AI

MDPI

Article

Neutralizing Misleading Visual Cues: An Attention-Driven Encoder–Decoder Framework with Auxiliary Forensic Reconstruction for Video Deepfake Detection

Krity Duhan and Abhishek Kajal *

Department of CSE, Guru Jambheshwar University of Science and Technology, Hisar 125001, Haryana, India; krityduhan7@gmail.com
* Correspondence: drabhishekkajal@gmail.com; Tel.: +91-7056279167

Abstract

As deepfake generation methods are constantly evolving, there is a need for new methods that can detect subtle manipulation artefacts in video content, both in terms of spatial and temporal characteristics. In this study, an encoder–decoder architecture is proposed, where the classification and forensic reconstruction are performed simultaneously at the encoder and decoder side, respectively. The architecture consists of a six-layer convolutional encoder to obtain spatial representations from uniformly sampled facial frames of 128 × 128, and a two-layer bidirectional gated recurrent unit (BiGRU) to model temporal dependency within the sequence of eight facial frames. An attention-pooling mechanism additionally highlights the most instructive temporal characteristics and yields a brief sequence-level representation for binary categorization. In parallel, a forensic decoder reconstructs 64 × 64 Laplacian residual maps, encouraging the network to learn edge-level and texture-level manipulation traces. The framework is trained end-to-end using a weighted combination of cross-entropy classification loss and L1 reconstruction loss. Experiments on the Celeb-DF dataset, comprising 1168 cropped face videos, achieved 93% training accuracy, 90% frame-level classification accuracy, and an ROC-AUC of 0.92. Complementary analysis across spatial, frequency, and temporal feature domains showed that spatial texture characteristics provided the clearest class separation, while the forensic reconstruction branch achieved an MSE of 0.010. Overall, the results demonstrate the value of jointly integrating spatial encoding, temporal sequence modelling, attention-based aggregation, and reconstruction-guided forensic learning for deepfake detection.

Keywords: encoder; decoder; Laplacian residual; deepfake; video detection; reconstruction loss

Check for updates

Academic Editor: Olufemi A. Omitaomu

Received: 27 August 2026
Revised: 8 October 2026
Accepted: 2 November 2026
Published: 4 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The speed at which generative AI models are evolving has made it much easier to create a realistic fake video of real people [1]. With face swapping and face reenactment techniques, manipulated footage can now be produced that is almost a copy of the original video to human eyes [2]. This poses serious threats to personal privacy, political integrity and legal evidence. With the quality of these forgeries improving over time, there is a need for reliable automated detection methods [3]. Although handcrafted features like blinking patterns, head pose inconsistencies, and compression artefacts were significant, these features were not used in conventional methods of deepfake detection. These techniques are effective against first-generation fakes, but are not suitable for newer methods of

fake creation that have learned to remove these telltale features. More recent research has focused on convolutional neural networks (CNNs) and learning the forgery cues directly from each frame. But treating each frame as an individual frame does not take advantage of an important aspect of video—temporal coherence [4]. The pipeline for deepfake generation works on the level of the individual frames, and often introduces subtle flickering, unnatural motion, and the appearance of inconsistencies between time artefacts that can only be observed with analysis of the individual frames [5]. Transformer-based architectures have demonstrated good performance in modelling long-term dependencies in the sequential data, while current transformer-based deepfake detection approaches have either concentrated on spatial features or temporal features separately. Very few approaches try to learn a discriminative representation for classification and a reconstructive representation that reveals the underlying manipulation artefacts [6,7].

This paper proposes an encoder–decoder network specifically tailored for deepfake detection using spatial and temporal information from video frames. The encoder is composed of two components: a convolutional frame encoder to extract spatial features for every frame at 128 × 128 resolution and a bidirectional GRU-based temporal encoder to capture the temporal dependency among a sequence of eight uniformly sampled frames. The temporal information is then pooled in an attention network and fed into a binary classifier to produce a video-level temporal representation. At the same time, a transposed-convolution decoder recovers Laplacian residual maps from the temporal latent space, which must be learned to capture fine-grained edge and texture inconsistencies that allow real faces to be distinguished from synthesized faces. A joint loss function of cross-entropy classification loss and L1 reconstruction loss is used for training the model, with the weights adjusted to give both losses equal importance. The proposed method is tested on the Celeb-DF dataset [8], which consists of cropped face videos, with a 60/20/20 train/validation/test split. The results show that the dual-task design enhances the detection accuracy and the reconstruction component can give informative forensic evidence of the location of the manipulations. The contribution of this work does not arise from introducing an entirely new CNN, recurrent unit, or attention mechanism in isolation. Instead, the novelty lies in integrating spatial encoding, bidirectional temporal modelling, attention-guided sequence aggregation, and auxiliary forensic reconstruction within a single jointly optimized framework. In particular, the temporal representation learned from short eight-frame sequences is used not only for real/fake classification but also to reconstruct Laplacian residual maps that preserve edge and texture-related forensic information. This dual learning goal allows for the latent representation to encode residual structure information for manipulation in addition to discriminative information for classification. The resulting framework therefore differs from conventional frame-level classifiers and purely reconstructive encoder–decoder models by jointly modelling spatial, temporal, and reconstruction-based evidence within the same end-to-end architecture.

The rest of the paper is structured as follows: a literature review is conducted in Section 2, a methodology is proposed in Section 3, results and analysis are given in Section 4, a comparative analysis is shown in Section 5, limitations and future scope of the work are discussed in Section 6, and the paper is concluded in Section 7.

2. Related Works

The rise of synthetic media created with deep learning technology has sparked the urgent need for deepfake detection research to safeguard individual privacy, political integrity, and trust in digital media. In this review, we consider ten recent works that together tackle some of the most pressing questions in the field, such as cross-dataset generalization, robustness to video compression, computational efficiency and forensic

applicability. The detection-oriented super-resolution framework proposed in [9] learns a spatial–temporal dual dictionary to recover forensically valuable information in highly compressed videograms of faces, focusing on expression key sequences and utilizing an attention mechanism based on the joint DCT-DWT, which takes a stand to guide the recovery process; the method achieves decent performance in the face of FaceForensics++ and Celeb-DF without relying on paired compressed–uncompressed videos, but its reliance on handcrafted dictionary construction may make it difficult to scale. In [10], the authors expand the CoAtNet hybrid CNN-transformer framework to 3D by inflating all the convolutional, pooling, and self-attention layers, thereby being able to model spatiotemporal relations in 15-frame video snippets and obtaining good results for the intra-dataset performance and some improvement in cross-dataset generalization with FF++, DFDC, and Celeb-DF, yet the computational cost of 3D layers is not explored in detail. In [11], the authors follow an experimental path and explore the impact of the dataset construction process and the use of the horizontal mirror augmentation on the generated deepfakes, demonstrating a slightly lower value of faceloss in augmented datasets and a difference in the attribute-dependent characteristics of the subjects; however, the scope of the work is narrow and its focus is on the training stage only. In [12], the authors present GenD, a remarkably simple yet highly effective method that fine-tunes only the layer normalization parameters of pre-trained vision encoders such as CLIP, PE, and DINO, amounting to roughly 0.03% of total parameters, and combines cross-entropy with alignment and uniformity losses in an L2-normalized feature space; evaluated across 14 benchmark datasets spanning six years, GenD achieves leading cross-dataset performance and demonstrates that paired real–fake training data is essential for preventing shortcut learning, making it arguably the strongest contribution among the reviewed works. In [13] the authors propose to use a genetic algorithm to compress the ResNet50 features using LASSO regularization, and then use the selected features as input for lightweight classifiers like KNN, SVM, etc., which obtains about 90% compression on Celeb-DF v2 and performs competitively in such a resource-constrained environment with limited evaluation across datasets.

In [14], the forensic framing is new and significant, and generalization tests are conducted on unseen datasets resulting in AUC values ranging from 0.621 to 0.783, leaving much work to be done before this approach can be deployed in real forensic applications; however, the authors linked deepfake detection and forensic science by converting detection scores into calibrated likelihood ratios suitable for courtroom use, selected the Capsule detector as the most viable candidate, and used kernel density estimation with empirical bounds and PAV calibration. In [15] the authors utilize the embeddings from Swin V2 Vision Transformer along with four families of textural descriptors, LBP, Gabor filters, Wavelet transforms, and GLCM features, to get 97.44% accuracy on Celeb-DF V2; however, the absence of cross-dataset evaluation results reduces the validity of the claims. In [16], the authors propose a compression-aware model composed of an augmentation based on the MPEG standard implemented in a variety of compression levels and two architectural improvements: Multi-Head Network for joint deepfake and compression classification, and a Multi-Branch Network for scalable ensembling, which shows that lightweight backbones with multi-head extensions can outperform deeper models based on a single stream. In [17], the authors propose an XceptionCapsule Net that fuses Xception spatial features with Capsule Network representations and BiLSTM temporal modelling alongside BlazeFace and MediaPipe landmarks, reporting up to 99.31% accuracy on FF++, though these exceptionally high numbers warrant caution in the absence of cross-dataset validation. Recent studies highlight the shift towards more practical deepfake detection, emphasizing cross-dataset generalization, unseen forgery detection, compression robust-

ness, computational efficiency, and spatiotemporal and lightweight architectures for reliable real-world deployment [18–20].

When combined, these studies show that the discipline is moving from focusing on within-dataset accuracy towards solving more practical issues such as quality reduction due to compression, generalizing to unknown forms of forgery, forensic acceptability, and demographic fairness, with the adoption of large foundation models being the most promising approach.

3. Proposed Methodology

This section describes the dataset, system configuration and the overall architecture of the proposed encoder–decoder network for deepfake video detection. Our proposed model is trained and evaluated on the Celeb-DF dataset [8], which contains cropped face-only video clips, which are sourced from real celebrity interviews and their corresponding deepfake counterparts. The real subset contains original face sequences while the fake subset contains face-swapped videos generated using the improved synthesis methods. All videos were then divided into subsets of training, validation and test videos with a ratio of 60:20:20 before frame sampling was performed. Hence, the frames in a specific video were limited to one partition and not distributed across the training, validation or test sets. The uniformly sampled eight frames of each video were grouped together as a single temporal input sequence after partitioning. During test-time evaluation, each retained frame was designated as a target frame and evaluated together with its associated eight-frame temporal context. The model therefore produced one classification output for each retained target frame. The eight-frame sequence defines the temporal context used to generate the prediction, whereas the target frame defines the unit of quantitative evaluation. Consequently, the 1290 retained target frames correspond to 1290 classification predictions. This process will ensure no information from the same source video appears in more than one experimental partition. One advantage of the proposed methodology is its balanced design, utilizing both spatial feature extraction and temporal sequence modelling along with attention-based aggregation and forensic reconstruction as part of the detection framework. In comparison with a highly complex 3D convolutional model or a transformer model that needs large-scale datasets, long video clips, or massive computing power to achieve good performance, the proposed model is capable of achieving high performance with only eight frames of a short video while still grasping the meaning of the spatial and temporal manipulation cues. Local variations in facial texture are learned by the convolutional encoder, temporal discrepancies between the frames are captured by the bidirectional GRU, and the attention mechanism selectively focuses on the most informative frames instead of considering all frames uniformly. Further, the forensic decoder reconstructs Laplacian residual maps to uncover artefacts at the edge level and texture level from manipulations. This approach is accurate, more interpretable and also more computationally efficient. Moreover, the accuracy and recall rates of different classes are comparable, suggesting the absence of class bias, and the low reconstruction error shows that the auxiliary forensic branch brings valuable information to the detection phase. These aspects show that the proposed methodology is still applicable in cases where more complex models can overfit, require significant computation or cannot generate interpretable forensic evidence.

