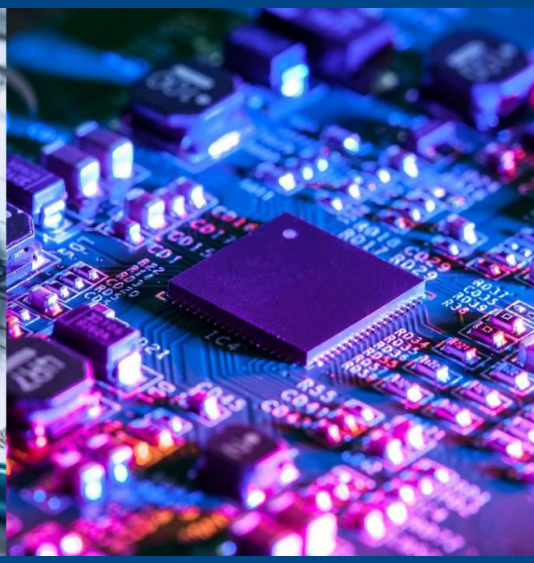
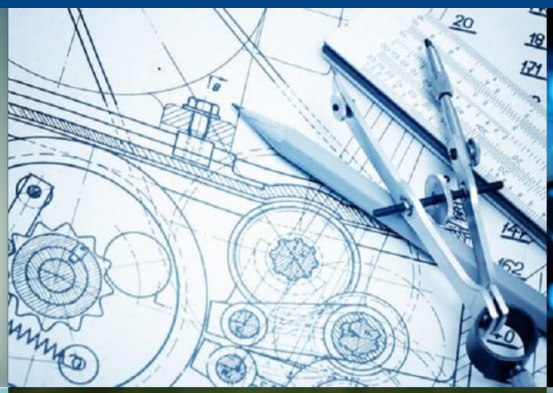
ISSN: 2582-7219

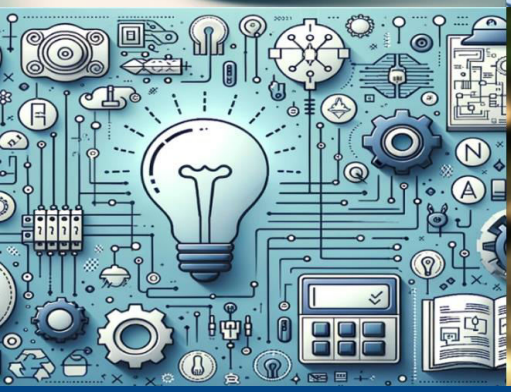




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 8, Issue 6, June 2025**

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806188

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Deep Fake Video Detection using Machine Learning

Smitha S M[1], Aniruddha[2], Akhilesh S U[3], Anusamrat M[4], Akash K N[5]

UG Students, Dept. of ECE, JNN College of Engineering, Shivamogga, Karnataka, India[2-5]

Assistant Professor, Dept. of ECE, JNN College of Engineering, Shivamogga, Karnataka, India[1]

ABSTRACT: With the rapid evolution of computational technologies and deep learning techniques, generating highly convincing AI-based synthetic videos—commonly termed deepfakes—has become increasingly feasible. These artificially manipulated videos, produced using frameworks such as Face2Face and Deepfake, present serious risks to societal trust by enabling unethical use cases including political deception, staged terror events, extortion, and non-consensual content distribution. To address these growing concerns, we introduce an advanced deep learning framework capable of accurately identifying manipulated videos. The proposed system employs a ResNeXt convolutional neural network for capturing spatial features at the frame level, combined with a long short-term memory (LSTM) network to model temporal dependencies across sequences. This hybrid model is proficient in detecting various deepfake categories, including identity swaps and expression reenactments. Training is conducted on a comprehensive and diverse dataset comprising FaceForensics++, the Deepfake Detection Challenge, and Celeb-DF, supported by rigorous preprocessing and data augmentation techniques to enhance generalization and performance. Through the use of refined LSTM optimization strategies, our model achieves robust and competitive results while maintaining architectural simplicity, offering a dependable machine learning-based approach to mitigate the threats posed by deepfake content.

