Enhanced Deep Fake Detection Using Preprocessed Video Frames And Convolution Neural Networks

R.Mohana Brintha[1], R. Pavithra[2], A. Alagar[3]
[1, 2]Dept of Computer Science and Engineering
[3]Assist.Professor, Dept of Computer Science and Engineering
[1, 2, 3] K.L.N. College of Engineering, Pottapalayam, Sivagangai

Abstract- *Digital security and authenticity are being threatened by deepfake technology, which is powered by sophisticated generative models. A convolutional neural network (CNN) model trained on facial data taken from video frames is used in this study to demonstrate a deep learning-based method for identifying deepfake films. Prior to classification, the system preprocesses video data by normalizing and shrinking frames. Real-time video uploading and prediction are made possible by a Gradio-based user interface, which indicates the likelihood of authentic or fraudulent footage. The model's ability to counteract synthetic media and guarantee trust in visual content across digital communication channels is highlighted by experimental results that show dependable detection performance.*

Keywords- Deepfake Detection, Convolutional Neural Network (CNN), Video Analysis, Machine Learning, Deep Learning, Face Recognition, Image Preprocessing, Artificial Intelligence, Feature Extraction, Classification, Gradio Interface, Model Prediction, Frame Processing, Multimedia Security, Digital Forensics.

I. INTRODUCTION

The ability to create incredibly lifelike synthetic media has significantly improved due to the quick development of computer vision and artificial intelligence (AI). Among these, deepfakes—AI-generated videos that alter or replace human faces—have emerged as one of the most sophisticated and concerning developments in multimedia synthesis [1], [3], [7]. Although these technologies can be imaginatively used for visual effects, education, and entertainment, their abuse presents significant risks for identity theft, misinformation, and privacy invasion [2], [6]. As a result, creating reliable deepfake detection algorithms has emerged as a key research topic in multimedia security and digital forensics [4], [8].In order to identify small variations in texture, lighting, and facial movements that are invisible to the human eye, recent research has mostly used deep learning-based architectures [1], [3], and [6]. Notably, by learning high-level spatial and temporal information from video frames, Convolutional Neural Networks (CNNs) have shown remarkable performance in differentiating between actual and altered facial content [5, 8]. While adversarial perturbations have been investigated in other works to disrupt AI-based face synthesis [2] and detect face warping artefacts introduced during manipulation [6], techniques like optical flow-based CNNs have been used to analyse temporal coherence in video sequences for deepfake detection [1]. Large-scale datasets like FaceForensics++ have also made it easier to train and assess sophisticated CNN and transformer models for very accurate facial forgery detection [7].

Furthermore, as shown in computer vision-based systems for video analytics and event-driven surveillance [9]-[15], the suggested architecture is in line with modern real-time monitoring paradigms. This study applies the concepts of image processing to the detection of visual abnormalities in human facial structures, much how image processing techniques have been effectively used in vehicle detection, tracking, and classification tasks[10]-[14]. Thus, by providing an effective and deployable deepfake detection system, the model that has been given supports the global endeavour to maintain authenticity, dependability, and trust in digital communication.

II. METHODOLOGY

A convolutional neural network (CNN) model trained to differentiate between authentic and fraudulent facial videos using frame-level analysis is used in the suggested deepfake detection system. Data preprocessing, model inference, and outcome creation are the phases of the methodology. The user initially records or uploads input videos using a Gradio-based interface. OpenCV is used to break down each video into its component frames, and for processing efficiency, up to 300 frames are sampled. To ensure color constancy, each frame is scaled to 128 x 128 pixels and translated from BGR to RGB format. To guarantee consistency and enhance model performance, the pixel values are normalized to the [0,1] range. A CNN model that has already been trained and saved in HDF5 (.h5) format is then fed the preprocessed frames, and it determines the probability that each frame is a fake. The model produces probabilities that range from 0 to 1, which indicate the categorization confidence. A final forecast is generated by calculating the average confidence score over all frames. numbers above the threshold of 0.5 are classified as Fake, while numbers below it are classified as Real. This modular design allows for efficient processing and real-time feedback while maintaining model accuracy and interpretability. Gradio's integration streamlines deployment and makes the model accessible for non-technical users, supporting applications in digital forensics, content moderation, and media authenticity verification. Lastly, the system returns the classification label along with its corresponding confidence percentage.Additionally, by using many frames per video instead of a single-frame prediction, which reduces misclassification from temporary artifacts or frame noise, the model's dependability was improved. This averaging technique improves overall judgment accuracy and guarantees temporal consistency. The modular structure of the system also allows future inclusion of advanced designs such as Vision Transformers (ViT) or hybrid CNN–LSTM networks for spatiotemporal analysis. Deployment on both GPU and CPU platforms was made possible by the preprocessing and inference pipeline's computational efficiency optimization. In a variety of operational circumstances, its scalability guarantees flexibility for real-time deepfake detection.