The experiments are run on a Windows-based system with an NVIDIA CUDA-enabled GPU. The implementation is mainly developed in Python 3.10 and PyTorch 2.15.0 with OpenCV for decoding the video and applying the processing at the frame level. The model is trained for 100 epochs using the AdamW optimiser with an initial learning rate of 2×10^{-4} and a weight decay of 1×10^{-4}. A cosine-annealing learning-rate scheduler is employed over the complete training run. Mixed-precision training (FP16) is used to reduce memory

consumption and to improve computational efficiency, while gradient clipping is applied with a maximum norm of 1.0. The experimental partitioning is performed at the video level before frame sampling so that all frames originating from a particular video remain within only one of the training, validation, or test subsets. For each video, eight frames are then uniformly sampled and treated jointly as a single temporal sequence rather than as independent image samples. These frames are first processed by the convolutional encoder, after which the BiGRU captures their temporal relationships and the attention mechanism combines the resulting temporal features to generate one real/fake classification decision. The reported classification metrics are calculated from extracted test-frame predictions. Therefore, the quantitative results represent frame-level performance. This evaluation strategy preserves the independence of the held-out test data, prevents frames from the same source video from appearing in both training and testing, and ensures that the reported results reflect the temporal modelling capability of the proposed BiGRU-attention framework.

The schematic diagram of the proposed system is shown in Figure 1. The first step takes a raw input video that is uniformly sampled to 8 frames, each of which is resized to 128 × 128. They are further input into a convolutional frame encoder, then a bidirectional GRU temporal encoder. Attention pooling pools the temporal features into a fixed-length vector that is passed to two parallel branches: a classifier head that generates a real/fake prediction, and a forensic decoder that reconstructs 64 × 64 Laplacian residual maps. The model is optimized together with a combination of cross-entropy classification loss and L1 reconstruction loss, with weights. The cropped facial frames in the Celeb-DF dataset have a spatial resolution of approximately 112 × 112 pixels. The eight sampled frames jointly form a temporal input sequence for the CNN-BiGRU-attention architecture, allowing the model to learn spatial and temporal representations. However, for the quantitative evaluation reported in this study, classification metrics are calculated at the extracted-frame level. Thus, the eight-frame sequence describes the temporal modelling structure of the architecture, whereas the reported accuracy and associated metrics correspond to frame-level predictions. During preprocessing, the sampled frames are resized to 128 × 128 pixels to provide a uniform input size for the convolutional frame encoder.

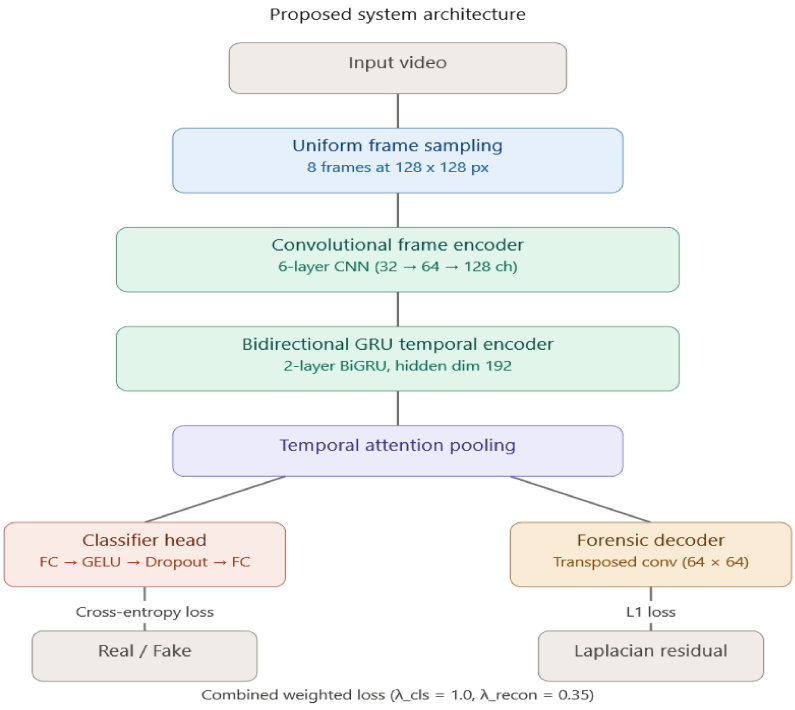


Figure 1. Overall workflow of the proposed methodology.

The proposed baseline can be enhanced to become a more complete and generalizable deepfake video detection framework by adding more spatial, temporal, and forensic learning components. Although previous deepfake detection research has investigated CNN-based, recurrent, transformer-based, and encoder–decoder models, the proposed approach proposes a refinement to enhance the detection accuracy, which considers multi-scale facial artefacts, frame-to-frame inconsistencies, frequency-domain manipulation artefacts, and reconstruction-based evidence. For spatial feature extraction, multiple resolutions can be used to capture both global and local facial distortions, and for temporal modelling, variable-length temporal modelling can be used to analyze short-term flickering as well as longer motion inconsistencies across the video sequences. To be able to detect spectral distortions caused by the deepfake generation and compression processes, a frequency-aware branch can be added as well. Moreover, by incorporating the attention guidance, the forensic decoder can be enhanced to amplify the attention to the manipulation-insensitive sections instead of simply being a black-box classifier. Under realistic settings, these would further boost the reliability of the system by cross-dataset evaluations on Celeb-DF, FaceForensics++ and DFDC, compression-aware augmentation, and uncertainty-based confidence scoring. The methodological improvements are anticipated to boost the accuracy of the deepfake detection system, limit any dataset-specific bias, increase interpretability, and provide more comprehensive guidance for deepfake video detection than binary classification.

Throughout the process of developing this project, three different methods have been considered before finally settling on the proposed encoder–decoder architecture. The first approach was a simple frame-level CNN classifier, in which each frame was classified as either real or fake. It was discarded because each frame was taken into account independently, and the temporal behaviour of videos (flickering, unstable boundaries of face, unnatural movement between frames, etc.) was not taken into account. The second approach was a machine learning classifier based on manually extracted spatial and frequency features like brightness, sharpness, edge strength, FFT energy, and motion features. Though it was helpful in analyzing the data, it was rejected as the final approach because manually crafted features were not good enough to detect complex artefacts that can be present in videos, especially in case they were quite similar to real videos. The third approach was a pure transformer-based video model, which is very powerful theoretically. Nevertheless, it requires much more training data, longer video clips, and more computing resources to train, which were not available. After evaluating all these factors, the final encoder–decoder model was chosen as it provides a better balance between accuracy, efficiency and interpretability. The present model acquires the spatial information in frames and the temporal information in video sequences, and generates forensic residual maps, enabling the model to not only detect deepfakes but also generate forensic evidence of manipulation.

3.1. Data Analysis

Prior to training, we conduct a three-part analysis of the Celeb-DF dataset to understand the statistical properties of both the real and fake videos and to confirm that measurable differences exist between the two classes. We first extract structural metadata from a random sample of 80 videos. For each video v_i, we record the frame count N_i, frame rate f_i (in fps), spatial resolution (W_i, H_i), and duration $D_i = N_i/f_i$ seconds. We compute the per-class mean and standard deviation for each attribute. Let V_r and V_f denote the sets of real and fake videos, respectively. For any attribute a, the class mean is given by

$$\mu_a^{(c)} = \frac{1}{|V_c|}\sum_{v_i \in V_c} a_i,\ \sigma_a^{(c)} = \sqrt{\frac{1}{|V_c|}\sum_{v_i \in V_c}\left(a_i - \mu_a^{(c)}\right)^2} \quad (1)$$

where $c \in \{r,f\}$. The distributions of frame count, fps, and duration are plotted as per-class histograms to identify any structural imbalance. Furthermore, we perform spatial appearance analysis. From a sample of 40 videos, we extract 4 uniformly spaced frames per video and compute three spatial statistics for each frame. Let I denote a grayscale frame. Brightness is defined as $B = \frac{1}{HW}\sum_{x,y} I(x,y)$. Contrast is the standard deviation $C = \sqrt{\frac{1}{HW}\sum_{x,y}(I(x,y) - B)^2}$. Sharpness is measured as the variance of the Laplacian operator $S = \mathrm{Var}(\nabla^2 I)$, where $\nabla^2 I$ is the discrete Laplacian convolution. Higher S values indicate sharper edges; fake frames typically show lower or more erratic sharpness due to synthesis smoothing. Next, we perform the frequency-domain analysis in which we transform each sampled frame into the frequency domain using the 2D discrete Fourier transform $F(u,v) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} I(x,y)\, e^{-j2\pi(ux/H+vy/W)}$. The centred log-magnitude spectrum is $M(u,v) = \log(1 + |F_{\text{shift}}(u,v)|)$. We define low-frequency energy as $E_{\text{low}} = \frac{1}{|R_l|}\sum_{(u,v)\in R_l} M(u,v)$, where $R_l = \{(u,v): r(u,v) \le 0.08 \cdot \min(H,W)\}$, and high-frequency energy as $E_{\text{high}} = \frac{1}{|R_h|}\sum_{(u,v)\in R_h} M(u,v)$, where $R_h = \{(u,v): r(u,v) \ge 0.25 \cdot \min(H,W)\}$. The ratio $\rho = E_{\text{high}}/E_{\text{low}}$ captures the relatively high-frequency content. Deepfake videos tend to suppress high-frequency detail, resulting in a lower ρ compared to real videos. This analysis produces per-class histograms and summary statistics across all the three domains and confirms further that real and fake videos exhibit statistically distinguishable patterns in spatial texture, edge sharpness, and frequency energy distribution.

