International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0339
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2363 - 2368

DeepFake Detection System Using Deep Learning Techniques

Sayali Dolas[1], Tanushri Kharkar[2], Neha Thakur[3], Darshana Kolte[4]
[1]Assistant professor, Dept. of CSE, Dr. D.Y.Patil Inst. of Engg. Mang. & Res., Pune, Maharashtra, India
[2,3,4]UG Scholar, Dept. of CSE, Dr. D.Y.Patil Inst. of Engg. Mang. & Res., Pune, Maharashtra, India
Emails: *sayalidolas5193@gmail.com[1], tanushrikharkar1302@gmail.com[2], thakurneha3118@gmail.com[3], darshana.kolte26@gmail.com[4]*

Abstract
Deepfake technology uses advanced artificial intelligence techniques to generate highly realistic but manipulated media content, creating serious concerns regarding authenticity and trust in digital platforms. Due to the rapid growth of social media, the spread of fake images and videos has increased significantly, leading to issues such as misinformation, identity theft, and cybersecurity threats. This study presents a Deepfake Detection System using machine learning techniques to distinguish genuine media from manipulated content. The system analyses visual features extracted from images and video frames and applies classification algorithms for detection. The proposed approach improves robustness and detection performance, providing reliable accuracy under various conditions and making it suitable for real-world applications.
Keywords: *Deepfake Detection, Deep Learning, Convolutional Neural Network (CNN), InceptionV3, Gated Recurrent Unit (GRU)*

1. Introduction
Recent progress in artificial intelligence and deep learning—especially in the area of Generative Adversarial Networks (GANs)—has made it possible to create highly realistic synthetic media, commonly referred to as deepfakes. As highlighted in studies [7] and [8], this technology has evolved rapidly and is now more accessible than ever before. However, the widespread availability of deepfake content has raised serious concerns, including the spread of misinformation, identity misuse, and various forms of cybercrime. Research works such as [9] and [10] emphasize the urgent need for reliable and effective detection mechanisms. In response to these challenges, this paper presents a Deep-Fake Detection System based on machine learning techniques. The proposed approach draws inspiration from facial feature extraction methods discussed in [1] and incorporates adaptive feature fusion strategies from [5] to enhance both accuracy and efficiency in detection. Furthermore, the system leverages convolutional neural networks to automatically learn discriminative features from both spatial and temporal patterns in video data. It also employs preprocessing techniques such as face alignment and normalization to ensure consistency in input data. Experimental results demonstrate that the model achieves high detection accuracy while maintaining computational efficiency suitable for real-time applications. The system is evaluated on standard benchmark datasets to ensure robustness and generalization. Overall, this work contributes toward building more secure and trustworthy digital media environments.

2. System Overview
2.1.Proposed System
The proposed system consists of preprocessing, feature extraction, model training, and classification modules. The system uses facial feature extraction methods from [1] and adaptive feature fusion methods from [5] to increase its detection capabilities.
2.2.System Architecture
System Architecture The system architecture consists of multiple modules including input, preprocessing, feature extraction, model training, and output. The workflow diagram represents the complete process from input media to final classification.
2.3.System Models
System Models Use Case Diagram Illustrates interactions between users and the detection system Class Diagram Represents system classes and their relationships Sequence Diagram Shows the sequence

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0339
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2363 - 2368

of operations during deepfake As shown in Figure 1 Deepfake Detection System Architecture Diagram

