International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04910
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

# Deepfake Video Detection Using CNN and Flask

**CHINTHAGINGALA VASUNDHARA, MOONA NIVEDITHADEV.**
Assistant professor, MCA FINAL SEMESTER, Master of Computer Applications,
Sanketika Vidya Parishad Engineering College, Visakhapatnam, India.

**ABSTRACT**:

In recent years, the rise of deepfake technology has posed a significant threat to the authenticity of digital media, enabling the creation of highly realistic yet manipulated videos that can deceive, misinform, and harm reputations. This project proposes a deepfake video detection system that utilizes Convolutional Neural Networks (CNN) for accurate frame-level classification, integrated within a Flask-based web application for user-friendly interaction. The system accepts video uploads via a web interface, extracts key frames using OpenCV, and classifies them as real or fake using a pre-trained CNN model trained on deepfake datasets. The final verdict is generated through aggregated predictions across frames, providing users with a reliable assessment of video authenticity. This solution offers an efficient, scalable, and accessible tool for combating deepfake content and enhancing digital media forensics.

**INDEX TERMS:** Deepfake detection, CNN, Flask, Video forensics, OpenCV, Fake video identification, Digital media authentication, Deep learning.

## 1.INTRODUCTION

The rapid advancement of artificial intelligence has led to the widespread creation of deepfakes manipulated videos generated using deep learning techniques that convincingly replace a person's face or voice with another's. These videos [5], often created using Generative Adversarial Networks (GANs), pose significant ethical, social, and security concerns, as they are increasingly used for misinformation [15], fraud, and digital impersonation. Detecting such manipulations has become a critical challenge due to the high level of realism achieved by modern deepfake generation techniques. In this study, we propose a deepfake video detection system that leverages Convolutional Neural Networks (CNNs) for frame-level analysis of video content [6]. The CNN model is trained to detect subtle artifacts and inconsistencies that differentiate real videos from fake ones. To make this system accessible and user-friendly, it is integrated into a web-based application using the Flask framework, allowing users to upload videos and receive real-time detection results [10]. This research aims to provide a practical and effective solution for deepfake detection, contributing to the broader efforts in digital media forensics and content verification.

## 1.1 EXISTING SYSTEM

The existing systems for deepfake video detection rely heavily on traditional Convolutional Neural Networks (CNNs) and pre-trained deep learning models such as VGGNet, Exception Net, and Rest Net, which analyze facial features to identify manipulation artifacts in videos. These systems are often integrated with large datasets like Face Forensics++ [3], DFDC, or Celeb-DF to train models on real vs. fake facial imagery. Most detection processes involve extracting video frames, preprocessing them, feeding them into CNN models for classification, and finally outputting the prediction results [13]. However, many of these systems are limited to offline desktop use, lack real-time web-based deployment, and provide minimal user interactivity. Additionally, without an easy-to-use interface or lightweight framework like Flask, these models are often difficult to integrate into web platforms for practical applications, especially for users with limited technical knowledge.

## 2. LITERATURE SURVEY

The rise of deepfake videos has spurred significant research into automated detection techniques [5], particularly those using deep learning. Convolutional Neural Networks (CNNs) have been widely applied due to their strong ability to extract spatial features from images and video frames. Afchar et al. (2018) introduced MesoNet, a shallow CNN architecture designed for detecting facial forgeries based on mesoscopic features [13]. Rossler et al. (2019) developed the FaceForensics++ dataset and evaluated multiple CNN models,