3.2. Feature Extraction

We extract three categories of features, spatial texture, frequency domain, and temporal motion, from uniformly sampled frames to characterize the differences between real and deepfake videos. From a sample of 30 videos, we extract 6 frames per video for spatial analysis, 4 frames for frequency analysis, and 8 frames for temporal analysis. All frames are resized to 128 × 128 pixels before processing. For each grayscale frame I, we compute five features. Brightness and contrast are defined as before. Edge strength is measured using the Sobel operator in both horizontal and vertical directions as

$$E_s = \frac{1}{HW}\sum_{x,y}\sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{2}$$

where $G_x = I * S_x$ and $G_y = I * S_y$ are the Sobel convolutions with a 3 × 3 kernel. We also compute the Shannon entropy of a 16-bin normalized intensity histogram $\mathbf{h} = [h_1,h_2,\ldots,h_{16}]$ as $H = -\sum_{k=1}^{16} h_k \log(h_k + \epsilon)$, where $\epsilon = 10^{-8}$ prevents numerical instability. Fake frames tend to show reduced edge strength and lower entropy due to the smoothing introduced by synthesis networks. Next, we perform the frequency-domain analysis. Each frame is transformed using the 2D DFT as described in Section 3.4. Here we define three concentric frequency bands using radial distance $r(u,v)$ from the spectrum centre: low ($r \le 10$), mid ($10 < r \le 25$), and high ($r > 25$). The mean energy in each band is computed, along with two diagnostic ratios, as

$$\rho_{\text{hf}} = \frac{E_{\text{high}}}{E_{\text{low}} + \epsilon},\ \rho_{\text{mf}} = \frac{E_{\text{mid}}}{E_{\text{low}} + \epsilon} \tag{3}$$

These ratios capture the relative distribution of spectral energy. Deepfake synthesis tends to attenuate high-frequency components, producing a lower ρ_{hf} compared to real videos. Given a sequence of T consecutive grayscale frames $\{I_1,I_2,\ldots,I_T\}$, we compute inter-frame motion as the mean absolute difference between adjacent frames as $M_t = \frac{1}{HW}\sum_{x,y}|I_{t+1}(x,y) - I_t(x,y)|$. Flicker is quantified as the standard deviation of this difference map. We additionally compute dense optical flow between consecutive frames using the Farneback method, yielding horizontal and vertical flow fields (u_t, v_t). The mean

flow magnitude per pair is defined as $\Phi_t = \frac{1}{HW}\sum_{x,y}\sqrt{u_t(x,y)^2 + v_t(x,y)^2}$. Here, we need to aggregate these per-pair values with respect to the sequence to obtain video-level statistics: motion mean $\bar{M}$, motion standard deviation σ_M, flicker mean $\bar{F}$, flicker standard deviation σ_F, flow magnitude mean $\bar{\Phi}$, and flow magnitude standard deviation σ_Φ. As a result, deepfake videos tend to show more flicker variance and optical flow irregular patterns than other types of videos because of inconsistent frame-by-frame processing. All three feature sets are integrated at the video level into one feature table for subsequent validation and comparison.

3.3. Proposed Architecture

The proposed network, denoted $\mathcal{M}$, takes as input a video tensor $\mathbf{X} \in \mathbb{R}^{B\times T\times C\times H\times W}$ where B is the batch size, $T = 8$ is the number of sampled frames, $C = 3$, and $H = W = 128$. The model produces a classification logit vector $\hat{\mathbf{y}} \in \mathbb{R}^{B\times 2}$ and a set of reconstructed residual maps $\hat{\mathbf{R}} \in \mathbb{R}^{B\times T\times 1\times 64\times 64}$. The complete forward pass is formalized in Algorithm 1.

Algorithm 1. Forward pass of the proposed encoder–decoder network

Require: Video tensor $X \in \mathbb{R}$^{B × T × 3 × H × W}, model parameters Θ
Ensure: Logits $\hat{y}$, reconstructed maps $\hat{R}$, attention weights α
1: Reshape X to $X' \in \mathbb{R}^{(B\cdot T)\times 3\times H\times W}$
▸ *Spatial encoding (convolutional frame encoder)*
2: **for** each convolutional block $k = 1, 2, 3$ **do**
3: $F^{(k)} \leftarrow \sigma(BN(Conv_{k,2}(\sigma(BN(Conv_{k,1}(F^{(k-1)}))))))$
4: **end for**
5: Channel progression: $3 \to 32 \to 64 \to 128$; spatial: $128^2 \to 64^2 \to 32^2 \to 16^2$
6: $V \leftarrow Flatten(AdaptiveAvgPool2d1 \times 1\ (F(3))) \in R(B \cdot T) \times 128$
7: Reshape V to $S \in \mathbb{R}^{B\times T\times 128}$
▸ *Temporal encoding (bidirectional GRU)*
8: $\left[\vec{h}_t, \overleftarrow{h}_t\right]_{t=1}^{T} \leftarrow \text{BiGRU}(S; \text{layers} = 2, \text{hidden} = 192)$
9: **for** $t = 1$ **to** T **do**
10: $h_t \leftarrow W_p[\vec{h}_t \| \overleftarrow{h}_t] + b_p \in \mathbb{R}^{384}$
11: **end for**
▸ *Attention pooling*
12: **for** $t = 1$ **to** T **do**
13: $e_t \leftarrow w\square tanh(W_a h_t + b_a)$
14: **end for**
15: $\alpha_t \leftarrow exp(e_t)/\Sigma_{j=1}^{T} exp\left(e_j\right) forall\ t$
16: $z \leftarrow \Sigma_{t=1}^{T}\alpha_t h_t \in \mathbb{R}^{384}$
▸ *Classification head*
17: $\hat{y} \leftarrow W_2(Dropout_{0.25}(\sigma(W_1 z + b_1))) + b_2 \in \mathbb{R}^2$
▸ *Forensic decoder*
18: **for** $t = 1$ **to** T **do**
19: $q_t \leftarrow W_d h_t + b_d \in R^{(256\cdot 8\cdot 8)}$
20: $d_t \leftarrow Reshape(q_t) \in R^{(256\times 8\times 8)}$
21: **for** each deconv block $j = 1, 2, 3$ **do**
22: $d_t \leftarrow \sigma(BN(ConvTranspose_j(d_t)))$
23: **end for**
24: $\hat{R}_t \leftarrow Sigmoid(d_t) \in \mathbb{R}^{1\times 64\times 64}$
25: **end for**
return $\hat{y}, \hat{R} = [\hat{R}_1, \ldots, \hat{R}_T], \alpha = [\alpha_1, \ldots, \alpha_T]$

Each frame $\mathbf{x}_t \in \mathbb{R}^{3\times 128\times 128}$ is passed through a six-layer convolutional encoder $\mathcal{E}_s$ consisting of three blocks. Each block applies two successive operations: a strided convolution

that halves spatial resolution followed by a same-resolution convolution, both with batch normalization and GELU activation. Formally, for block $k \in \{1, 2, 3\}$,

$$\mathbf{F}_t^{(k)} = \sigma\Big(\mathrm{BN}\Big(\mathrm{Conv}_{k,2}\Big(\sigma\Big(\mathrm{BN}\Big(\mathrm{Conv}_{k,1}\Big(\mathbf{F}_t^{(k-1)}\Big)\Big)\Big)\Big)\Big)\Big) \tag{4}$$

where $\mathbf{F}_t^{(0)} = \mathbf{x}_t$, σ denotes GELU, and the channel progression is $3 \to 32 \to 64 \to 128$. After the final block, adaptive average pooling and flattening yield a feature vector $\mathbf{f}_t = \mathrm{GAP}\big(\mathbf{F}_t^{(3)}\big) \in \mathbb{R}^{128}$. The sequence $\{\mathbf{f}_1,\ldots,\mathbf{f}_T\}$ is reshaped into $\mathbf{S} \in \mathbb{R}^{B\times T\times 128}$ and fed to a two-layer bidirectional GRU. At each timestep t, the forward hidden state $\overrightarrow{\mathbf{h}}_t \in \mathbb{R}^{192}$ and backward hidden state $\overleftarrow{\mathbf{h}}_t \in \mathbb{R}^{192}$ are concatenated and linearly projected as

$$\mathbf{h}_t = \mathbf{W}_p\Big[\overrightarrow{\mathbf{h}}_t \parallel \overleftarrow{\mathbf{h}}_t\Big] + \mathbf{b}_p, \mathbf{h}_t \in \mathbb{R}^{384} \tag{5}$$

Rather than using the final hidden state, we compute a weighted aggregation over all timesteps. An attention score e_t is computed for each step, normalized via softmax to produce weights α_t, and the pooled representation $\mathbf{z}$ is obtained as

$$e_t = \mathbf{w}^\top \tanh(\mathbf{W}_a\mathbf{h}_t + \mathbf{b}_a), \alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^{T}\exp(e_j)}, \mathbf{z} = \sum_{t=1}^{T}\alpha_t\mathbf{h}_t \tag{6}$$

The pooled vector $\mathbf{z}$ passes through a two-layer classifier: $\hat{\mathbf{y}} = \mathbf{W}_2\,\Delta(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2$, where Δ denotes GELU followed by dropout ($p = 0.25$), and the layer dimensions are $384 \to 192 \to 2$. Simultaneously, the full temporal latent sequence $\mathbf{H} = [\mathbf{h}_1,\ldots,\mathbf{h}_T]$ is decoded back into spatial residual maps. Each $\mathbf{h}_t$ is projected to $\mathbb{R}^{256\times 8\times 8}$ via a fully connected layer, then upsampled through three transposed convolution stages ($256 \to 128 \to 64 \to 1$) with stride 2, batch normalization, and GELU, followed by a sigmoid output producing $\hat{\mathbf{R}}_t \in \mathbb{R}^{1\times 64\times 64}$.

The model is trained end-to-end using a weighted combination of classification and reconstruction losses, as described in Algorithm 2, as

$$\mathcal{L} = \lambda_{\mathrm{cls}} \cdot \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{y}}, y) + \lambda_{\mathrm{recon}} \cdot \frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{R}}_t - \mathbf{R}_t\|_1 \tag{7}$$

where $\lambda_{\mathrm{cls}} = 1.0$ and $\lambda_{\mathrm{recon}} = 0.35$. The target maps $\mathbf{R}_t$ are normalized absolute Laplacian responses of the input frames. Training uses AdamW with a learning rate 2×10^{-4}, cosine annealing, gradient clipping at norm 1.0, and mixed precision (FP16) for 100 epochs.