I. INTRODUCTION

In an era shaped by technological breakthroughs, the expanding horizons of innovation bring both extraordinary possibilities and profound challenges. Among the most urgent of these challenges is the manipulation of digital media, which poses a serious threat to the foundations of trust in our interconnected world. Such manipulations not only undermine public confidence but also destabilize institutions, fuel the spread of misinformation, and cause irreparable harm to individuals by distorting the truth. The consequences ripple through society, fostering deep mistrust, intensifying polarization, and eroding the credibility of media and democratic systems. On a personal level, these issues can inflict severe psychological distress, tarnish reputations, and damage relationships, highlighting the pressing need for solutions that safeguard both individual well-being and societal stability. Addressing these multifaceted challenges requires a blend of innovation and precision, with machine learning emerging as a pivotal force in the fight against digital manipulation. Machine learning, with its ability to detect imperceptible patterns and irregularities, offers a sophisticated approach to identifying and mitigating threats that would otherwise go unnoticed. By harnessing this technology, we can create tools capable of protecting communities, preserving trust, and restoring authenticity to the digital landscape. In doing so, we ensure that the rapid advancements in technology are directed toward ethical, responsible, and beneficial outcomes, reinforcing its role as a powerful driver of progress and positive change.

II. LITERATURE REVIEW

Recent studies have explored various strategies to address the growing threat of deepfake content. One approach leverages a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to detect forged facial videos by capturing both spatial and temporal inconsistencies across frames. Such hybrid architectures have shown improved accuracy in distinguishing real from manipulated content.Comprehensive reviews of deepfake technology emphasize its rapid advancement and highlight the social, political, and ethical concerns associated with its misuse. These works underline the need for proactive detection mechanisms and regulatory measures.Another line of research focuses on training CNNs to differentiate between real and computer-generated facial images. These models are trained on mixed datasets and demonstrate high performance in classification tasks, effectively laying the groundwork for real-time detection systems.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Some researchers propose innovative biometric-based methods, such as detecting abnormal eye-blinking behavior, which is often overlooked in synthetic videos. Temporal modeling of such subtle cues enhances the reliability of forgery detection in video sequences.Recurrent convolutional architectures have also been explored for facial manipulation detection. By integrating temporal analysis with spatial feature extraction, these models can identify inconsistencies resulting from techniques like face-swapping or expression reenactment.In efforts to develop lightweight and real-time solutions, compact neural networks focusing on mesoscopic-level features have been introduced. These models balance computational efficiency with detection accuracy, making them suitable for deployment in constrained environments.Large-scale datasets have been developed to support and benchmark deepfake detection algorithms. These datasets include a variety of manipulation types and enable standardized performance comparisons across detection models, thereby accelerating progress in the field.Lastly, advancements in meta-learning offer promising directions for adaptable deepfake detection. Model-agnostic frameworks allow networks to quickly adjust to new manipulation styles or domains, which is essential in keeping pace with evolving generative techniques.

III. DESIGN AND IMPLEMENTATION

With the widespread use of social media platforms, deepfakes have emerged as a significant threat associated with artificial intelligence. These highly realistic, face-swapped videos are often exploited in malicious contexts such as political misinformation, fabricated criminal incidents, revenge pornography, and blackmail. High-profile examples involving public figures like Brad Pitt and Angelina Jolie underscore the urgency of developing reliable detection mechanisms.

Deepfakes are typically generated using software tools such as FaceApp and FaceSwap, which rely on pre-trained neural networks based on Generative Adversarial Networks (GANs) or autoencoders. These models learn to mimic facial movements and expressions with high precision, making manual detection extremely challenging. To address this, our proposed system employs a two-stage deep learning architecture. First, a pre-trained ResNeXt convolutional neural network (CNN) is used to extract detailed spatial features from individual video frames. These frame-level features are then passed to a Long Short-Term Memory (LSTM) based recurrent neural network, which captures temporal dependencies across frames to identify subtle inconsistencies introduced during deepfake manipulation.The model is trained on a large and diverse dataset compiled from sources including FaceForensics++, the Deepfake Detection Challenge, and Celeb-DF. This ensures exposure to various manipulation techniques and helps the system generalize well across real-world scenarios. To further enhance robustness, the dataset is balanced and undergoes preprocessing and augmentation.