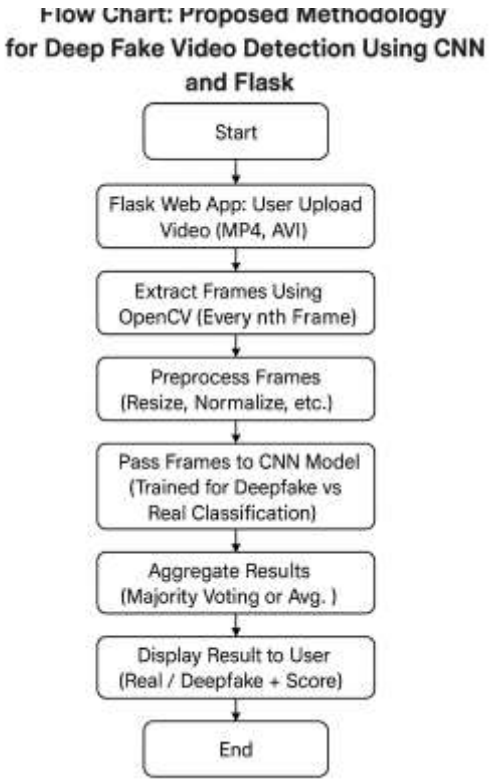
demonstrating their effectiveness in detecting subtle artifacts in manipulated videos. Further advancements have integrated temporal information to improve accuracy across video sequences (Nguyen et al., 2019), though such models often demand more computational resources. Other studies have focused on physiological inconsistencies such as eye blinking (Li & Lyu, 2019), which are often missed by GAN-based generators [10]. Despite high detection accuracy, many of these models remain limited to offline use. Recently, lightweight CNN models have been integrated into web applications using Python-based frameworks like Flask for real-time, user-friendly deployment. These systems allow users to upload videos and receive instant authenticity assessments but are often constrained by performance on diverse datasets [1]. Our research builds upon these foundations by combining a CNN-based detection model with a Flask web interface, aiming to create an efficient, accessible, and real-time tool for identifying deepfake content.

## 3. CHALLENGES

- **High Realism of Deepfakes:** Modern deepfake generation techniques produce highly realistic facial movements [6], expressions, and voice synchronization, making it increasingly difficult for CNN models to distinguish between real and fake content.
- **Limited Dataset Generalization:** CNN models trained on one dataset (e.g., FaceForensics++) often perform poorly on videos from different sources due to variations in resolution, lighting, compression, and generation methods [11].
- **Frame-Level Prediction Inconsistency:** Since detection is done frame-by-frame, inconsistent results across frames can lead to inaccurate overall predictions, especially if only a few frames are manipulated.
- **Computational Overhead:** Processing large video files, extracting frames, and running them through CNN models is resource-intensive and time-consuming, especially on machines without GPUs [12].
- **Lack of Audio-Visual Fusion:** The system focuses only on visual cues. Ignoring audio inconsistencies (like lip-sync issues) limits the model's ability to detect multimodal deepfakes.
- **No Real-Time Detection:** The Flask-based system processes uploaded videos offline. Real-time detection or streaming video analysis is currently not supported [9], limiting real-world applicability.
- **User Interface Limitations:** While Flask provides a basic web interface, it lacks advanced user features such as video previews, frame-wise analysis, and interactive feedback.
- **Security and Privacy Concerns:** Uploading videos for analysis raises concerns about user data privacy [5], especially when sensitive or personal media is involved.

## 4. PROPOSED METHODOLOGY

The proposed methodology involves building an automated deepfake video detection system using Convolutional Neural Networks (CNN) and deploying it through a Flask-based web application. Initially [8], real and fake video datasets are collected and preprocessed by extracting individual frames using OpenCV, followed by face detection and cropping using MTCNN or Dlib. These face images are then resized and normalized to serve as input to a CNN model, which may be a custom architecture or a fine-tuned pretrained model like Exception or VGG16. The CNN is trained to classify each frame as real or fake based on learned facial features [14]. After training, the model is integrated with a Flask web server [13], where users can upload videos, and the system processes them in real time to display detection results. The methodology ensures high detection accuracy, efficient video handling, and a user-friendly interface for practical applications in identifying manipulated media content.




## 5. ADVANTAGES.

**5.1. High Accuracy in Detection (Using CNN) –** CNNs excel at analyzing visual patterns and features in images and videos [3]. They can learn subtle differences between real and manipulated videos that are hard for humans to spot. CNNs can detect facial inconsistencies, unnatural movements, or artifacts caused by deepfake generation techniques. This leads to robust and reliable deepfake detection.