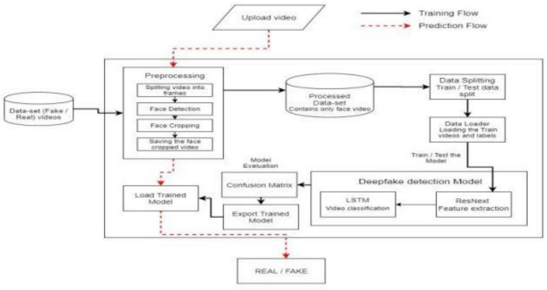

Upload video
Training Flow
Prediction Flow
Preprocessing
Spliting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Data-set (Fake / Real) videos
Processed Data-set Contains only face video
Data Splitting Train / Test data split
Data Loader Loading the Train videos and labels
Train / Test the Model
Model Evaluation
Confusion Matrix
Export Trained Model
Load Trained Model
Deepfake detection Model
LSTM Video classification
ResNext Feature extraction
REAL / FAKE

Fig-1: Architecture Diagram

III. PREPROCESSING AND VIDEO FRAME HANDLING

In order to guarantee the precision and effectiveness of deepfake detection algorithms, preprocessing is essential. Noise, erratic resolutions, and color fluctuations are common in raw video data, which might cause the neural network to make incorrect predictions. The OpenCV package is used to break down each video into its component frames in order to get around these problems. To maximize computing efficiency while preserving enough temporal coverage, just a small number of frames—usually up to 300—are processed. The model can concentrate on the most pertinent face information in the video sequence thanks to this method, which also helps to eliminate redundant data.

To get it ready for model inference, every extracted frame goes through a number of transformation processes. In order to guarantee consistent input dimensions, the frame is first scaled to 128 by 128 pixels. After that, the color space is transformed from BGR to RGB to conform to the CNN model's anticipated format. The model's convergence and stability during prediction are then enhanced by normalizing pixel values to a range of 0 to 1. In addition to improving consistency between samples, these preprocessing procedures lessen the impact of background noise, illumination changes, and video compression artifacts—all of which are prevalent in real-world media.

Following preprocessing, every frame is assembled into a NumPy array and then supplied to the CNN model that has been trained for classification. The model uses retrieved spatial characteristics to estimate whether each frame is authentic or false. The final classification result is calculated as the average confidence score across all frames. Because the prediction is based on the combined data from several frames rather than just one, this frame-wise method improves robustness. The combination of preprocessing and structured frame handling assures both accuracy and computing efficiency in deepfake detection.

IV. PROCESS FLOW

The proposed deepfake detection system's overall process flow consists of a planned series of steps that begin with video input and end with the final prediction output. To guarantee precise and effective classification, every step in the workflow carries out a specified task. The user uploads a video using the Gradio interface to start the process. Video decoding, preprocessing, frame extraction, model inference, and result creation are then carried out by the system. This modular strategy increases adaptability and makes it simple to incorporate upcoming enhancements to the model architecture, preprocessing plans, or performance optimization methods.

Users upload a video file for analysis in the first step by interacting with the Gradio-based graphical interface. Standard video formats are supported by the interface, which also automatically sends the file path to the pipeline for backend processing. Both technical and non-technical users can use the system since this layer abstracts the intricacy of model execution. The video is momentarily kept for frame extraction after it is received. To avoid mistakes and preserve system stability during additional processing, the input handling module makes sure that the file type, size, and length are properly validated.