For practical deployment, a user-friendly frontend application has been developed. Users can upload a video through the interface, which is then processed by the backend detection model. The system analyzes the input and returns a classification label—either “deepfake” or “real”—along with the model’s confidence score, providing a transparent and efficient solution for real-time deepfake detection.

In our system, the deepfake detection model was developed using the PyTorch framework and trained on a balanced dataset comprising an equal number of real and manipulated video samples. This balance was maintained deliberately to mitigate any potential bias during the training process and to ensure consistent classification performance across both categories. The architectural design of the proposed model is depicted in the accompanying system diagram. During the development phase, raw datasets were subjected to preprocessing, including face detection and cropping, to isolate the regions of interest. This step ensured that only face-centric video segments were used for training and evaluation, thereby improving the model’s focus and accuracy. To effectively detect deepfakes, it is crucial to understand their generation process. Most deepfake creation tools—including those based on Generative Adversarial Networks (GANs) and autoencoders—operate by taking a source image and a target video as input. These tools extract frames from the video, detect the facial region in each frame, and sequentially replace the original face with the source face. The modified frames are then reassembled into a video using various pre-trained models that enhance visual realism by minimizing residual artifacts.Although these synthetic videos often appear convincingly realistic, the generation process tends to leave subtle inconsistencies or artifacts—such as edge mismatches, unnatural blinking patterns, or minor texture distortions—which are not easily detectable by the human eye. Our approach leverages these residual cues through deep learning-based feature extraction and temporal analysis to effectively distinguish deepfakes from genuine videos.

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806188

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

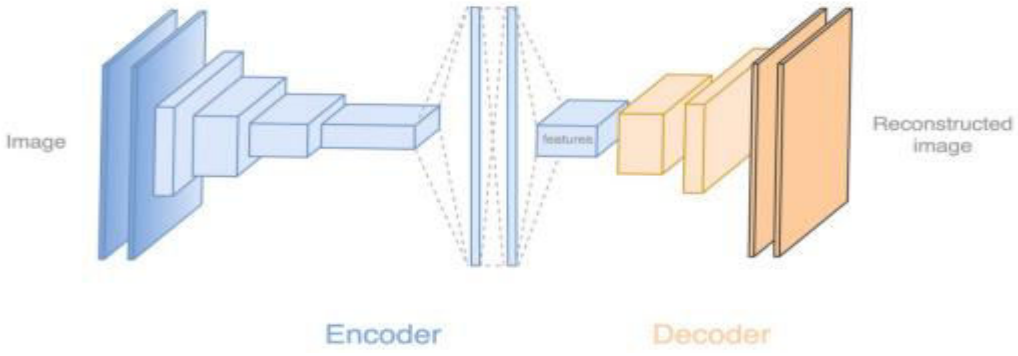


Figure 3.1: Deepfake architecture

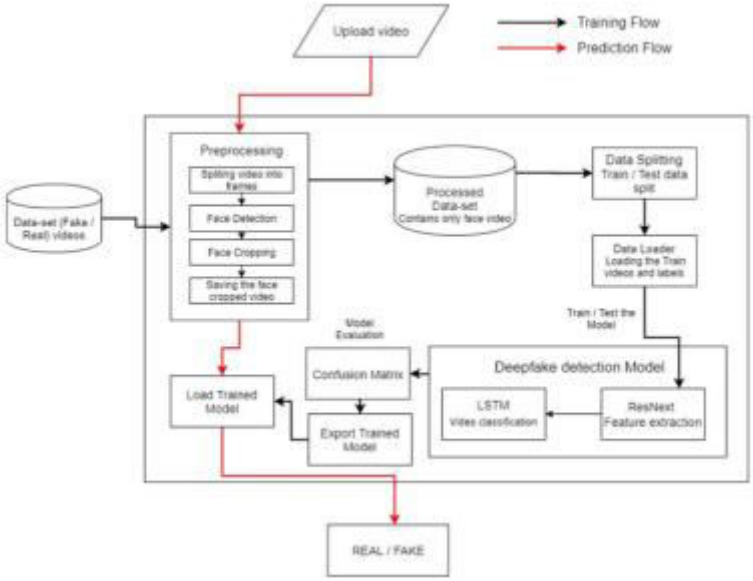


Figure 3.2: System Architecture

IV. METHODOLOGY

Model Architecture Details

The proposed deepfake detection framework is built using a hybrid deep learning architecture comprising convolutional and recurrent neural network components. Each component plays a crucial role in effectively capturing spatial and temporal features from input video data.