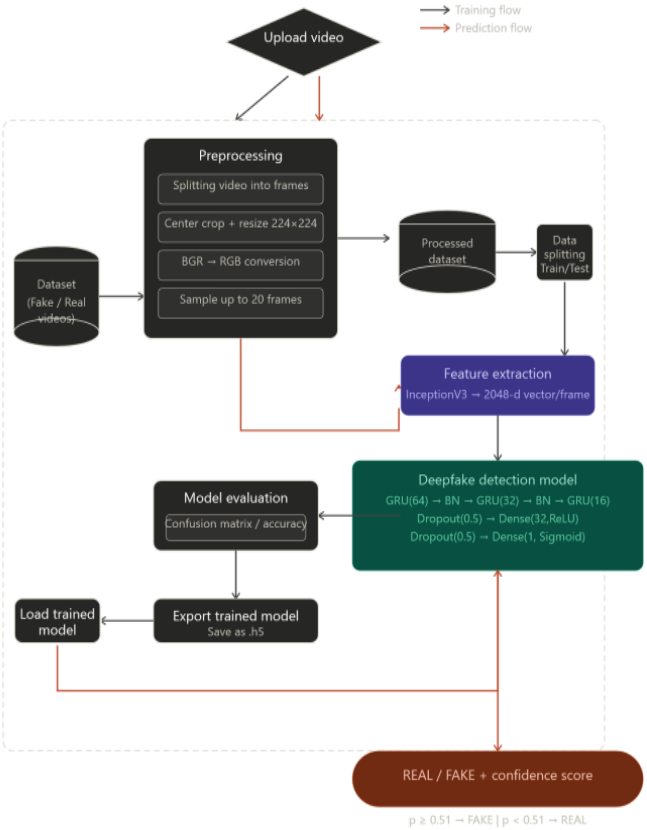

Figure 1 Deepfake Detection System Architecture Diagram

3. Proposed Methodology

The proposed system employs a two-stage deep learning pipeline for deepfake video detection. The first stage uses an InceptionV3 convolutional neural network to extract spatial features from individual frames. The second stage uses a Gated Recurrent Unit (GRU) based[2] sequence model to analyse extracted frame features for temporal pattern detection which enables the system to determine whether the video content is REAL or FAKE. The complete pipeline is described through the following algorithms[3].

3.1.System Overview

Algorithm 0 presents the master pipeline that coordinates all four modules, including pre-processing, feature extraction, model training, and inference. It serves as the main entry point for both training and prediction modes in the system.

Algorithm 0: DeepFake Detection
INPUT:
Video file V
Mode ∈ {TRAINING, INFERENCE}

OUTPUT:
Label L ∈ {REAL, FAKE}
Confidence score p

Step 1:
F ← PREPROCESS(V)
Step 2:
(Φ, M) ← EXTRACT_FEATURES(F)
Step 3:
IF Mode = TRAINING THEN
W* ← TRAIN_MODEL(Φ, M)
Save trained model as "deepfake_model.h5"
ELSE IF Mode = INFERENCE THEN
Load pre-trained weights W*
(p, L) ← CLASSIFY(Φ, M, W*)
Return confidence score p and label L
END IF

3.2.Video Preprocessing

The system uses OpenCV to decode all uploaded videos. The process includes centre-cropping frames to create square images which designers then resize to dimensions of 224×224 pixels. The conversion process changes BGR colour space into RGB colour space. The system uses Algorithm 1 to sample up to 20 frames from each video[4].

Algorithm 1: Video Frame Extraction and Preprocessing
INPUT:
Video file V
Maximum frames T = 20
Target frame size S = 224 × 224

OUTPUT:
Preprocessed frame array F of shape
(T × 224 × 224 × 3)

BEGIN

Step 1:

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0339
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2363 - 2368

Initialize empty frame list
F ← [ ]
Step 2:
Open video stream from V using OpenCV
Step 3:
REPEAT
Read next frame f from video stream
IF frame read fails THEN
BREAK
END IF
y ← height of frame f
x ← width of frame f
min_dim ← minimum(y, x)
start_x ← (x ÷ 2) − (min_dim ÷ 2)
start_y ← (y ÷ 2) − (min_dim ÷ 2)
f ← cropped square region from frame f
f ← Resize(f, S)
f ← Convert(f, BGR → RGB)

Append f to F
IF number of frames in F = T THEN
BREAK
END IF
UNTIL video stream ends
Step 4:
Release video stream
Step 5:
RETURN F as NumPy array
END

3.3.Spatial Feature Extraction

The researchers used InceptionV3 which had been trained on ImageNet for their transfer learning implementation. The system produces 2048-dimensional feature vectors for each frame after the classification head has been removed and Global Average Pooling[5][6] has been applied. The vector contains detailed spatial data which enables users to obtain information without needing to train the convolutional layers for specific tasks. Algorithm 2 formalizes this process[7]

Algorithm 2: Spatial Feature Extraction via InceptionV3 -

INPUT:
Preprocessed frame array F of shape
(N × 224 × 224 × 3)

OUTPUT:
Feature matrix Φ of shape (20 × 2048)
Padding mask M of shape (20,),
where M[j] ∈ {0, 1}

BEGIN

Step 1:
Load the InceptionV3 model with ImageNet weights
Remove the classification layer
Apply Global Average Pooling
Step 2:
Initialize feature matrix
Φ ← zeros(20, 2048)
Initialize padding mask
M ← zeros(20)
Step 3:
Determine valid frame length
video_length ← number of frames in F
length ← minimum(20, video_length)
Step 4:
FOR j = 0 TO length − 1 DO
f_j ← F[j]
f_j ← InceptionV3_preprocess(f_j)
Φ[j] ← InceptionV3(f_j)
M[j] ← 1
END FOR
Step 5:
RETURN Φ and M

END

3.4.Temporal Classification via GRU Sequence Model

The feature[8] matrix generated through Algorithm 2 goes to a stacked GRU network. Masking ensures that zero-padded timesteps are ignored during computation. Batch Normalization stabilizes training between GRU layers while two Dropout layers (rate=0.5) protect against overfitting. The final Dense layer with Sigmoid activation produces a probability score p, threshold at 0.51 to classify the video[11].