**5.2. Automated and Scalable Solution** CNN-based models automate the detection process without manual inspection. Once trained [7], the system can handle large volumes of video data quickly and efficiently. This scalability is essential for platforms that deal with massive user-generated content.

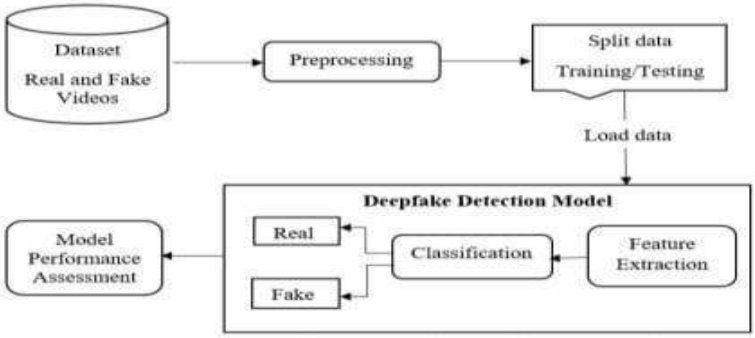
**5.3. Real-Time or Near Real-Time Detection –** With optimized CNN architectures and modern GPUs [10], detection can be performed in real-time or near real-time.

## 6. DEEP LEARNING ALGORITHM

The core deep learning algorithm used for deepfake video detection is the Convolutional Neural Network (CNN), which is highly effective in capturing spatial hierarchies in facial features. Models such as XceptionNet, ResNet, and VGG16 are commonly implemented due to their strong performance in image classification tasks [6]. These models are trained on labeled datasets of real and fake videos to detect inconsistencies like pixel-level artifacts, unnatural blinking, or distorted facial movements. The CNN learns patterns from input video frames (converted into images) [10], processes them through multiple convolutional and pooling layers, and classifies them as real or fake [11]. Flask is used to build the web interface where users can upload videos, and the trained model performs prediction and returns the result in real-time.

## 7.ARCHITECTURE

The architecture of the Deepfake Video Detection system using CNN and Flask consists of a web-based interface developed with Flask that allows users to upload video files [15]. Once a video is uploaded, OpenCV is used to extract individual frames, and face detection is performed using MTCNN or Dlib. These extracted face regions are preprocessed and fed into a Convolutional Neural Network (CNN) [3], such as XceptionNet or VGG16, which has been trained to distinguish between real and manipulated (deepfake) facial features. The CNN processes each frame to generate predictions, and the final result is determined through aggregation (e.g., majority voting). This result whether the video is real or fake is then displayed back to the user via the Flask interface, providing an end-to-end deepfake detection pipeline [14].




## 8. METHODS

The Deepfake Video Detection using CNN and Flask method involves collecting a dataset of real and fake videos [13], extracting individual frames using OpenCV, and detecting and cropping facial regions for preprocessing. These facial images are resized and normalized before being fed into a Convolutional Neural Network (CNN) [6], which may be a custom-built model or a pretrained architecture like VGG16 or Exception. CNN is trained to classify images as real or fake using binary cross entropy loss and an optimizer like Adam. Once trained, the model is integrated into a Flask web application that allows users to upload videos, extracts frame, analyzes them using the CNN [1], and presents the final prediction indicating whether the video is deepfake or authentic along with the confidence level. The system is evaluated using accuracy, precision, recall, and ROC-AUC, and can be deployed locally or on cloud platforms for real-time detection.

## 9. TOOLS

The Deepfake Video Detection system using CNN and Flask utilizes several key tools for its development and deployment [6]. Python serves as the core programming language, while Flask is used to build the web-based interface for uploading and displaying results. OpenCV is employed to extract frames from the uploaded video, and face detection is carried out using MTCNN or Dlib. The core deep learning model is developed using TensorFlow or Keras, often leveraging pre-trained CNN architectures like XceptionNet or VGG16 [12]. Supporting libraries like NumPy and Pandas handle data processing, while Matplotlib or Seaborn assist in visualizing training metrics. Frontend development includes HTML, CSS, and optionally JavaScript or Bootstrap for styling. Tools like Jupiter Notebook or VS Code are used for development, with GitHub for version control. Optionally [8], MySQL or SQLite can be integrated for data storage, and tools like Anaconda manage project environments.