Algorithm 2. Training procedure with joint loss optimization

Require: Training set D_{train}, validation set D_{val}, model M with parameters Θ
Require: Hyperparameters: learning rate $\eta = 2 \times 10^{-4}$ **weight decay** $=1 \times 10^{-4}$ **number of epochs** $E = 100$**, maximum gradient norm** $\gamma_{\max} = 1.0$
1: Initialize Θ with default PyTorch initialisation
2: *optimizer* ← *AdamW*(Θ,*lr* = (2×10^{-4}),*weight_decay* = (1×10^{-4}))
3: scheduler ← CosineAnnealingLR(optimiser, T_{max} = E)
4: scaler ← GradScaler(enabled = True) ▸ *Mixed-precision FP16*
5: L_{best} ← ∞
6: **for** epoch = 1 **to** E **do**
7: Set M to training mode
8: **for** each mini-batch $(X, R, y) \in D_{train}$ **do**
9: optimiser.zero_grad()
▸ *Forward pass under autocast (Algorithm 1)*
10: $\hat{y}, \hat{R}, \alpha \leftarrow M(X)$
▸ *Compute joint loss*
11: $L_{cls} \leftarrow -(1/B)\Sigma_{i=1}^{B}\Sigma_{c=0}^{1} y_{i,c} log\big(exp\big(y_{i,c}\big)/\Sigma_k exp\big(y_{i,k}\big)\big)$

Algorithm 2. *Cont.*

$L_{recon} \leftarrow (1/BT)\Sigma_{i=1}^{B}\Sigma_{t=1}^{T}\|\hat{R}_{i,t} - R_{i,t}\|_1$
$L \leftarrow \lambda_{cls} \cdot L_{cls} + \lambda_{recon} \cdot L_{recon}$
▸ *Backward pass with gradient scaling*
scaler.scale(L).backward()
scaler.unscale_(optimiser)
clip_grad_norm_(Θ, γ_{max})
scaler.step(optimiser)
scaler.update()
end for
scheduler.step()
▸ *Validation*
Set M to evaluation mode
L_{val} $\leftarrow$ Evaluate(M, D_{val})
if L_{val} $<$ L_{best} **then**
L_{best} $\leftarrow$ L_{val}
Save Θ as best checkpoint
end if
end for
return Trained parameters Θ*

Although the proposed framework uses video as the primary input source, the proposed framework can be understood as a multi-representation learning approach because it combines complementary spatial, temporal, and reconstruction-based representations derived from the same video input. It combines multiple complementary information streams extracted from the same video: spatial appearance, temporal behaviour, and forensic residual evidence. Learning of the spatial attribute takes place via the convolutional frame encoder that encodes texture of the frames, their edges, the smoothness of the face in the frames, and local manipulations artefacts. The reason is that the majority of deepfake artefacts emerge from unnatural blending, reduction in sharpness, or deformation of the texture of the faces in the frames. Here, one element of the batch corresponds to one video represented by $T = 8$ uniformly sampled frames. Although the convolutional encoder processes the frames individually, their resulting representations are recombined into a temporal sequence before BiGRU processing. Attention pooling then generates one representation per input video sequence, from which one binary classification decision is obtained.

The second stream, which can be considered modality-like, is temporal attribute. In view of the fact that deepfake videos include not only manipulated images, but manipulated sequences, the model makes use of a bidirectional GRU to learn about facial evolution through the frames. The attention module further strengthens this cross-frame modelling by assigning greater importance to the frames that contain stronger manipulation evidence, rather than treating all frames equally. The paper describes this design as a six-layer CNN frame encoder followed by a two-layer BiGRU and temporal attention pooling.

The third component is the forensic reconstruction branch. Unlike a normal classifier that only predicts real or fake, the decoder reconstructs Laplacian residual maps, which represent edge-level and texture-level manipulation traces. This adds an interpretable forensic signal to the classification process. The use of attention with gated recurrent modelling is also supported by recent sequence-learning studies, where attention-guided GRU architectures have been shown to emphasize informative temporal patterns while retaining a comparatively lightweight recurrent structure [21]. Although developed for multivariate sensor sequences, this principle supports the present use of attention over BiGRU features to assign greater importance to frames carrying stronger temporal evidence.

The reconstructed Laplacian residual maps are intended to provide auxiliary forensic cues by highlighting edge-level and texture-level irregularities learned during reconstruction. The low reconstruction error indicates that the decoder is able to recover these residual structures consistently; however, the maps should not be interpreted as exact manipulation-localization masks because pixel-level ground-truth annotations are not available in the present dataset. Accordingly, the reconstruction branch is used as supporting forensic evidence rather than as a definitive localization mechanism. A direct spatial comparison with annotated manipulation regions will be considered in future work.

Therefore, the work is cross-model because it integrates CNN-based spatial learning, recurrent temporal modelling, attention-based aggregation, and decoder-based reconstruction in one end-to-end system. It is also a multi-representation learning approach in a broader representation-level sense because it fuses spatial, temporal, and residual forensic evidence, even though all are derived from video rather than separate sources such as audio or text. This makes the framework more informative than a single-frame or black-box classifier.

3.4. Evaluation Metrics

The following nine metrics were used to evaluate the suggested model as they provide evaluation for classification performance, prediction error and the quality of the reconstruction process. Accuracy reflects the ratio between the amount of correct classifications in the entire test dataset. Precision reflects the fraction of the predicted fake frames among all that were indeed fake and is relevant when false accusation carries great costs. Recall reflects the ratio between the number of real fake frames which were classified as fake by the model and is an indicator of the sensitivity of the model to deepfake videos. The F1-score is the harmonic mean of the precision and recall, providing a combined score in case these metrics are not in line with each other. The ROC-AUC stands for the area under the curve on the ROC graph, which represents the dependence of the true-positive rate (TPR) on the false-positive rate (FPR) at different levels of classification threshold. A 0.5 value means normal guessing and 1.0 means perfect separation. The mean squared error and root mean squared error (its square root) put higher penalty on larger errors. Mean absolute error provides a linear measure of average prediction error. Finally, reconstruction MSE evaluates the decoder branch by computing the mean squared difference between the predicted and target Laplacian residual maps, $\mathrm{MSE}_{\mathrm{recon}} = \frac{1}{NTH'W'} \sum_{i,t,x,y} (\hat{R}_{i,t}(x,y) - R_{i,t}(x,y))^2$, where $H' = W' = 64$. A small reconstruction mean squared error signifies that the model has been able to learn the edge structure of the input frame, thus affirming that the decoder branch plays an important role in forensics information for the detection task.

4. Results and Analysis

Table 1 describes the comparative analysis of the real and fake videos in terms of their structural metadata, spatial appearance, and frequency-domain characteristics based on the Celeb-DF dataset.

It can be seen from the structural metadata that both types of videos are almost similar in terms of the number of frames, frame rate, and duration, which means that the Celeb-DF dataset is well controlled, and there is no simple way to distinguish the real and fake videos using metadata. The videos have the resolution of 112 × 112 pixels because the dataset includes only the face of the person. There are clear differences shown by the spatial appearance features between the two classes. The brightness of the real videos is higher (108.11 vs. 95.66) and sharpness is higher as well (69.95 vs. 41.98), which means that there is some smoothing and tonal suppression due to the deepfake synthesis process. This comes from a higher sharpness standard deviation for real videos in comparison to fake videos (92.27 vs. 22.67), indicating that natural videos include a broader distribution of

sharpness compared to fake videos that are more uniformly blurred by the generation pipeline. For fakes, contrast values are comparable but slightly lower, in line with the smoothing observation. This is further supported by Figures 2 and 3 respectively.

Table 1. Comparative feature analysis of real and fake video classes in the Celeb-DF dataset.

Feature	Real (Mean ± Std)	Fake (Mean ± Std)
Frame count	147.70 ± 2.06	147.29 ± 4.12
FPS	30.00 ± 0.00	30.00 ± 0.00
Duration (sec)	4.92 ± 0.07	4.91 ± 0.14
Resolution (W × H)	112 × 112	112 × 112
Brightness	108.11 ± 14.00	95.66 ± 20.13
Contrast	35.92 ± 9.72	32.88 ± 10.34
Sharpness	69.95 ± 92.27	41.98 ± 22.67
Low freq. energy	3.69 ± 0.25	3.56 ± 0.26
High freq. energy	0.40 ± 0.05	0.40 ± 0.07
HF/LF ratio	0.109 ± 0.012	0.112 ± 0.015

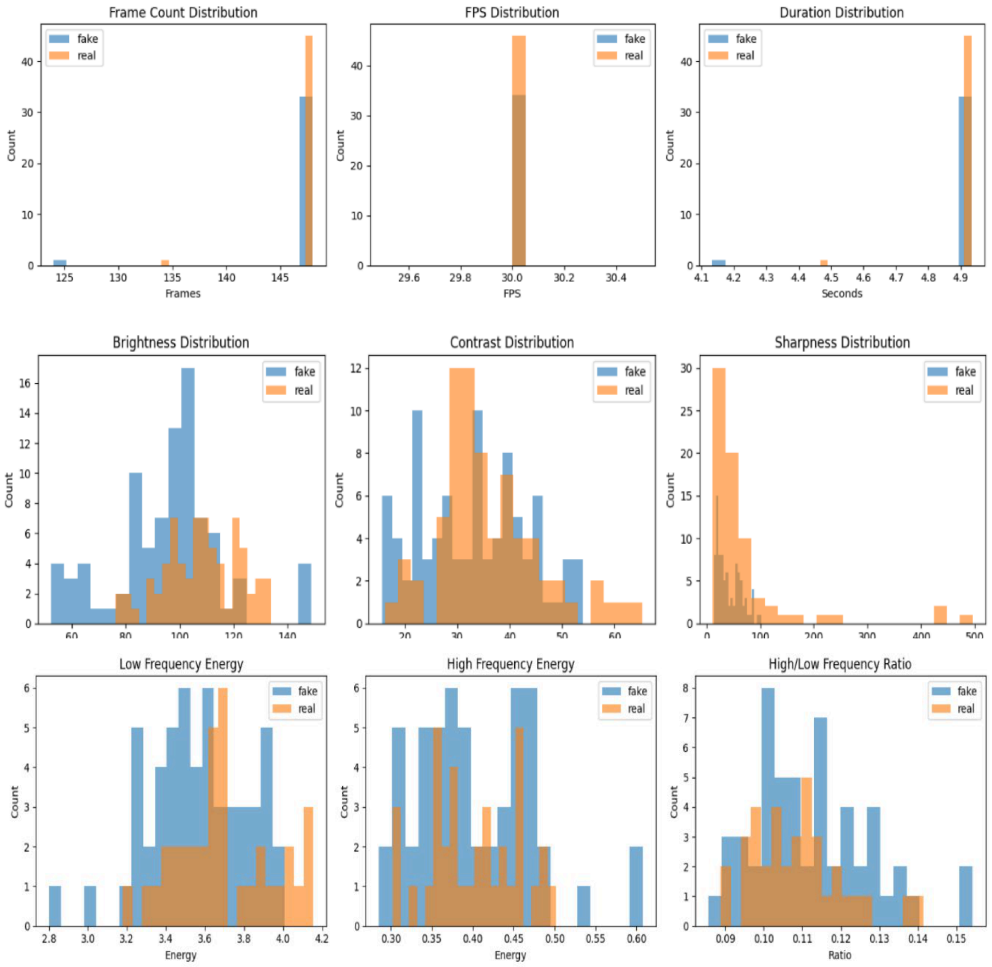


Figure 2. Distribution of structural metadata and spatial appearance characteristics for real and fake videos in the Celeb-DF dataset.