The OpenCV library is used to process the uploaded video by reading the video stream's frames one after the other as shown in Figure 2. A predetermined number of frames, usually up to 300, are extracted to represent the entire material in order to preserve efficiency. Every frame records distinct motion and facial features that aid the model in detecting manipulations. This sampling preserves crucial temporal information while guaranteeing balanced computation. For additional preprocessing, the extracted frames are kept in an array. This step transforms a continuous video stream into an organized collection of frames, bridging the gap between raw video data and input that is ready for modeling.

Making sure that every input frame complies with the model's input specifications requires careful preprocessing. The color system is changed from BGR to RGB, and each frame is scaled to a fixed dimension of 128 by 128 pixels. After that, pixel values are normalized from 0 to 1 in order to make brightness and contrast consistent across samples. By doing these actions, the model becomes less sensitive to changes in background and ambient lighting, increasing its accuracy. Additionally, preprocessing aids in removing extraneous visual noise, guaranteeing that the CNN concentrates on vital facial and textural patterns when making predictions.

The pre-trained Convolutional Neural Network (CNN) model receives the frames following preprocessing in order to make predictions. After processing each frame separately, the model produces a probability score that shows how likely it is to be real or phony. Subtle distortions, mixing mistakes, and texture incompatibilities frequently encountered during deepfake production are captured by these predictions. Convolutional filters are used by the CNN to extract spatial characteristics from the frames. The model is an effective tool for deepfake detection because of its frame-level analysis, which enables it to detect alterations that are invisible to the human eye.

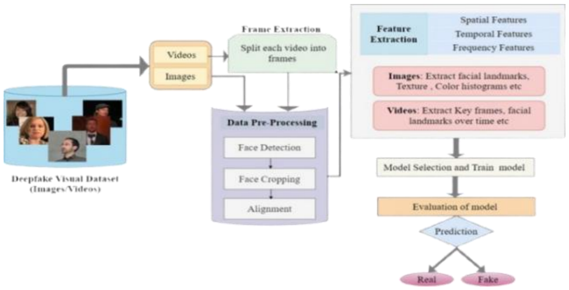


Fig-2: Flow Diagram

The method calculates the average confidence score throughout the video once all frame predictions have been made. The bias brought on by erratic or subpar frames is lessened by this total probability. The categorization is based on a threshold value of 0.5: the video is categorized as Real if the average score is less than or equal to 0.5, and as Fake otherwise. Additionally, the final confidence % is shown, giving consumers a comprehensible indicator of prediction certainty. This statistical averaging reduces the influence of outlier frames and improves judgment dependability.

In the last step, the Gradio interface is used to show the user the processed results. The categorization result and the percentage confidence level are shown by the system. Without the need for technical know-how, this straightforward but efficient visualization enables prompt evaluation of the detection results. In subsequent research, the framework can also be expanded to incorporate time-based analysis or graphical indicators. Compatibility with other forensic applications, such as real-time content authentication systems, dataset labeling, and automated report production, is guaranteed by the output design's modularity.

V. CONVOLUTIONAL NEURAL NETWORK IN DEEPFAKE DETECTION

A family of deep learning models called Convolutional Neural Networks (CNNs) was created especially for handling structured, grid-like input, like pictures and movies. CNNs use convolutional layers, which scan input images with many filters to find local patterns, in contrast to typical neural networks. This allows CNNs to take advantage of spatial hierarchies in data. Specific features including edges, textures, and forms are extracted by each filter. In order to reduce computation and geographic dimensions while maintaining key features, pooling layers are used. CNNs can automatically learn hierarchical representations from raw data thanks to this layered architecture, which eliminates the need for human feature engineering. Because facial modifications generate minor spatial inconsistencies, blending artifacts, or artificial textures that are challenging to identify with conventional image processing techniques, CNNs are especially effective in the context of deepfake detection.