## 10. TESTING

The testing phase of the Deepfake Video Detection system involves evaluating the performance of the trained CNN model on unseen video data to assess its real-world effectiveness. A separate test set [2], not used during training or validation, is utilized to extract frames and feed them into the model. The predictions for each frame are aggregated to determine the overall classification of the video as real or fake. Key performance metrics such as accuracy [8], precision, recall, F1-score, and AUC-ROC are calculated to measure how well the model generalizes. Additionally [15], confusion matrices are used to visualize the distribution of correct and incorrect predictions. The Flask application is also tested to ensure smooth video upload, frame processing, and correct integration with the CNN for delivering accurate and fast results on the front end.

## 11. OUTPUTS

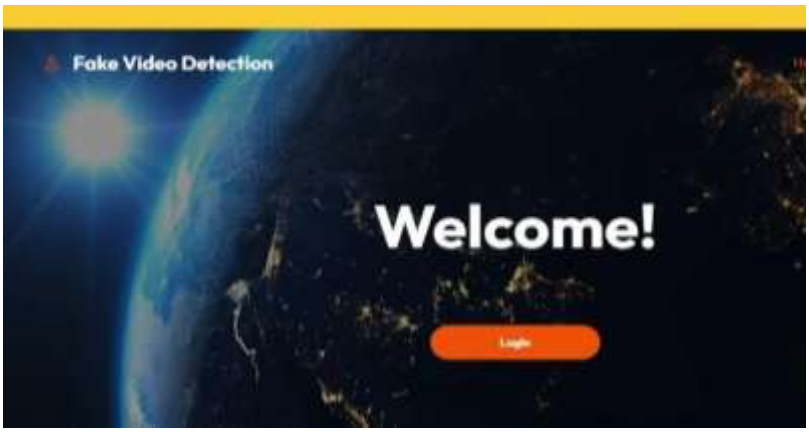



Figure 10.1: Home Page

Once the user logs in, they can access the deep fake detection features, such as uploading and viewing results. The login feature allows user authentication and authorization.


Figure 10.2: About Page

This page provides information about deepfake detection and methodology we have used.

**Registration And Login Page**

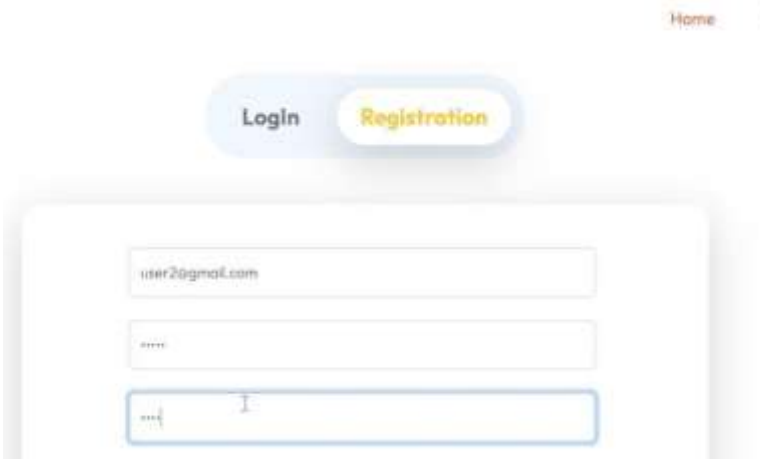



Figure 10.3: Registration Page


Figure 10.4: Login Page

The registration and login page allows users to create a new account, while the login page allows existing users to access their account.