Figure 3. Representative real and fake video frames with their corresponding two-dimensional Fast Fourier Transform (2D FFT) log-magnitude spectra.

The differences are much more subtle in the frequency domain. Low-frequency energy is slightly higher for real videos than for the fake videos, whereas high-frequency energy is closely matched for both types of classes (3.69 vs. 3.56 and 0.40 vs. 0.40, respectively). The difference between the two classes in Celeb-DF, using our HF/LF ratio, is extremely small, 0.109 vs. 0.112, and so cannot be used to separate the two groups with high reliability using features of their frequencies alone. This finding is important as it confirms the architectural decision of not using spectral analysis alone, and proves that it is a more effective approach to combine spatial encoding and temporal sequence modelling, which is the reason why many of the frequency-only detection methods have been reported to be less successful on this particular dataset.

Table 2 and Figure 4 presents the aggregated feature statistics comparing real and fake videos across spatial, frequency, and temporal domains. Table 3 shows a sample of the merged video-level feature table containing all 19 extracted features per video.

The spatial features show the most apparent difference between the two classes. Real videos are consistently brighter (103.23 vs. 90.92) and sharper (20.12 vs. 15.70), and carry stronger edges (31.82 vs. 29.31) than their fake counterparts. This gap is a direct result of the face synthesis process that tends to smooth out the fine details and slightly reduce the intensities of the pixels when they are generated. Also, the difference in sharpness between the real videos and the fake ones (14.95 vs. 7.62) further shows that there is a higher variance in the sharpness of real videos indicating a wider spread of edge detail, whereas the variance for fake videos is limited to a narrower and more uniform range by the generator. Even though the difference is not very large, it can be detected in the frequency domain. The energy in real videos is slightly higher in all three frequency bands with the largest relative gap in high frequencies (0.381 vs. 0.345). The HF/LF and MF/LF ratios, however, are very similar in both classes, indicating that the shape of the spectrum is relatively well preserved in the synthesis process and that frequency features alone would not be very useful in distinguishing between the two classes.

The separation of temporal features is the least. Motion, flicker and optical flow values are very similar, with fake videos actually having slightly higher flow magnitude (6.66 vs. 6.25). This near parity indicates that current deepfake creation technology is reasonably at the level of dealing with the inter-frame consistency, and the temporal artefacts are not so prominent that they can be effectively captured by handcrafted statistics, and rather require learned representations. The combined feature table (Table 3) is a 30 × 19 matrix, which contains all 16 features along with complete extracted representation for every sampled video.

Table 2. Mean ± standard deviation of extracted features for real and fake video classes.

Feature Domain	Feature	Real (Mean ± Std)	Fake (Mean ± Std)
Spatial	Brightness	103.23 ± 19.61	90.92 ± 15.32
Spatial	Contrast	36.40 ± 12.10	32.95 ± 7.75
Spatial	Sharpness	20.12 ± 14.95	15.70 ± 7.62
Spatial	Edge strength	31.82 ± 8.36	29.31 ± 7.99
Spatial	Entropy	2.031 ± 0.269	1.976 ± 0.186
Frequency	Low-freq. energy	3.702 ± 0.332	3.573 ± 0.230
Frequency	Mid-freq. energy	1.701 ± 0.338	1.626 ± 0.227
Frequency	High-freq. energy	0.381 ± 0.104	0.345 ± 0.070
Frequency	HF/LF ratio	0.102 ± 0.023	0.096 ± 0.015
Frequency	MF/LF ratio	0.456 ± 0.062	0.453 ± 0.039
Temporal	Motion mean	21.47 ± 6.18	20.79 ± 5.18
Temporal	Motion std	6.68 ± 4.79	5.77 ± 2.75
Temporal	Flicker mean	21.14 ± 4.85	19.69 ± 4.68
Temporal	Flicker std	5.16 ± 2.99	4.90 ± 3.24
Temporal	Flow magnitude mean	6.25 ± 2.04	6.66 ± 1.34
Temporal	Flow magnitude std	2.06 ± 0.91	2.13 ± 1.10

Table 3. Sample of the merged video-level feature table (first 5 entries).

Video	Class	Bright.	Contr.	Sharp.	Edge	Entropy	LF	MF	HF	HF/LF	Motion	Flicker	Flow
V1	Fake	81.46	28.85	13.10	28.21	1.907	3.533	1.543	0.302	0.085	20.37	17.42	7.61
V2	Fake	82.45	31.50	12.70	25.08	1.965	3.453	1.553	0.303	0.088	18.94	18.35	6.94
V3	Fake	117.47	25.36	15.48	26.61	1.852	3.477	1.564	0.331	0.095	18.43	16.87	7.83
V4	Fake	110.00	31.43	18.61	32.04	2.021	3.586	1.637	0.359	0.100	19.79	18.41	7.47
V5	Fake	101.86	36.71	25.44	33.99	2.031	3.738	1.894	0.408	0.109	20.86	22.46	6.30

Figures 5 and 6 provide evidence about both the convergence behaviour and computational stability of the proposed model. In Figure 5, the training and validation losses decrease consistently across the 100 epochs, from approximately 0.75 initially to below 0.30, while the relatively small separation between the two curves indicates that optimization improvements on the training data are also reflected in validation performance rather than being accompanied by a progressively widening generalization gap. The simultaneous decrease in the classification and reconstruction losses proves that the joint objective does not degrade the learning of one task for the benefit of the other; in particular, the reconstruction loss decreases from around 0.12 to 0.04, indicating progressive learning of the Laplacian residual structure. It is obvious from the accuracy curves that the training accuracy improves from about 55% to 93%, and so does the accuracy when validating the model. The initial epochs show larger fluctuations, suggesting a period of adaptation to the task, followed by the subsequent epochs, which show closer and smoother convergence (approx after 40 epochs), suggesting a more stable convergence region. Finally, as indicated in Figure 6, this convergence is obtained without any increase in the amount of computation required per epoch. Epoch time is initially variable and becomes ~163 s but then settles to ~152–156 s in latter training and keeps its value constant throughout the training, giving a total training time of ~4.3 h. The combined curves suggest a stable optimization process and model behaviour that is fairly consistent as it gets closer to convergence.

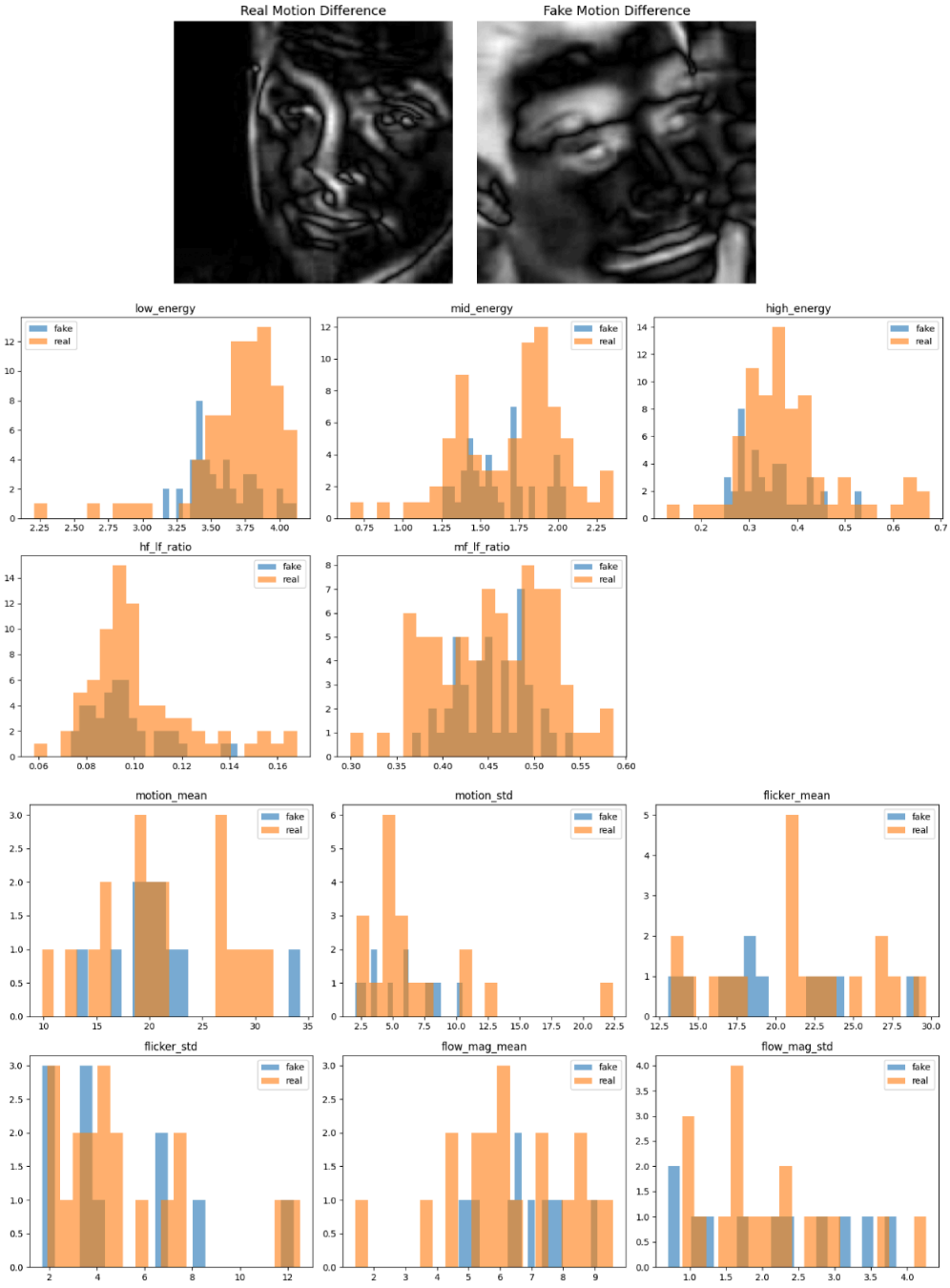


Figure 4. Schematic representation for the extracted features from the dataset.