Algorithm 3: Deepfake Classification via GRU Sequence Model

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0339
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2363 - 2368

INPUT:
Feature matrix Φ of shape (20 × 2048)
Padding mask M of shape (20,)
Trained model weights W*

OUTPUT:
Confidence score p ∈ [0.0, 1.0]
Predicted label L ∈ {REAL, FAKE}

BEGIN

Step 1:
x ← GRU Layer 1
Units = 6
Return full sequence = TRUE
Apply mask M
Input = Φ
Output shape = (20 × 64)
Step 2:
x ← BatchNormalization(x)
Step 3:
x ← GRU Layer 2
Units = 32
Return full sequence = TRUE
Output shape = (20 × 32)
Step 4:
x ← BatchNormalization(x)
Step 5:
x ← GRU Layer 3
Units = 16
Return full sequence = FALSE
Output shape = (16,)
Step 6:
x ← Dropout(x, rate = 0.5)
Step 7:
x ← Dense Layer(x)
Units = 32
Activation = ReLU
Step 8:
x ← Dropout(x, rate = 0.5)
Step 9:
p ← Dense Layer(x)
Units = 1
Activation = Sigmoid
Step 10:
IF p ≥ 0.51 THEN
L ← "FAKE"
ELSE
L ← "REAL
END IF
Step 11:
RETURN p, L
END

3.5.Model-Training-Procedure

The model uses Binary Cross-Entropy loss together with the Adam[12] optimizer which operates at a learning rate of 1e-4 for its training process. Early Stopping (patience=5) watches over validation loss while it brings back the optimal weights which stop overfitting. A Model Check-point callback saves the best model during training, as summarized in Algorithm[14]

Algorithm 4: Model Training

INPUT:
Training set D_train = (Φ_train, M_train, Y_train)
Validation set D_val = (Φ_val, M_val, Y_val)
Maximum epochs E = 50
Batch size B = 16
Learning rate η = 0.0001
Early stopping patience = 5 epochs

OUTPUT:
Optimized model weights W*

BEGIN

Step 1:
Initialize GRU model architecture
Step 2:
Configure training parameters
Optimizer ← Adam(learning rate = η)
Loss Function ← Binary Cross-Entropy
$L(y, p) = -[\, y \cdot \log(p) + (1 - y) \cdot \log(1 - p) \,]$
Step 3:
Configure callbacks
Early Stopping:
Monitor = Validation Loss
Patience = 5 epochs
Restore best weights = TRUE
Model Checkpoint:
Save best model only = TRUE

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0339
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2363 - 2368

Save path = "deepfake_model.h5"
Step 4:
Initialize monitoring variables
best_val_loss ← ∞
no_improve_count ← 0
Step 5:
FOR epoch = 1 TO E DO
FOR each mini-batch b of size B from D_train DO
p ← ForwardPass(Φ_b, M_b
loss ← BinaryCrossEntropy(Y_b, p)
grads ← Backpropagate(loss)
W ← AdamUpdate(W, grads, η)
END FOR
val_loss ← EvaluateModel(D_val)
IF val_loss < best_val_loss THEN
best_val_loss ← val_loss
W* ← Save current model weights
no_improve_count ← 0
ELSE
no_improve_count ← no_improve_count + 1
IF no_improve_count ≥ 5 THEN
BREAK
END IF
END IF
END FOR
Step 6:
Restore best model weights W*
Step 7:
RETURN W*
END

4. System Design and Implementation

- Technologies Used -
- The frontend development uses HTML, CSS and JavaScript programming languages.
- The backend system operates using Flask framework and Python programming language.
- The system utilizes OpenCV and NumPy and Matplotlib and Pandas and TensorFlow and PyTorch and Kears as its libraries and tools.
- The system uses pre-trained deep learning models which include InceptionV3 and GRU as its artificial intelligence implementation method.
- The following algorithms have been implemented :
- CNN (Convolutional Neural Network) performs image classification through its image processing capabilities[13].
- RNN (Recurrent Neural Network) The system uses face detection algorithms to identify human faces
- The system employees spatial and frequency analysis methods to extract essential features from data.
- The system uses binary classification algorithms to differentiate between real and fake predictions.
- The security features of the system include :
- The system requires user authentication and authorization for access to its resources.
- The system establishes secure API communication channels to protect transmitted data.
- The system performs data validation processes while managing error situations.
- The system implements measures to prevent unauthorized personnel from entering restricted areas. As shown in Figure 1 User Interface View Figure 2 View of video preview