**Upload Page:**

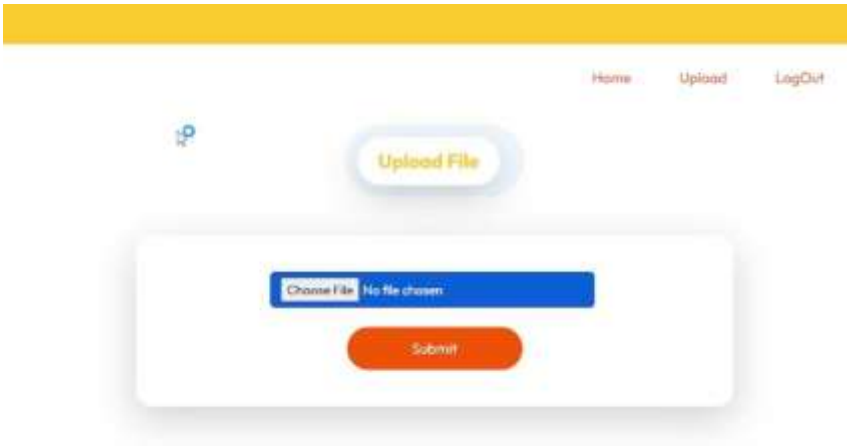



Figure 10.5: Upload Page

Once the video is uploaded, the system will analyze it and provide a classification result indicating whether the video is real or fake.

**Result Page:**

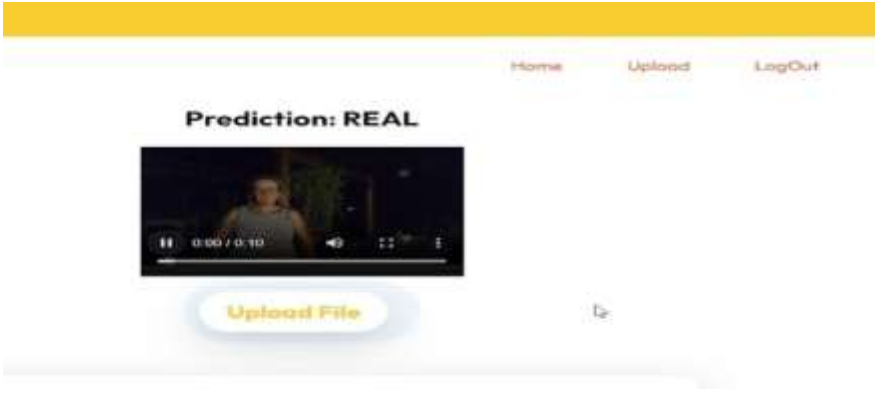



Figure 10.6: Real Video Output

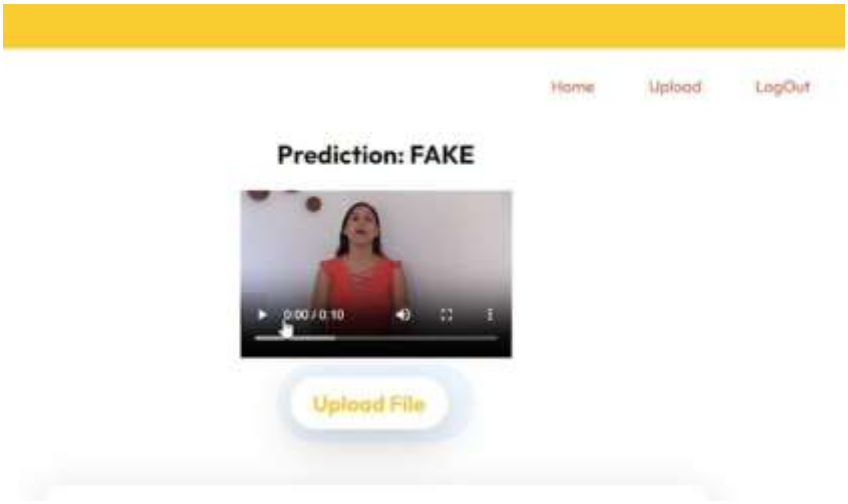



Figure 10.7: Fake Video

## 12. CONCLUSION

In conclusion, the Deepfake Video Detection system using CNN and Flask provides an effective and efficient approach to identifying manipulated videos by analyzing facial features through deep learning. By leveraging a Convolutional Neural Network trained on a well-curated dataset of real and fake videos, the system accurately classifies video content based on frame-by-frame analysis. The integration with a Flask-based web interface allows for user-friendly video uploads and real-time detection results, making it practical for real-world applications. The project demonstrates promising results in terms of accuracy and reliability, and it can be further enhanced by incorporating advanced models, larger datasets, and real-time video stream analysis for more robust deepfake detection in various domains like media, law enforcement, and cybersecurity.