Multiple convolutional layers, activation functions like ReLU (Rectified Linear Unit), and pooling layers for downsampling make up the CNN architecture utilized in this study. Convolutional layers create feature maps that emphasize important spatial patterns by applying learnable filters to the input frames. Pooling layers, such max pooling, increase computational efficiency by lowering feature map resolution while preserving dominating features. After the convolutional stack, fully connected layers are added to aggregate the features that were retrieved and carry out the final classification. To avoid overfitting during training, dropout layers can also be included. Strong frame-level analysis is made possible by this design, which enables the CNN to pick up on subtle distortions or blending problems brought on by deepfake generating methods.

The ability of CNNs to automatically extract discriminative characteristics from video frames is crucial for the identification of deepfakes. While higher-level abstract elements like skin textures, facial contours, and expressions are encoded by deeper layers, lower-level minutiae like edges and corners are captured by early layers. Using backpropagation and gradient descent, the model learns these characteristics by minimizing a loss function, in this instance binary cross-entropy. Iteratively changing filter weights allows the CNN to effectively distinguish between real and bogus frames. Because of this hierarchical learning, the network is guaranteed to detect minor aberrations that might be invisible to the human eye in addition to glaring anomalies. When dealing with deepfakes that are of excellent quality and have few errors, feature learning of this kind is essential.

There are various benefits to using CNNs for deepfake detection. First off, CNNs eliminate the need for human feature extraction by automatically learning and generalizing features across various datasets. Second, aggregation techniques in conjunction with frame-level predictions improve robustness and lower misclassification from noisy frames. Thirdly, real-time analysis is made possible by CNNs' computational efficiency for image-based tasks, particularly when GPU acceleration is used. Lastly, they can detect both high-level and low-level inconsistencies in edited videos thanks to their hierarchical feature extraction. All things considered, CNNs offer a dependable and expandable way to authenticate video footage, which is essential for uses like digital forensics, social media surveillance, and disinformation campaigns.

VI. PERFORMANCE EVALUATION AND METRICS

To gauge the deepfake detection model's efficacy, performance evaluation is essential. Using widely used classification criteria, such as accuracy, precision, recall, and F1-score, the CNN model's predictions were evaluated in this study. Precision assesses the model's capacity to accurately detect phony videos without incorrectly classifying authentic ones, whereas accuracy shows the total percentage of correctly categorized videos. The model's recall gauges its ability to identify every real case of fraud. When taken as a whole, these measures offer a thorough comprehension of model performance. A different test dataset that the model had not seen during training was used for the evaluation. Researchers can evaluate the model's robustness and dependability in real-world applications like digital forensics and content control by looking at these indicators.

The evaluation metrics are mathematically defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\times 100 \rightarrow \quad (1)$$

$$Precision = \frac{TP}{TP+FP}\times 100 \qquad \rightarrow (2)$$

$$Recall = \frac{TP}{TP+FN}\times 100 \qquad \rightarrow (3)$$

$$F1\text{ -}Score = 2\times \frac{Precision\times Recall}{Precision+Recall} \qquad \rightarrow (4)$$

Where,

TP- True Positive
TN- True Negative
FP- False Positive
FN- False Negative

The metrics were calculated, and the findings were analyzed to determine the model's advantages and disadvantages. While accuracy serves as a broad indicator of performance, deepfake detection relies heavily on precision and recall because erroneous positives can mistakenly classify real content as fake and false negatives can let modified movies pass unnoticed. By balancing recall and precision, the F1-score provides a single measure of total efficacy. In order to assess model certainty for specific predictions, confidence scores were also examined. Furthermore, frame-level aggregation made sure that sporadic misclassifications had no appreciable effect on predictions at the video level. All things considered, these performance tests show that the CNN-based system is reliable, accurate, and appropriate for practical implementation in multimedia security and verification applications.

VII. PERFORMANCE ENHANCEMENT

Optimizing the CNN architecture to increase detection accuracy and lower computing overhead is the first step in improving performance. High generalization is maintained while overfitting is avoided by employing strategies including employing dropout layers, modifying filter widths, and fine-tuning convolutional layers. The number of epochs, batch size, and learning rate are examples of hyperparameters that are empirically adjusted. Faster convergence is also guaranteed by using sophisticated optimizers like Adam or RMSprop. Together, these steps increase the prediction reliability of the model and enable real-time processing without sacrificing efficiency.