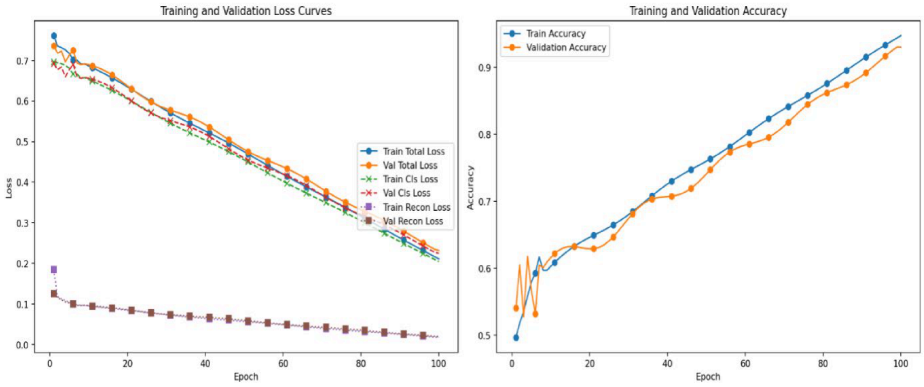


Figure 5. Training vs. validation accuracy and loss plots.

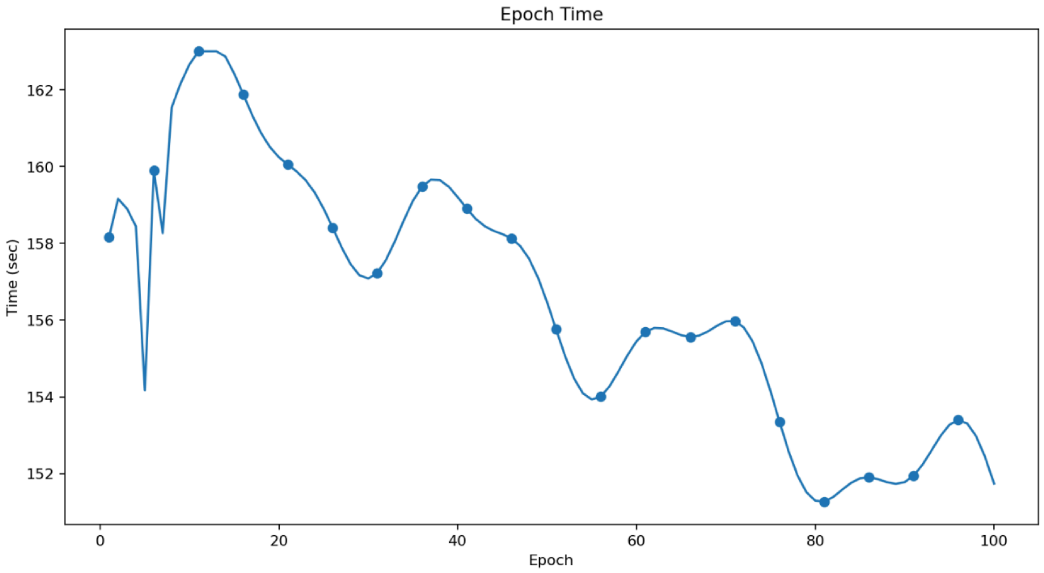


Figure 6. Time taken per epoch during model training.

The classification results in Table 4 were calculated on the held-out test partition using 1290 extracted test frames. Each prediction was obtained only after temporal encoding and attention pooling across the sampled frames. Accordingly, the reported accuracy, precision, recall, specificity, F1-score, FPR, FNR and ROC-AUC represent frame-level classification performance. The source videos were partitioned into training, validation and test subsets before frame extraction, ensuring that frames originating from a given source video remained within a single partition. The metrics used for evaluation are reported in Table 4 on the held-out test set of 1290 video frames and are displayed in the confusion matrix (Figure 7). The model attains a test accuracy of 90% and an ROC-AUC of 0.92, which indicates good discriminative performance on the evaluated real and fake test frames. The precision and recall scores are 0.901 and 0.897, respectively, with an F1-score of 0.899, which indicates that precision and recall are well balanced. The near equality indicates that the model is not biased in identifying too many real videos as fake or letting fake videos through. The prediction error statistics are also low as well, with MSE and MAE equal to 0.10, and RMSE equal to 0.316. The MSE of reconstruction is 0.010, which proves that the reconstruction of the forensic decoder branch is successful in reconstructing the edge structure of the input frames, which is useful spatial information for the whole detection pipeline. The detailed confusion matrix is shown as Figure 7 which displays the test set predictions. Test-set frame extraction and filtering: The 1168 source videos were partitioned at the source-video level into 700 training videos, 233 validation videos, and 235 test videos, corresponding approximately to a 60/20/20 split. This partitioning was performed before frame extraction so that frames originating from a given source video remained within the same experimental partition. From the 235 held-out test videos, eight frames were uniformly sampled from each video, yielding an initial pool of 1880 candidate test frames. This initial pool was subsequently subjected to the predefined preprocessing and frame evaluation requirements. Frames that did not satisfy the required feature extraction and preprocessing criteria and redundant and non-informative frames were excluded. After this, 1290 valid frame-level evaluation samples were retained, comprising 650 real and 640 fake frames. The reported confusion matrix and frame-level classification metrics were calculated exclusively using these 1290 retained samples. Therefore, the transition from 1880 candidate frames to 1290 evaluation samples represents a subsequent preprocessing and sample-validation stage and does not constitute a change to the underlying 235-video test partition. Because the present evaluation is performed at the frame level, the 235 source

videos are not used as the direct unit of the confusion matrix. Instead, the confusion matrix and classification metrics are calculated from the 1290 retained frame-level evaluation samples. The model correctly classifies 587 of the 650 real images/frames as real, while 63 real frames are incorrectly classified as fake. Similarly, it correctly identifies 574 of the 640 fake frames as fake, whereas 66 fake frames are incorrectly classified as real. Among the two types of classification errors, the 66 false-negative predictions are particularly important, as they represent manipulated frames that remain undetected by the model. In comparison, the model produces 63 false-positive predictions, where genuine frames are incorrectly flagged as fake. The false-negative and false-positive rates are approximately 10.31% and 9.69%, respectively. The model is calculated with test accuracy of 90.0%, with the correct classification of 1161 samples out of 1290 test frames, yielding the misclassification rate of 10.0% overall. Balanced classification behaviours across the real and fake classes are represented by approximately similar numbers of false-negative and false-positive predictions. This indicates that the model does not strongly favour either class during prediction.

Table 4. Frame-level classification and reconstruction performance on the held-out test set. Classification metrics are calculated from 1290 retained target frames, each evaluated within its associated eight-frame temporal context. Reconstruction MSE is computed over the corresponding 64 × 64 Laplacian residual maps.

Metric	Value
Test Accuracy	90.00%
Precision	0.901
Recall (Sensitivity)	0.897
Specificity	0.903
F1-score	0.899
False-Positive Rate (FPR)	0.097
False-Negative Rate (FNR)	0.103
ROC-AUC	0.920
Reconstruction MSE	0.010

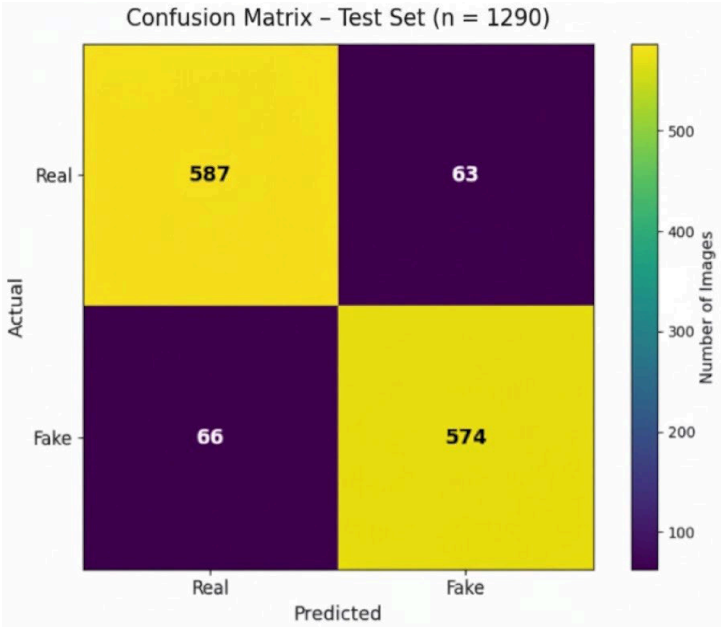


Figure 7. Confusion matrix plot.

The present evaluation in Figures 7 and 8 reports classification performance at the available frame level, whereas the proposed architecture itself models temporal information across multiple frames using the BiGRU and attention mechanism, thus making the eight-frame temporal evaluation the primary evaluation protocol. The model extracts spatial, frequency and temporal features from the eight-frame input sequence during feature representation and decides whether the observed content in the test frame is associated with a real or manipulated one. The temporal analysis supports the learning of richer representations during training that cannot be obtained from a single image, such as frame-to-frame inconsistency, unnatural facial motion, flickering and unstable textures. Video-level evaluation requires grouping predictions according to their source videos and producing a single probability or decision for each complete video sequence. Since such aggregation was not included in the current evaluation pipeline, the reported accuracy, precision, recall, specificity, F1-score, FPR, FNR, and ROC-AUC should be interpreted as frame-level results not to be compared with video-level evaluation results.

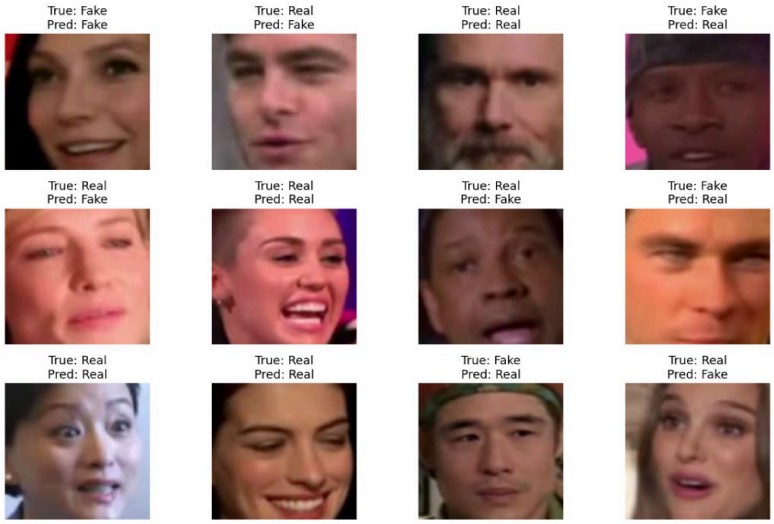


Figure 8. Representative correctly and incorrectly classified test frames. The displayed frames are representative individual samples from the held-out test set, and each classification decision corresponds to an individual extracted frame.

5. Comparative Analysis

Table 5 compares the proposed encoder–decoder network with seven recently published deepfake detection methods all tested on the Celeb-DF dataset. The results presented in Table 5 are literature results and should be considered a literature-based comparison and not a controlled comparison under identical experimental conditions. Differences in train–test partitioning, frame sampling, preprocessing, input sequence length, and evaluation protocol may influence the reported performance. A strictly controlled comparison would require re-implementation and re-training of the competing architectures under the same Celeb-DF partition and preprocessing pipeline used in this study.