- **ResNeXt Convolutional Neural Network (CNN):** The model incorporates a pre-trained ResNeXt-50 CNN (32×4d) to extract frame-level spatial features. This residual network architecture includes 50 layers and leverages grouped convolutions to enhance performance while maintaining computational efficiency. The ResNeXt model processes individual frames and outputs 2048-dimensional feature vectors representing key facial characteristics.
- **Sequential Layer:** A sequential container is used to store and organize the extracted feature vectors in temporal order. This ensures that features from video frames are preserved in their original sequence, which is essential for accurate temporal analysis by the LSTM.
- **Long Short-Term Memory (LSTM) Layer:** To capture temporal dependencies between video frames, a single-layer LSTM network is employed. It takes the 2048-dimensional frame-level features as input, with 2048 hidden units and a dropout rate of 0.4 to mitigate overfitting. The LSTM enables the model to identify subtle frame-to-frame

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

variations, such as inconsistencies introduced during deepfake generation. By comparing features at time t with those at t–n, the LSTM effectively models short- and long-term dependencies.

- **ReLU Activation Function:** The Rectified Linear Unit (ReLU) is used as the activation function to introduce non-linearity into the model. ReLU outputs zero for negative inputs and returns the input directly for positive values. It is computationally efficient and well-suited for deep architectures, helping to avoid vanishing gradient problems commonly seen with sigmoid activations.
- **Dropout Layer:** A dropout layer with a rate of 0.4 is integrated after the LSTM to reduce overfitting and improve the model's generalization capability. By randomly disabling neurons during training, the model becomes less sensitive to specific neurons and learns more robust feature representations.
- **Adaptive Average Pooling Layer:** A 2D Adaptive Average Pooling layer is included to reduce feature map dimensionality and extract relevant local patterns. This operation helps minimize variance and computational cost while preserving critical information for the final classification stage.

This combination of spatial and temporal processing layers enables the model to analyze both appearance-based features and temporal inconsistencies, enhancing its ability to distinguish deepfake videos from genuine ones with high reliability.

Training Details:

Dataset Splitting: The dataset is divided into training and testing subsets, ensuring a larger portion for training (approximately 1,500 videos) while maintaining a sufficient number for testing. Both sets are balanced, containing an equal distribution of real and deepfake videos—50% from each class—to prevent model bias.

Data Loading: A DataLoader is implemented to efficiently manage the input pipeline, loading video batches along with their corresponding labels. The batch size is configured to 4 to optimize GPU memory usage and processing efficiency.

Model Training: The model is trained over 20 epochs with a learning rate of 0.00001 and a weight decay of 0.001. These parameters are optimized to promote stable learning and reduce overfitting. The Adam optimizer is chosen for its adaptive learning capabilities and ability to handle sparse gradients effectively.

Loss Function – Cross Entropy: Cross-entropy loss is used as the objective function, suitable for binary classification tasks. It measures the divergence between predicted outputs and actual labels, guiding the model toward accurate predictions.

Activation – Softmax Layer: A softmax function is used in the final output layer to convert raw model scores into normalized probability values. This allows the output to be interpreted as the confidence level of class predictions. Since the task involves binary classification, the softmax layer contains two nodes: one representing **REAL** and the other **FAKE**.

Performance Evaluation – Confusion Matrix: To evaluate model performance, a confusion matrix is employed. It presents the number of correct and incorrect predictions categorized by class, offering valuable insights into the types of errors made. This tool is crucial for understanding the strengths and weaknesses of the classifier and for computing accuracy and other performance metrics.

Model Export: Once training is complete, the trained model is exported for deployment. This allows the system to be used for real-time prediction on incoming video data, supporting practical deepfake detection scenarios.