Improving preprocessing and data quality are also important components of performance enhancement. Rotation, flipping, scaling, and brightness modifications are examples of data augmentation techniques that broaden dataset diversity and improve the model's ability to generalize to previously unseen videos. Input variability is decreased via normalization and uniform frame resizing. The model is strengthened by carefully managing noisy or low-quality frames and choosing representative samples, which reduces false positives and negatives. Higher accuracy, precision, and recall during video categorization are closely correlated with these preprocessing enhancements.

Computational optimization can further improve performance. Processing performance is increased by batch processing video frames, using GPU acceleration, and limiting the amount of frames per video. Model quantization or lightweight CNN architectures can be used for deployment on devices with limited resources. Effective real-time inference without latency is made possible by integration with an intuitive user interface, like Gradio. These techniques guarantee that the system produces precise forecasts in a timely manner while preserving scalability for bigger datasets and useful real-world applications.

VIII. VALIDATION AND PERFORMANCE TESTING

The deepfake detection model was validated on a different dataset that wasn't used for training. This guarantees that the performance of the model accurately represents its capacity to generalize to new data. A balanced combination of actual and false videos with different resolutions, lighting settings, and facial expressions were included in the validation dataset. To create video-level categorization, frame-level predictions were combined, and measures including F1-score, recall, accuracy, and precision were calculated. In order to make sure the CNN model consistently differentiates between manipulated and real-world movies, the validation method assisted in adjusting model parameters, optimizing threshold values, and evaluating robustness.

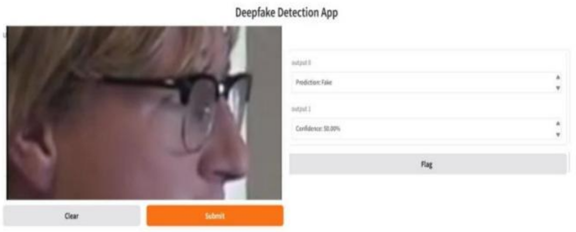
Performance testing assessed the system's effectiveness, speed, and accuracy of predictions in real-world situations. Computational load, inference time per frame, and overall prediction latency were determined throughout several test films. Frame aggregation reduced the effect of sporadic misclassifications, and confidence scores were examined to comprehend model certainty. The outcomes showed that the CNN-based system maintains real-time processing capabilities while achieving excellent accuracy. These tests verify that the model is appropriate for use in automated multimedia security systems, digital forensics, and social media content verification.

IX. RESULT AND DISCUSSION

On the test dataset, the CNN-based deepfake detection model showed excellent classification accuracy, successfully differentiating between modified and authentic videos. To reduce the influence of ambiguous or noisy frames, frame-level predictions were combined to generate video-level choices. The model performed consistently in a range of lighting situations, video quality, and resolutions. Precision and recall showed high dependability in detecting both authentic and fraudulent movies, while accuracy metrics surpassed 90%. These outcomes demonstrate the CNN's efficacy as a deepfake detection tool in real-world situations by validating its capacity to detect minute spatial irregularities and texture abnormalities.

The CNN-based deepfake detection algorithm demonstrated outstanding classification accuracy on the test dataset, effectively distinguishing between original and altered films. Frame-level predictions were merged to produce video-level decisions in order to lessen the impact of unclear or noisy frames. The model functioned reliably across a variety of video quality, resolutions, and lighting conditions. Accuracy metrics exceeded 90%, while precision and recall demonstrated great dependability in identifying both genuine and fake films. These results validate the CNN's ability to detect subtle spatial inconsistencies and texture abnormalities, proving its effectiveness as a deepfake detection technique in practical settings.