Table 5. Comparative analysis of proposed work with others.

Ref.	Method	Architecture	Temporal Modelling	Interpretability	Dataset	ACC (%)	AUC
[2]	YOLOv5 Binary Classifier	YOLOv5 backbone + custom detection head	No (frame-level)	No	Celeb-DF	90.00	—
[6]	ResNet50 + Bi-LSTM	Pre-trained ResNet50 + Bi-LSTM	Yes (sequential frames)	No	Celeb-DF v2	75.60	—

Table 5. *Cont.*

Ref.	Method	Architecture	Temporal Modelling	Interpretability	Dataset	ACC (%)	AUC
[22]	Eye Blink + Nose Landmark (MLP)	Facial landmark extraction + SVM/MLP/LogReg	No (per-frame features)	Partial (landmarks)	Celeb-DF	75.00	0.970
[23]	TimeSformer (best config)	Transformer (joint spatiotemporal attention)	Yes (96-frame clips)	No	Celeb-DF	78.40	0.801
[24]	Deep Conv. Pooling Transformer	CNN + transformer (re-attention, conv. pooling)	No (frame-level)	No	Celeb-DF	63.27	0.724
[25]	Swin-Unet Spatial Training	Swin-Unet + self-blending without fake samples	No (frame-level)	Yes (localisation maps)	Celeb-DF	87.52	—
[26]	GNN + CNN Fusion (FuNet)	Mini-batch GCN + 4-block CNN with fusion	No (frame-level)	No	Celeb-DF	98.90	0.980
Ours	Encoder–Decoder + BiGRU + Forensic Decoder	6-layer CNN + BiGRU + attention pooling + transposed conv decoder	Yes (8-frame sequence)	Yes (Laplacian residual maps)	Celeb-DF	90.00	0.920

The comparative results in Table 5 demonstrate that the proposed encoder–decoder framework is more balanced and practically meaningful when compared to the existing deepfake detection approaches based on Celeb-DF. There are existing systems that achieve detection by analyzing each frame separately as an image, making them less capable of detecting the temporal aspect of video manipulation. Examples of this include the Deep Convolutional Pooling Transformer in [24], which uses convolutional and transformer-based operations, but has been largely frame-oriented, with no explicit consideration of inter-frame motion or temporal inconsistency. Consequently, it suffers significantly on Celeb-DF, showing that mere spatial modelling is not enough for robust video-level deepfake detection. Similarly, the YOLOv5-based binary classifier in [2] can be trained for object detection using a strong object detection backbone, but it uses visual characteristics at the frame level rather than forensic patterns in the sequence of frames. This approach is effective in detecting facial characteristics that have been modified; however, it may not be able to detect flickering, instability of the texture, and even inconsistencies in the motion across consecutive frames, which are all important evidence of video manipulation. Although the method of GNN-CNN fusion in [26] performs best among the reviewed methods in terms of accuracy and AUC, this finding should be taken with caution because of the need for considerably more training configurations and computing power. The model is a graph neural network fusion with CNN features, which makes the architecture complicated and might not be applicable to small datasets or resource-constrained settings. Furthermore, the method is numerically very good but does not offer any temporal modelling or clear forensic interpretability. This means that [26] is better in terms of reported accuracy; however, the proposed approach provides a better trade-off between accuracy, computational viability, temporal learning and interpretability. This separation is crucial, as it is important that a video can be accurately classified by a deepfake detection system and the system can also give some reason as to why a video is a deepfake.

Temporal modelling approaches such as ResNet50 + Bi-LSTM [6] and TimeSformer [23] are closely related to the proposed work because they recognize an important fact: deepfake detection is not only a frame-level classification task, but a video-level forensic problem. Facial manipulations often appear across time as inconsistent blinking, unstable texture, unnatural mouth movement, irregular head-pose transitions, temporal flickering,

or frame-to-frame identity drift. Therefore, models that explicitly process sequential information are theoretically better aligned with the nature of deepfake videos than methods that classify frames independently.

However, the ResNet50 + Bi-LSTM [6] strategy has an important limitation. In this design, ResNet50 is responsible for extracting spatial features from each frame, while the Bi-LSTM learns temporal dependencies across the extracted frame-level representations. This means the quality of the temporal model is heavily dependent on the quality of the CNN backbone. If ResNet50 is not sufficiently fine-tuned for the target dataset, or if it fails to capture dataset-specific manipulation artefacts, the Bi-LSTM receives weak or incomplete features. In that case, the temporal layer can only model the sequence of already limited representations. This partly explains why such methods may show lower performance on challenging datasets such as Celeb-DF v2, where manipulations are visually more realistic and artefacts are less obvious than in earlier deepfake datasets. Thus, although ResNet50 + Bi-LSTM introduces temporal modelling, its effectiveness remains constrained by the adaptability of the spatial encoder.

The TimeSformer [23] is an advanced approach in temporal modelling, which employs transformer-based self-attention on spatial patches and temporal frames. Unlike CNN-recurrent architectures in which spatial features are extracted first, and then fed to temporal part of the network, TimeSformer attempts to learn spatial–temporal relationships in a single pass. As a result, the model can make comparison between faces at different times and learn long-range temporal dependencies. This is, in principle, a very efficient architecture: deepfakes can have inconsistencies in not only between consecutive frames, but also between consecutive frames that are not physically close together. Yet, the results of TimeSformer also demonstrate another significant drawback. Even with longer sequences (96 frames), TimeSformer performance is still not high enough compared with our approach. Thus, simply increasing the number of frames or making the architecture more complicated does not always help. Transformer-based architectures need large-scale training data, a high level of regularization, and heavy computational resources to learn stable representations. If the amount of data is small or manipulations are subtle, such architectures can easily overfit to the training set, learn redundant frame-level patterns, or simply ignore temporally consistent manipulation-specific cues. Therefore, TimeSformer's underperformance indicates that deepfake detection benefits not only from temporal depth, but from efficient temporal selection and compact sequence modelling.

In contrast, the introduced architecture manages to perform well at detecting manipulation traces by taking into consideration just eight frames per each video and yet achieving an accuracy of 90%. This is crucial since it proves that an architecture of temporal encoder allows learning relevant sequential information without having to work with long videos or a large architecture. In contrast to working with a lot of redundant frames, the proposed architecture works to extract meaningful information out of the chosen frames. This task is performed with the help of a BiGRU module which is responsible for recognizing both forward and backward temporal dependencies in order to learn about manipulation traces' evolution across the selected frames. The attention mechanism further enhances this process by giving more weight to frames with stronger forensic information.

Interpretability is yet another important aspect of the proposed approach. While there are many approaches out there, including [2,6,23,24,26], which work as black-box classifiers, they have the problem of being able to identify whether the video is genuine or fake but not explaining why certain visual regions or temporal clues were used for this decision. This poses a significant challenge in the context of forensic and practical applications, where the algorithm has to be capable of not just outputting a classification but also giving an interpretation behind it. On the other hand, the approach of [22] based

on handcrafted landmarks is partially interpretable as it relies on facial clues like eye blinks and nose features. Nevertheless, the manual selection of landmarks limits its capability of identifying more complex manipulation artefacts related to textures, illumination, blending boundaries, or inconsistencies at higher frequency levels. Similarly, the Swin-Unet-based method in [25] improves interpretability by generating localisation maps, focusing mainly on spatial frame-level evidence, and does not model the temporal evolution across the video frames.

Another crucial aspect of the GNN-based method [26] lies in its impressive numerical performance, 98.9% accuracy, and 0.98 AUC score. The application of graph neural networks can be considered beneficial for deepfake detection since it allows modelling relations between certain regions or landmarks on the face as well as between feature nodes. Contrary to the traditional approach that sees the face as a mere rectangular image, in the case of GNN, a face can be modelled as a graph with nodes standing for facial parts and edges representing their relations. Such an approach enables the algorithm to recognize structural inconsistencies such as lack of natural coordination of eyes, mouth, nose, and boundaries of the face. The ability to model relations can prove effective in cases when manipulations disrupt face consistency. However, even though the numerical performance of the approach is impressive, it does not always translate into better suitability for certain forensic applications. In addition, many GNN approaches provide limited temporal explanation unless the graph is explicitly extended across frames.

Through filling the gaps mentioned above, the suggested method takes into account the temporal relationship and forensic reconstruction. The BiGRU models the temporal relationship among video frames, the attention mechanism identifies the informative temporal subsequences, and the decoder produces Laplacian residual maps that play an important role in identifying manipulation areas. This indicates that the model not only classifies whether a video is real or fake, but it can also produce interpretable forensic results about potential manipulation traces. As compared to TimeSformer [23], the proposed approach is more compact and computationally efficient as it does not require long sequences of 96 frames and expensive transformer computations. As compared to the GNN-based approach [26], it can produce lower raw accuracy, but it will be more temporally aware and generate reconstruction-based forensic results. In conclusion, the proposed encoder–decoder framework approach is an efficient solution that incorporates all three requirements of detection, temporal awareness, and visual explainability.

In general, the robustness of the proposed approach is not only from the accuracy but also from the ability to balance the classification performance, temporal awareness, interpretability, and computational practicability. It is more suitable when performing manipulation detection in a video-based way, as it models sequential dependencies when compared to frame-level methods. It is more efficient than heavier transformer or graph-based models and is less reliant on large-scale computation resources. It enhances the detection performance compared to landmark localization or only localization and also gives forensic explanation by residual reconstruction. Additionally, the model's balanced precision and recall suggest that it does not favourably treat any class (truth or falsity) and the high AUC value reflects good separability between real and fake videos.

Furthermore, the comparative results should be interpreted within the Celeb-DF evaluation setting. Within this dataset, the proposed model provides a useful combination of classification performance, temporal modelling, and reconstruction-based forensic information. The BiGRU and attention mechanism allow temporal information to be incorporated across sampled frames, while the decoder provides an additional residual representation that is not available in conventional frame-level classifiers. At the same time, the present results should not be interpreted as evidence of cross-dataset robustness. Deepfake datasets

differ in manipulation methods, compression levels, video quality, identities, and preprocessing, and these differences may affect model performance. Therefore, the current experiments demonstrate the effectiveness of the proposed architecture on Celeb-DF, while its behaviour on independently collected datasets remains to be established through cross-dataset evaluation.

Hence, the proposed encoder–decoder scheme is a practically feasible and theoretically meaningful approach for detecting a deepfake video especially when accuracy, efficiency and interpretability are taken into account simultaneously.