Figure 1 User Interface View

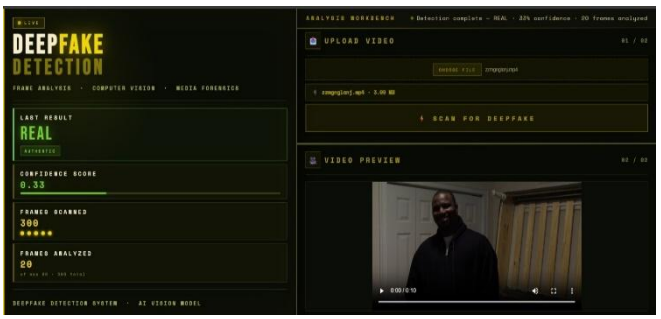

Figure 2 View of video preview

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0339
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2363 - 2368

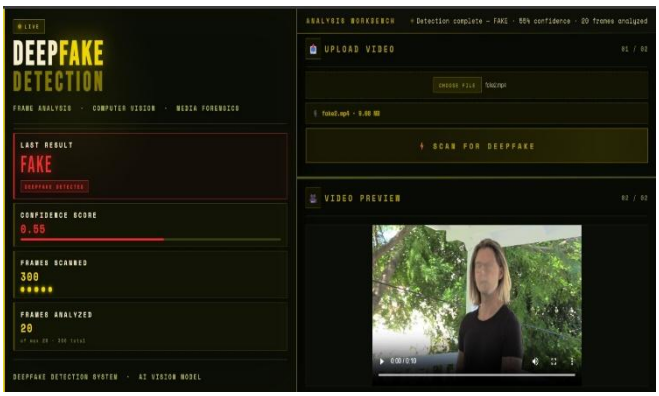

Figure 3 Video Detection Preview

Conclusion

The proposed Deepfake Detection System provides an effective and scalable approach for identifying manipulated media using deep learning techniques. The system automates the processes of preprocessing, feature extraction, model training, and classification to achieve reliable detection performance. Although the system produces accurate results, challenges such as dependency on high-quality datasets and high computational requirements still exist. In the future, the system can be enhanced by integrating more advanced deep learning models to improve detection accuracy and efficiency. Real-time deepfake detection capabilities can also be implemented to strengthen security applications. Additionally, the system can be expanded into mobile and web-based platforms for better accessibility and can incorporate multimodal analysis, including audio feature examination, to improve overall detection performance and reliability.

Acknowledgements

We would like to thank our institution and faculty for their support and guidance in completing this project.

References

[1]. B. Carter, N. Dilla, M. Callahan, and A. Ambala, "Deepfake detection via facial feature extraction and modeling," Canyon Journal of Undergraduate Research, vol. 3, no. 1, pp. 69–75, Jul. 2025.

[2]. U. Ezeakunne, C. Eze, and X. Liu, "Data-driven fairness generalization for deepfake detection," arXiv preprint arXiv:2412.16428, Dec. 2024

[3]. L. Lin, X. He, Y. Ju, X. Wang, F. Ding, and S. Hu, "Preserving fairness generalization in deepfake detection," arXiv preprint arXiv:2402.17229, Feb. 2024.

[4]. Y. Furuhashi, J. Yamagishi, X. Wang, H. H. Nguyen, and I. Echizen, "Exploring active data selection strategies for continuous training in deepfake detection," IEEE Access, 2025.

[5]. F. Wang, et al., "Deepfake detection based on adaptive fusion of spatial and frequency-aware cues," IEEE Access, 2024.

[6]. L. Lin, et al., "Preserving fairness generalization in deepfake detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2024.

[7]. T. Luan, "A survey on deepfake detection technologies," IEEE Transactions on Artificial Intelligence, 2023.

[8]. R. Mubarak, et al., "A survey on the detection and impacts of deepfakes in visual, audio and textual formats," IEEE Access, 2023.

[9]. G. Chandel and A. Kumar, "Deepfake detection using AI and machine learning algorithms," IEEE Access, 2023.

[10]. P. Naga, P. Rayi, R. R. Yandra, and R. Subbanna, "Deepfake detection using AI-based signal processing," Journal of Engineering Studies (JES), 2023.

[11]. C. Sahu, "Deepfake detection system," Shri Rawatpura Sarkar University Report, 2023.

[12]. S. Shrivas, A. Rai, and M. Lakshmi, "Deepfake detection," IEEE Access, 2023.

[13]. S. Stamnas and V. Sanchez, "Exposing deepfakes using differential anomaly detection," University of Warwick, 2023.

[14]. L. Kroiß and J. Reschke, "Deepfake detection of face images based on a convolutional neural network," Ostbayerische Technische Hochschule Regensburg, Germany, 2023.