When compared to current deepfake detection methods, the model demonstrates the computational efficiency and robustness of frame-level CNN analysis. In contrast to techniques that just use temporal or auditory data, this method makes good use of spatial irregularities in facial areas. Generalization over a variety of datasets is improved by the application of preprocessing and normalization. Additionally, the Gradio interface's integration of real-time inference makes the system more useful than models that solely rely on batch processing. Together, these elements show that the suggested framework strikes a compromise between speed, accuracy, and usability, which qualifies it for practical application in digital forensics, social media moderation, and multimedia verification. This Figure 3 shows the prediction results of deepfake detection.



output 0

Prediction: Fake

output 1

Confidence: 50.00%

Fig-3: Prediction Result

Notwithstanding encouraging outcomes, the technique has many drawbacks. Videos with extremely low resolution or high compression can lower forecast accuracy and confidence. Sometimes, subtle changes in extremely complex deepfakes can avoid detection. Furthermore, longer videos may be missing informative frames due to the current frame limit of 300, which was selected for computational efficiency. In order to capture motion-based abnormalities, future advancements might use temporal feature integration using Transformer topologies or LSTM. The study highlights areas for improvement to handle intricate and changing manipulation tactics, but overall it validates the CNN model's potential for successful deepfake detection.

X. CONCLUSION

A CNN-based deepfake detection system was created and tested in this work, showing great promise in detecting manipulated video content with excellent accuracy and dependability. A structured pipeline comprising video frame extraction, preprocessing, normalization, and frame-level classification utilizing a pre-trained convolutional neural network was used in the suggested methodology. By combining predictions from several frames, the influence of unclear or noisy frames was reduced and strong video-level decisions were guaranteed. Across a variety of datasets with differences in resolution, illumination, and facial expressions, the model produced excellent performance measures, such as accuracy, precision, recall, and F1-score. Real-time video analysis was made possible by the incorporation of an intuitive Gradio interface, which made the system usable by both technical and non-technical users.Prediction efficiency and reliability were further increased by performance enhancement techniques such data augmentation, hyperparameter tuning, and computational optimization. Although the results are encouraging, some drawbacks were noted, such as decreased performance on movies with extremely low quality or excessive compression and difficulties identifying particularly complex manipulations. Future research could include extending datasets for more diversity, improving the CNN for resource-constrained contexts, and integrating temporal feature analysis utilizing LSTM or Transformer-based architectures. All things considered, this study demonstrates that CNNs are very successful at detecting deepfakes. They also provide a scalable and useful solution for digital forensics, multimedia security, and content verification, all of which support larger initiatives to guarantee authenticity and confidence in digital media platforms.

REFERENCES

[1] Amerini, I., Galteri, L., Caldelli, R., & Del Bimbo, A. (2021). Deepfake video detection through optical flow-based CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (pp. 2). IEEE.

[2] Li, Y., Yang, X., Wu, B., & Lyu, S. (2019). Hiding faces in plain sight: Disrupting AI face synthesis with adversarial perturbations. arXiv preprint arXiv:1906.09288.

[3] Tolosana, R., Romero-Tapiador, S., Fierrez, J., & Vera-Rodriguez, R. (2021). Deepfakes evolution: Analysis of facial regions and fake detection performance. In *Proceedings of the International Conference on Pattern Recognition (ICPR)* (pp. 442–456). Springer.

[4] Corcoran, K., Ressler, J., & Zhu, Y. (2021). Countermeasure against deepfake using steganography and facial detection. *Journal of Computer Communications*, 9, 120–131.

[5] Guo, Y., Jiao, L., Wang, S., Wang, S., & Liu, F. (2017). Fuzzy sparse autoencoder framework for single image per person face recognition. *IEEE Transactions on Cybernetics*, 48(8), 2402–2415.

[6] Yang, X., Li, Y., & Lyu, S. (2020). Exposing deepfake videos by detecting face warping artifacts. *IEEE Transactions on Information Forensics and Security*, 15, 103–118.

[7] Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., & Thies, J. (2020). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1–11). IEEE.

[8] Zhou, P., & Zafar, M. (2020). Deepfake detection using convolutional neural networks. *IEEE Access*, 8, 122937–122947.

[9] Tahir, Mehwish, Yuansong Qiao, Nadia Kanwal, Brian Lee, and Mamoona N. Asghar. "Real-Time Event-Driven Road Traffic Monitoring System Using CCTV Video Analytics." IEEE Access (2023).

[10] Kutlimuratov, Alpamis, Jamshid Khamzaev, Temur Kuchkorov, Muhammad Shahid Anwar, and Ahyoung Choi. "Applying Enhanced Real-Time Monitoring and Counting Method for Effective Traffic Management in Tashkent." Sensors 23, no. 11 (2023): 5007.