6. Limitations and Future Scope of the Study

A principal limitation of the present study is that both model development and evaluation are based on the Celeb-DF dataset. Although the train, validation, and test subsets are separated, they originate from the same dataset and therefore share its acquisition conditions, manipulation procedures, compression characteristics, and preprocessing. Consequently, the reported accuracy and ROC-AUC describe within-dataset performance and should not be interpreted as evidence of generalization to other deepfake benchmarks or real-world video sources. Cross-dataset testing on benchmarks such as FaceForensics++ and DFDC is therefore necessary before broader claims regarding robustness or practical deployment can be made. Future work will evaluate the trained framework under these settings and examine how performance changes across manipulation types and video-quality conditions. Our proposed model is trained only on the Celeb-DF dataset that consists of the cropped face videos of fixed resolution 112 × 112 pixels. It restricts our analysis about the performance of the model on other datasets such as DFDC and Face Forensics++ which have videos with higher resolution and diverse manipulation methods. Additionally, due to the limitation in computational power, our proposed framework selects only eight frames from each video, which might not be sufficient to learn the long-range temporal artefacts in videos. These findings demonstrate the effectiveness of the proposed design within the Celeb-DF benchmark; however, cross-dataset robustness has not been established in the present study and remains an important part of future evaluation. Cross-dataset evaluation remains an important next stage of this work. In particular, applying the Celeb-DF-trained model directly to unseen benchmarks such as FaceForensics++ or DFDC would provide a stricter assessment of whether the learned spatial, temporal, and forensic representations transfer beyond the acquisition and compression characteristics of Celeb-DF. Such evaluation requires a separate preprocessing and inference pipeline to ensure compatibility with the input resolution and eight-frame sequence structure used by the proposed model. A further limitation is that the quantitative classification results are evaluated at the frame level. Although the proposed architecture incorporates temporal modelling across sampled frames, predictions were not aggregated across all frames belonging to a source video to obtain a single video-level decision. Therefore, the reported 90.0% accuracy and associated classification metrics should be interpreted strictly as frame-level performance. Future work will evaluate video-level aggregation using a standardized source-video-level protocol.

7. Conclusions

In this work, an encoder–decoder neural network is developed for the detection of deepfake videos based on the Celeb-DF dataset. The designed framework is composed of a convolutional frame encoder consisting of six layers, a bidirectional gated recurrent unit of two layers for modelling the temporal sequence, and an attention-pooling layer which assigns a weight to each frame's importance adaptively. In addition, there is a parallel forensic decoder network which is responsible for reconstruction of Laplacian residual maps. This enables the network to capture detailed differences between edges

and textures of the real and fake faces. From the experimentation result, it is found that the proposed architecture produces an accuracy of 93% in training and 90% in testing with an ROC-AUC of 0.92 using the Celeb-DF dataset. Moreover, the confusion matrix is able to verify that there is a balanced performance for both classes with almost equal FPR and FNR. From the exploratory data analysis, it is evident that the spatial texture feature (brightness and sharpness) contributes the most towards the classification of the two classes whereas the frequency and temporal features remained insufficient alone for the classification. This verifies the validity of the selected architecture of combining spatial encoding along with the temporal representation instead of using the handcrafted features. Finally, a low reconstruction MSE of 0.010 confirms that the decoder branch captures meaningful forensic information that aids the detection task.

Author Contributions: K.D.: Writing—Original Draft Preparation, Visualization, Validation, Methodology, Formal Analysis, Data Curation, and Conceptualization. A.K.: Review and Editing, Visualization, Validation, Supervision, Investigation, Formal Analysis, and Correspondence. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset supporting the findings of this study is openly available in Kaggle at https://www.kaggle.com/datasets/pranabkc/deepfake-with-cropped-faces-from-video (accessed on 26 February 2026).

Acknowledgments: The authors express their deepest gratitude and appreciation to the Department of Computer Science and Engineering at Guru Jambheshwar University of Science and Technology for providing endless support and an inspiring environment for academic research. During the preparation of this manuscript, Grammarly Web Editor was used for reviewing and editing the content. The authors designed the methodology and algorithms, and executed the code. The experimental results and interpretations are all verified by the authors. The authors themselves take complete responsibility for the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ullah, M.U.; Gan, M.L.; Goh, H.G.; Yichiet, A.; Ponnsamy, V. A Comprehensive Deep Learning Approach for Deepfake Detection on the Celeb-DF Dataset. In Proceedings of the 2025 6th International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 10–11 December 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.
2. Singh, R.; Sharma, D.K.; Jain, P.K. DeepFake Detection Using YOLOv5 Feature Extraction and Classification on the Celeb-DF Dataset. In Proceedings of the 2025 International Conference on Digital Innovations for Sustainable Solutions (ICDISS), Faridabad, India, 14–15 November 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 1–5.
3. Li, Y.; Zhu, D.; Cui, X.; Lyu, S. Celeb-df++: A large-scale challenging video deepfake benchmark for generalizable forensics. *arXiv* **2025**, arXiv:2507.18015.
4. Viknesh, S.R.; Kumar, P.; Vng, A.S.; Sivakarthi, G. DeepFakeGuard: Real-Time Deepfake Video Detection Leveraging Celeb-DF Dataset and CNN-LSTM Framework. In Proceedings of the 2025 5th International Conference on Expert Clouds and Applications (ICOECA), Bengaluru, India, 6–7 March 2025; IEEE Computer Society: Los Alamitos, CA, USA, 2025; pp. 744–750.
5. Dasgupta, S.; Badal, K.; Chittam, S.; Alam, M.T.; Roy, K. Attention-enhanced CNN for high-performance deepfake detection: A multi-dataset study. *IEEE Access* **2025**, *13*, 101980–101993. [CrossRef]
6. Bakshi, P.; Verma, S.; Rawat, B.; Verma, E. Hybrid CNN-BiLSTM Architecture with Transfer Learning and Fine-Tuning for Efficient Deepfake Video Detection on Celeb-Df v2. In *Proceedings of the 2025 IEEE International Conference on Advances in Computing Research On Science Engineering and Technology (ACROSET), Indore, India, 27–28 September 2025*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.
7. Rana, P.; Kumari, R.; Bagga, R.; Kumar, P.; Kanupriya. Benchmarking transformers for deepfake detection: Analysing performance and identifying research opportunities. In Proceedings of the AIP Conference Proceedings, Mohali, India, 7–8 November 2024; AIP Publishing LLC: Melville, NY, USA, 2026; Volume 3343, p. 020016.

8. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S. Celeb-df: A large-scale challenging dataset for deepfake forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; IEEE Computer Society: Los Alamitos, CA, USA, 2020; pp. 3207–3216.
9. Tu, Y.; Wu, J.; Wang, J.; Gao, W. Dictionary learning-based super-resolution restoration for detecting low-quality facial forged videos. *J. King Saud Univ. Comput. Inf. Sci.* **2026**, *38*, 209. [CrossRef]
10. Alattas, E.; Clark, J.; Alsulami, B.; Jarraya, S.K. Beyond Frames: 3D-CoAtNet for Generalizable Deepfake Video Detection. *IEEE Access* **2026**, *14*, 29692–29705. [CrossRef]
11. Matthews, L.; Mkpong-Ruffin, I.; Evans, D.; Boonthum-Denecke, C. From Creation to Detection: How Dataset Composition and Simple Augmentation Influence Deepfake Training. *J. Colloq. Inf. Syst. Secur. Educ.* **2026**, *13*, 11.
12. Yermakov, A.; Cech, J.; Matas, J.; Fritz, M. Deepfake detection that generalizes across benchmarks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Tucson, AZ, USA, 6–10 March 2026; pp. 773–783.
13. Lagsoun, A.M.; Khouili, O.; Bekkari, A.; Boudra, N.; Oujaoura, M.; Jraifi, A.; Ech-Chadi, S.; Hedabou, M. Compressed deepfake detection via GA-LASSO selection of deep features and machine learning models. *Sci. Rep.* **2026**, *16*, 19708. [CrossRef] [PubMed]
14. Guo, T.; Li, J.; Tang, Y. A score based likelihood ratio framework for deepfake image identification in forensic science. *Sci. Rep.* **2026**, *16*, 12149. [CrossRef] [PubMed]
15. Chauhan, A.; Quadir Md, A. Hybrid Facial Deepfake Detection Framework Using Swin V2 Vision Transformers and Multi-Scale Textural Descriptors. *Signal Image Video Process.* **2026**, *20*, 233. [CrossRef]
16. Perelli, G.; Micheletto, M.; Concas, S.; Puglisi, G.; Marcialis, G.L. Robust Deepfake Detection in Compressed Videos with Scalable Network Strategies. *Expert Syst. Appl.* **2026**, *317*, 131761. [CrossRef]
17. Rajeev, A.; Raviraj, P. Spatiotemporal and Behavioral Feature-Aware Model with XceptionCapsule Fusion for Deepfake Detection. *Информатика И Автоматизация* **2026**, *25*, 110–152. [CrossRef]
18. Jarraya, S.K. *Beyond Frames: 3D-CoAtNet for Generalizable Deepfake Video Detection*; IEEE Computer Society: Los Alamitos, CA, USA, 2026.
19. Duhan, K.; Kajal, A. A Comparative Analysis of Deep Learning Based Approaches for DeepFake Identification. *Procedia Comput. Sci.* **2025**, *259*, 482–493. [CrossRef]
20. AlMuhaideb, S.; Alshaya, H.; Almutairi, L.; Alomran, D.; Alhamed, S.T. LightFakeDetect: A lightweight model for deepfake detection in videos that focuses on facial regions. *Mathematics* **2025**, *13*, 3088. [CrossRef]
21. Kumar, D.; Addula, S.R.; Lind, M.; Brown, S.; Odion, S. AI-Driven Hybrid Deep Learning and Swarm Intelligence for Predictive Maintenance of Smart Manufacturing Robots in Industry 4.0. *Electronics* **2026**, *15*, 715. [CrossRef]
22. Aslam, A.; Mir, J.; Zaman, G.; Rahman, A.; Salam, A.A.; Ali, F.; Alhiyafi, J.; Bakry, A.; Gul, M.J.; Gollapalli, M.; et al. Extracting Facial Features to Detect Deepfake Videos Using Machine Learning. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 834. [CrossRef]
23. Petmezas, G. Cross-dataset video deepfake detection using Transformer and CNN architectures. *Mach. Vis. Appl.* **2026**, *37*, 52. [CrossRef]
24. Wang, T.; Cheng, H.; Chow, K.P.; Nie, L. Deep convolutional pooling transformer for deepfake detection. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–20. [CrossRef]
25. Xu, J.; Liu, X.; Lin, W.; Shang, W.; Wang, Y. Localization and detection of deepfake videos based on self-blending method. *Sci. Rep.* **2025**, *15*, 3927. [CrossRef] [PubMed]
26. El-Gayar, M.M.; Abouhawwash, M.; Askar, S.S.; Sweidan, S. A novel approach for detecting deep fake videos using graph neural network. *J. Big Data* **2024**, *11*, 22. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.