Architectural Design

Dataset Collection and Preparation

To ensure the model is optimized for real-time deepfake detection, we curated a diverse dataset by aggregating videos from multiple publicly available sources, including FaceForensics++ (FF), the Deepfake Detection Challenge (DFDC), and Celeb-DF. These datasets were merged to form a custom dataset designed to enhance the system's performance across various video types and manipulation techniques.To mitigate model bias during training, an equal number of authentic and manipulated videos were selected—50% real and 50% fake. Since the scope of this study is limited to visual deepfakes, we excluded audio-altered samples found within the DFDC dataset. A Python script was employed to filter out such audio-modified videos.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

After preprocessing, the dataset comprised:

- **DFDC**: 1,000 real and 1,000 fake videos
- **FaceForensics++**: 500 real and 500 fake videos
- **Celeb-DF**: 200 real and 200 fake videos

This aggregation resulted in a balanced dataset of **1,700 real and 1,700 fake videos**, totaling **3,400 video samples** for training and evaluation.

Preprocessing

In this phase, each video undergoes a structured preprocessing pipeline to isolate and retain only the essential visual information—specifically, the facial region—while removing background noise and irrelevant content. The objective is to generate a refined dataset composed solely of face-centric video clips suitable for deepfake detection.The preprocessing begins by splitting each video into individual frames. Facial detection is then applied to every frame, and the region containing the face is cropped accordingly. Frames without detectable faces are discarded. Once the facial regions are extracted, the cropped frames are recombined to form a new video composed exclusively of facial content. This process is systematically applied to all videos, resulting in a processed dataset where each sample contains only the facial region.

To ensure consistency and address computational limitations, a threshold for the number of frames per video was defined. Given that a 10-second video at 30 frames per second (fps) yields 300 frames, processing all of them simultaneously can be computationally intensive. Therefore, based on the experimental constraints of our Graphics Processing Unit (GPU), we limited each processed video to the first **150 sequential frames**. This also preserves temporal coherence for LSTM-based sequence modeling.

All newly generated face-only videos were standardized to a frame rate of **30 fps** and a fixed resolution of **112 × 112 pixels**, ensuring uniform input for subsequent model training.

Figure 4.1: Preprocessing of video

Model Architecture

The proposed architecture integrates both Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to effectively identify deepfake content. Specifically, a pre-trained **ResNeXt-50 (32x4d)** model is employed for **frame-level feature extraction**, while a **Long Short-Term Memory (LSTM)** network is used to perform **temporal analysis** across video frames for classification.The model workflow begins with the input video being divided into frames. Each frame is passed through the ResNeXt model, which is optimized for deep architectures and efficient

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

learning via residual connections. ResNeXt outputs **2048-dimensional feature vectors**, which capture high-level spatial characteristics from each frame.

These feature vectors are then passed sequentially into a single-layer **LSTM network**. The LSTM has **2048 latent dimensions** and **2048 hidden units**, with a **dropout rate of 0.4** to reduce overfitting. The LSTM is responsible for learning temporal dependencies between frames—enabling the model to detect subtle inconsistencies across time that are indicative of manipulated content. The frame at time t is compared with previous frames (t - n) to capture motion patterns and anomalies.

Additional components of the model include:

- **Sequential Layer**: Used to organize frame features in temporal order before input to the LSTM.
- **Adaptive Average Pooling Layer**: With an output size of 1×1, it reduces spatial dimensions and variance in extracted features before passing them to fully connected layers.
- **Fully Connected (Linear) Layer**: Transforms the 2048-dimensional LSTM output into two final logits corresponding to "Real" and "Fake" classes.
- **Leaky ReLU Activation Function**: Applied to introduce non-linearity while avoiding the dead neuron issue associated with standard ReLU.
- **Softmax Layer**: Converts final logits into class probabilities, enabling interpretation of the prediction with associated confidence.

The model is trained using a **batch size of 4**, and training labels are loaded via a **PyTorch DataLoader**. This architecture allows the system to not only classify videos as real or fake but also to provide a confidence score for each prediction.

Hyperparameter Tuning

Hyperparameter tuning plays a crucial role in optimizing model performance by carefully selecting parameters that influence the learning process. After multiple iterations and experimentation, the most effective set of hyperparameters for our dataset was identified. To support an adaptive learning mechanism, the **Adam optimizer** was employed in conjunction with the model parameters. A **learning rate of 1e-5 (0.00001)** was selected to ensure smoother convergence towards a global minimum during gradient descent. Additionally, a **weight decay of 1e-3 (0.001)** was used to prevent overfitting by penalizing large weights.

