake Detection: Improving Generalizability through Frequency Space LearningFranc¸ois Chollet. Xception: Deep learning with depthwise separableconvolutions. In Proceedings of the IEEE conference on computer visionand pattern recognition, pages 1251–1258, 2017.
[6] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes,Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. arXiv preprint arXiv:2006.07397, 2020.
[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deepresidual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June2016.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deepresidual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778,2016.
[9] Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory.Neural Computation, 9(8):1735–1780, 1997.
[10] Berthold KP Horn and Brian G Schunck. Determining optical flow. Artificial intelligence, 17(1-3):185–203, 1981.
11 Asokan and Sivakumar, Fault detection and diagnosis for three-tank system using robust residual generator(2009),Indian journal of science and Technology, Vol.2 No. 7,PP:23-29
12. Gokila and Sivakumar, HYBRID DEEP LEARNING TECHNIQUES FOR ENHANCED ASPECT BASED SENTIMENT ANALYSIS AND CONTEXTUAL FEATURE OPTIMIZATION(2023), Volume 38 No. 11s,PP:1605-1629