applied sciences

MDPI

Review

An Investigation into the Utilisation of CNN with LSTM for Video Deepfake Detection

Sarah Tipper, Hany F. Atlam * and Harjinder Singh Lallie

Cyber Security Centre, Warwick Manufacturing Group, University of Warwick, Coventry CV4 7AL, UK; hl@warwick.ac.uk (H.S.L.)
* Correspondence: hany.atlam@warwick.ac.uk

Abstract: Video deepfake detection has emerged as a critical field within the broader domain of digital technologies driven by the rapid proliferation of AI-generated media and the increasing threat of its misuse for deception and misinformation. The integration of Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) has proven to be a promising approach for improving video deepfake detection, achieving near-perfect accuracy. CNNs enable the effective extraction of spatial features from video frames, such as facial textures and lighting, while LSTM analyses temporal patterns, detecting inconsistencies over time. This hybrid model enhances the ability to detect deepfakes by combining spatial and temporal analysis. However, the existing research lacks systematic evaluations that comprehensively assess their effectiveness and optimal configurations. Therefore, this paper provides a comprehensive review of video deepfake detection techniques utilising hybrid CNN-LSTM models. It systematically investigates state-of-the-art techniques, highlighting common feature extraction approaches and widely used datasets for training and testing. This paper also evaluates model performance across different datasets, identifies key factors influencing detection accuracy, and explores how CNN-LSTM models can be optimised. It also compares CNN-LSTM models with non-LSTM approaches, addresses implementation challenges, and proposes solutions for them. Lastly, open issues and future research directions of video deepfake detection using CNN-LSTM will be discussed. This paper provides valuable insights for researchers and cyber security professionals by reviewing CNN-LSTM models for video deepfake detection contributing to the advancement of robust and effective deepfake detection systems.

Keywords: deepfake; convolutional neural network (CNN); long short-term memory (LSTM); video deepfake detection; feature extraction

check for updates

Citation: Tipper, S.; Atlam, H.F.; Lallie, H.S. An Investigation into the Utilisation of CNN with LSTM for Video Deepfake Detection. *Appl. Sci.* **2024**, *14*, 9754. https://doi.org/10.3390/app14219754

Academic Editors: Mara Pistellato and Byung-Gyu Kim

Received: 6 October 2024
Revised: 21 October 2024
Accepted: 23 October 2024
Published: 25 October 2024

1. Introduction

Deepfakes are a form of online media depicting digitally synthesised humans. When used maliciously, this technology harms the fundamental human right to privacy, with the freedom of unreasonable constraints on the construction of one's own identity threatened through the often-non-consensual use of personal identities [1]. Therefore, the creation of deepfake detection systems acts as a privacy technology to ensure humans can identify authentic content whilst upholding the integrity of democratic political systems, providing transparency for manipulated audio and visual clips that are created to cause harm and spread misinformation. For example, deepfakes of political leaders urging the public to vote for a specific party or relaying false information about medical practices pose a threat to democracy and well-being [2].

Several Machine Learning (ML) techniques and tools have been developed for deepfake video detection to combat the growing threat of synthetic media manipulation. Common approaches include CNNs and Recurrent Neural Networks (RNNs), which can detect subtle artefacts or inconsistencies in facial movements, lighting, and textures often missed by the human eye [2]. Tools such as FaceForensics++, Deepware Scanner, and Microsoft's

Video Authenticator leverage these techniques to identify discrepancies between real and manipulated content. Researchers also employ adversarial training, where a Generative Adversarial Network (GAN) creates deepfakes while another model attempts to detect them, improving accuracy over time. These ML-based techniques continuously evolve to counter increasingly sophisticated deepfake technologies, providing crucial defences against misinformation, fraud, and identity theft [3,4].

The combination of CNN and LSTM has proven to be particularly effective for video deepfake detection, offering impressive accuracy levels, sometimes approaching 100% [2]. CNNs shine at extracting spatial features from individual video frames, capturing fine-grained details such as inconsistencies in facial textures, lighting, and image resolution. LSTM networks, on the other hand, analyse temporal patterns, detecting irregularities in how these features evolve over time, which can signal deepfake manipulation. By integrating both spatial and temporal analysis, this hybrid model allows for a comprehensive evaluation of videos, making it difficult for deepfakes to evade detection. This approach not only boosts accuracy but also provides a more robust defence against increasingly sophisticated deepfake algorithms, ensuring that manipulated content is identified more reliably in real-world applications [2,3].