Since the task is a **binary classification problem**, **cross-entropy loss** was used as the objective function, which is well-suited for measuring the difference between predicted probabilities and actual class labels.To make efficient use of available computational resources, **batch training** was implemented. A **batch size of 4** was found to be optimal for our hardware setup, allowing for stable training without exhausting GPU memory.

Furthermore, a simple yet scalable **user interface** was developed using the **Django web framework**, enabling real-time interaction with the model. The main page (index.html) allows users to upload a video for analysis. Once uploaded, the video is processed by the trained model, which returns a prediction label—**"Real"** or **"Fake"**—along with the model's **confidence score**. The prediction result is displayed on a separate results page (predict.html), overlaid on the video during playback for clear visualization.

This integrated architecture ensures a user-friendly experience while maintaining high model performance and scalability for future deployment.

Data flow

Data Flow Diagram – Level 0

The Level 0 Data Flow Diagram (DFD) provides a high-level overview of the system's data processing flow, emphasizing both input and output components equally.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Input**: The system begins with the user uploading a video, which serves as the primary input for analysis.
- **System Processing**: The uploaded video is processed within the system. During this stage, relevant features are extracted and analyzed, including facial characteristics, frame sequences, and temporal patterns. The system internally handles video decoding, frame segmentation, and classification through the trained deep learning model.
- **Output**: The system generates a classification result, indicating whether the uploaded video is **real** or **deepfake**. The output also includes the model's **confidence score**, which reflects the reliability of the prediction.

This DFD Level 0 provides a simplified visual representation of the core functionality, highlighting the flow of data from video upload to the final classification outcome.

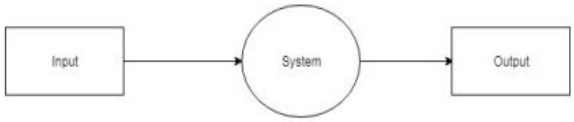


Figure 4.2: DFD Level 0

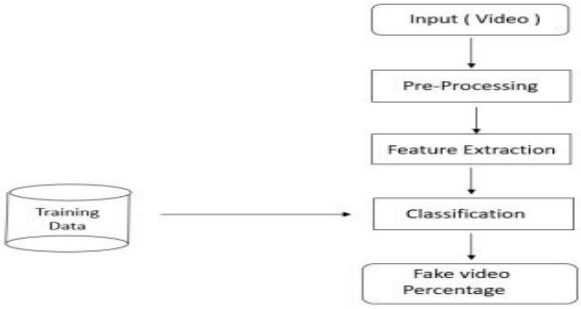


Figure 4.3: DFD Level 1

Data Flow Diagram – Level 1

The Level 1 Data Flow Diagram (DFD) offers a more detailed view of the internal processes of the system, expanding on the basic structure presented in Level 0.

1. **Overview**: DFD Level 1 breaks down the system into sub-processes, providing a clearer understanding of the internal data flow and how various components interact during execution.
2. **System Functionality**: This level outlines the sequential procedures involved in the classification process. It illustrates key stages such as:

- Video upload and validation
- Frame extraction and face detection
- Feature extraction using a pre-trained CNN (ResNext)
- Temporal analysis using LSTM
- Classification and prediction generation

3. **Input/Output Detail**: Each sub-process shows both the inputs it receives and the outputs it generates. For example, the feature extraction process receives face-cropped video frames and produces high-dimensional feature vectors, which are then passed to the LSTM for temporal pattern recognition.

This level of detail helps visualize the logical architecture of the system, ensuring better understanding of the internal mechanisms and data dependencies at each stage.

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806188

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

DFD level-2 enhances the functionality used by user etc.