## 13. FUTURE SCOPE

The future scope of the Deepfake Video Detection system includes enhancing detection accuracy by integrating advanced deep learning models such as transformers or spatiotemporal CNNs that can analyze both facial features and motion patterns across frames. Real-time detection in live video streams, including integration with surveillance and video conferencing platforms, can make the system more practical and responsive. Additionally, extending the model to detect other forms of media manipulation such as voice deepfakes, full-body swaps, and AI-generated avatars can broaden its application. Incorporating blockchain or digital watermarking for source verification and improving robustness against adversarial attacks will further strengthen security. Finally, deploying the system on mobile and edge devices can expand accessibility and enable real-time deepfake detection in low-resource environments.

## 14. ACKNOWLEDGEMENT

Chinthagingala Vasundhara working as an Assistant professor in master of computer application in sanketika vidya parishad engineering college, Visakhapatnam Andhra Pradesh. With 2 years of experience in computer science and engineering (CSE), accredited by NAAC. with her area of interest in java full stack. She is dedicated and emerging academician in the field of Computer Science, currently serving as a faculty member. With a strong foundation in technical concepts and a passion for teaching, she has begun his academic journey by actively mentoring students in practical and innovative projects. As a faculty, she has already made a significant impact by guiding student teams through their academic projects with clarity, enthusiasm, and

technical proficiency. His commitment to student success and interest in research-driven teaching make him a promising contributor to academic excellence.

Moona Niveditha Dev is pursuing her final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning Moona Niveditha Dev has taken up his PG project on DEEPFAKE VIDEO DETECTION USING CNN AND FLASK and published the paper in connection to the project under the guidance of Chinthagingala Vasundhara, Assistant Professor in SVPEC.

## 15.REFERENCES

[1] A. Rossler et al., "FaceForensics++: Learning to detect manipulated facial images," https://arxiv.org/abs/1901.08971
[2] B. Dolhansky et al., "The Deepfake Detection Challenge (DFDC) dataset," https://arxiv.org/abs/2006.07397
[3] Y. Li et al., "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," https://arxiv.org/abs/1909.12962
[4] F. Chollet, "Exception: Deep learning with depthwise separable convolutions," https://arxiv.org/abs/1610.02357
[5] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," https://arxiv.org/abs/1412.6980
[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," https://arxiv.org/abs/1409.1556
[7] K. He et al., "Deep residual learning for image recognition," https://arxiv.org/abs/1512.03385
[8] I. Goodfellow et al., "Generative adversarial nets," https://arxiv.org/abs/1406.2661
[9] A. Afchar et al., "MesoNet: A compact facial video forgery detection network," https://arxiv.org/abs/1809.00888
[10] P. Korshunov and S. Marcel, "Deepfakes: A new threat to face recognition?" https://arxiv.org/abs/1812.08685
[11] R. Tolosana et al., "Deep Fakes and beyond: A survey of face manipulation and fake detection," https://doi.org/10.1016/j.inffus.2020.07.007
[12] Z. Li et al., "A survey of deepfake detection techniques," https://doi.org/10.1109/ACCESS.2021.3053441
[13] Y. Nirkin et al., "FSGAN: Subject agnostic face swapping and reenactment," https://arxiv.org/abs/1908.05932
[14] OpenCV, "Open Source Computer Vision Library," https://opencv.org/
[15] TensorFlow, "TensorFlow Machine Learning Framework," https://www.tensorflow.org/