Although the integration of CNN and LSTM for video deepfake detection provides countless opportunities and advantages compared to relevant techniques, there is a lack of research studies that systematically evaluate the effectiveness of this hybrid model and provide the necessary details for the effective implementation of this model. While existing studies demonstrate the potential of this hybrid model, they often focus on isolated implementations and lack a comprehensive review of the techniques, datasets, and performance metrics used across various approaches. There is limited insight into the optimal configurations of CNN-LSTM architectures and the best practices for improving detection accuracy in different contexts, particularly as deepfake generation techniques evolve. Furthermore, the challenges of high computational costs and scalability issues associated with training these models remain underexplored.

This paper aims to provide a comprehensive investigation of the current landscape of video deepfake detection techniques that utilise the hybrid model of CNN with LSTM. By systematically reviewing state-of-the-art approaches, this paper identifies the most common feature extraction techniques employed in these tools and examines the datasets frequently used for training and testing. This paper also evaluates the performance of these models on different datasets to highlight their strengths and limitations. A key focus of this paper is also to investigate the factors that have the greatest influence on detection accuracy, offering insights into how CNN-LSTM models can be optimised. Additionally, this paper compares CNN-LSTM models with alternative approaches that do not employ LSTM to assess their relative effectiveness in deepfake detection. This paper also explores the challenges associated with implementing these tools and proposes potential solutions to overcome these barriers. Lastly, open issues and future research directions related to the use of CNN-LSTM in video deepfake detection models will be presented. Through this systematic review, this paper contributes valuable guidance for future research and the development of more robust, efficient, and accurate video deepfake detection systems. Compared to similar reviews, this paper provides a distinct contribution by focusing specifically on the integration of CNN with LSTM for video deepfake detection. While other reviews may cover a broad range of detection techniques, this paper narrows its scope to examine the unique strengths, challenges, and optimisation strategies of CNN-LSTM models, providing a focused analysis of how this hybrid approach effectively captures both spatial and temporal features in videos.

The contributions of this paper can be summarised as follows:

- Conducting a comprehensive investigation of the current landscape of state-of-the-art video deepfake detection studies and tools that leverage CNN with LSTM.
- Identifying the most common feature extraction techniques employed within video deepfake detection techniques utilising CNN with LSTM.

- Examining the most commonly used datasets in the development and evaluation of video deepfake detection techniques that integrate CNN with LSTM and evaluating their performance.
- Investigating the key factors that have the most significant influence on detection accuracy when employing CNN with LSTM in video deepfake detection.
- Comparing the effectiveness of CNN-LSTM models against alternative models that do not incorporate LSTM in video deepfake detection.
- Examining the challenges of implementing CNN-LSTM-based video deepfake detection systems and offering insights into possible solutions.
- Investigating open issues and future research directions regarding the integration of CNN with LSTM in video deepfake detection.

The rest of this paper is organised as follows: Section 2 provides an overview of video deepfake creation and detection as well as an overview of CNN and LSTM algorithms; Section 3 introduces the research methodology that was adopted to conduct this review; Section 4 presents the analysis of the results; Section 5 provides the results and discussion, with the answers to research questions presented in detail; Section 6 presents open issues and future research directions regarding utilising CNN and LSTM for video deepfake detection; and Section 7 provides the conclusions.

2. Video Deepfake Detection with CNN-LSTM

This section provides an overview of video deepfake detection using the combination of CNN and LSTM. It starts by discussing how deepfakes are created and detected, followed by a brief introduction to the key features of CNN and LSTM.

2.1. Deepfake Creation and Detection

Deepfakes are hyper-realistic photos, videos, or audio recordings of humans that are digitally manipulated using ML methods. The technology can falsely depict people talking, altering their mannerisms, their movements, and their facial expressions despite the real person not conducting these actions, and cannot always be detected by the human eye. Often, common targets are celebrities or influential figures, whose image is widespread to the public, commonly in the form of traditional and social media. By using a large dataset of images and videos, ML can be utilised to create and apply digital manipulations [3].