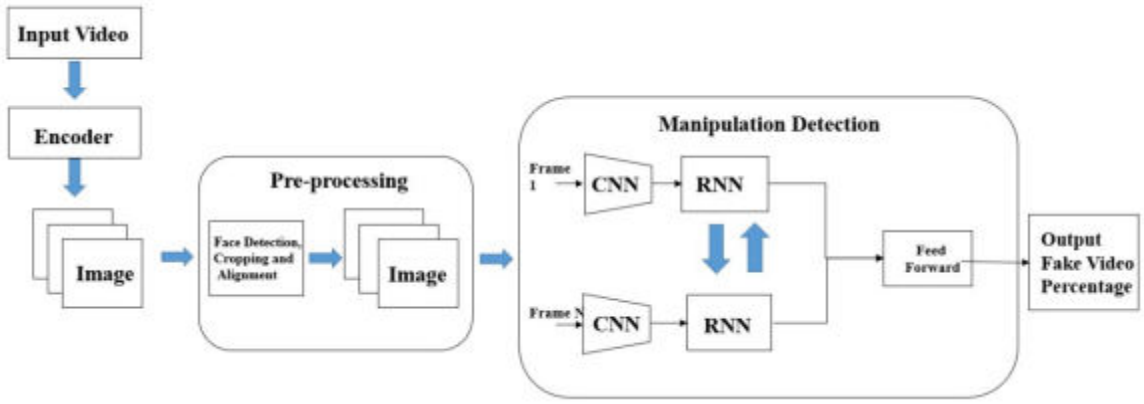


Figure 4.4: DFD Level 2

V. RESULTS AND DISCUSSION

This paper aims to effectively differentiate between authentic and manipulated video content using advanced machine learning approaches. During the testing phase, approximately 150 videos were evaluated, with the system successfully classifying 80 genuine videos as real and accurately detecting 65 as deepfakes. A small subset of five videos posed classification challenges due to either highly refined manipulation techniques or degraded video quality characterized by significant noise and distortion. The detection performance is influenced by factors such as video resolution, clarity, and duration. In some cases, the system's accuracy diminishes when dealing with complex backgrounds or indistinct facial features. Overall, classification accuracy is not consistent across all inputs, as it heavily depends on the alignment between the input characteristics and the distribution of the training data. While the system shows reliable results in most scenarios, further enhancements are necessary to improve resilience against advanced forgery techniques and suboptimal video conditions, thereby increasing its applicability in real-world environments.

Test Case Description	Expected result	Status
Upload a WORD file	Error message-Max limit 100Mb	PASS
Upload a 200 Mb video fle	Error Message- NO Face detected cannot process the video	PASS
Videos with many faces	ERROR	PASS
Deepfake video	FAKE	PASS
Enter/Predict in URL	Redirect to /Upload	PASS
Upload without selecting any file	Alert message-Please select video	PASS
Upload a real video	REAL	PASS
Upload a face cropped real video	REAL	PASS
Upload a face crop fake video	FAKE	PASS

Figure 5.1: Testcase Tabulation

V. CONCLUSION AND FUTURE WORK

In conclusion, we have successfully implemented a neural network–based system capable of classifying videos as either deepfakes or authentic, while also providing a confidence score for each prediction. The proposed model demonstrates strong performance in analyzing and evaluating video content with high accuracy. As the system is

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

designed to continuously learn and adapt, its performance is expected to improve over time with additional data and training.

Looking ahead, there are several avenues for enhancement. For instance, the current web-based interface can be extended into a browser plugin for more seamless and user-friendly access. Additionally, while the present system focuses exclusively on detecting facial deepfakes, future iterations of the algorithm could be expanded to identify full-body manipulations, thereby broadening its applicability in detecting more sophisticated synthetic media.

REFERENCES

[1] T. Vignesh, P. H. Tarun, R. Parthav, and V. Bhargavi, "Deepfake face detection using machine learning with lstm," in 2024 10th International Conference on Communication and Signal Processing (ICCSP). IEEE, 2024, pp. 1633–1638.
[2] M. Westerlund, "The emergence of deepfake technology: A review," Technology innovation management review, vol. 9, no. 11, 2019.
[3] L. M. Dang, S. I. Hassan, S. Im, J. Lee, S. Lee, and H. Moon, "Deep learning based computer generated face identification using convolutional neural network," Applied Sciences, vol. 8, no. 12, p. 2610, 2018.
[4] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," in 2018 IEEE International workshop on information forensics and security (WIFS). Ieee, 2018, pp. 1–7.
[5] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," Interfaces (GUI), vol. 3, no. 1, pp. 80–87, 2019.
[6] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," in 2018 IEEE international workshop on information forensics and security (WIFS). IEEE, 2018, pp. 1–7.
[7] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1–11.
[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in International conference on machine learning. PMLR, 2017, pp. 1126–1135.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com