GAN is the most common technique employed to generate different types of deepfakes, which is an unsupervised learning algorithm that deals with two subnetworks: trained adversarial with a generator that generates new examples and learns class distribution and a discriminator that classifies input data as real or fake. The generator takes random noise as input to generate an image before the discriminator outputs a probability between zero and one of genuineness. When the discriminator incorrectly classes a fake image as real, the network can generate realistic data [4]. Another technology utilised for creating deepfakes is a Variational Autoencoder (VAE), which consists of an encoder and decoder. The encoder transforms high-dimensional input data into a distribution over latent space whilst the decoder samples from the posterior distribution to capture the variability in the data to generate diverse outputs [5]. Deepfake generation can be categorised into three categories: attribute manipulation, identity swap, and face synthesis.

Facial attributes refer to the visual characteristics of faces and can be defined as the inherent properties of human faces, which are categorical and interpretable and include human features such as eyes, lips, noses, beards, hair, and material objects such as glasses [6]. Facial attribute manipulation technology alters appearance by changing facial properties, for example, the modification of hair colour from brown to blonde using a GAN is an example of attribute manipulation, generating a new fake image of a human who does not technically exist. Several studies recognise identity swap as one of the most common deepfake techniques [7], referring to the process of swapping a human face in a source image to a target image. Often, an autoencoder is deployed, which is a type of Neural Network (NN) that uses unsupervised learning to turn inputs into outputs [8]. A face must be detected in both the source and the target media in the pre-processing stage as facial attributes are identified and blended into the target face before overlapping the regions [9]. Alternatively, face synthesis refers to the creation of non-existent faces that are purely computer synthesised. Combining CNN and GANs is a common creation method, where the network classifies images with a pre-trained discriminator and generator that manipulates the attributes of the face to generate images [10]. Like attribute manipulation, segmentation mapping can be used as input to generate completely synthetic faces in pictures or videos [11]. However, control over which facial attributes are manipulated remains limited.

To detect deepfake content using unsupervised learning, a feature extraction process is required to determine what facial attributes and external elements can be used to detect whether a face is real or a deepfake. This process involves gathering a large number of samples, selecting some for training, cutting the video into sequential frames, re-scaling the images to be of the same proportions and sizes, and ensuring they all contain the desired features.

During the feature extraction stage, facial landmarks are plotted which can also be used for deepfake detection. Through plotting key attributes, the model can detect inconsistencies, both in a single frame if the landmarks are far off and using a sequence of frames when monitoring how consistently facial landmarks change during a video. Another method of deepfake detection is monitoring the consistency of the audio to the visual cues from the lips by detecting if the mouth is open or closed when the audio occurs [12] or capturing inconsistencies such as a lack of lip synchronisation to determine the audio–visual dissonance over a video [13]. However, this could falsely mislabel a real video with audio corruption or poor quality.

A common method to detect deepfakes is using spatial–temporal analysis, which uses data characterised by space and time to analyse data frame legitimacy. Deepfakes lack temporal coherence as frame manipulation produces low-level generated artefacts [14], referring to original artefacts created to look like real video frames. Although the generative method of the deepfake impacts its granularity, this method can detect details a human eye cannot see. A common spatial–temporal approach to deepfake detection is CNN with LSTM, where facial landmarks are mapped and sequential frames are captured to follow features iterating over the time–space domain to extract inconsistencies [15].

2.2. Convolutional Neural Network (CNN)

Convolution, a mathematical process, involves the merging of two functions by shifting one across the other. At every overlapping point, the values of the functions are multiplied, generating a new function that illustrates how the shape of one is modified by another. It can be expressed mathematically as follows:

$$(f \times g)(x) = \int_{-\infty}^{\infty} f(\tau)\ \cdot g(x-\tau)\ d\tau \tag{1}$$

where

- $f(x)$ and $g(x)$ are the input functions;
- τ is the integration variable;
- $(f \times g)(x)$ is the convolution of $f(x)$ and $g(x)$ evaluated at (x).

This equation represents the operation of sliding $g(x)$ along the x-axis, multiplying it with $f(x)$ at each position, and then integrating it into all the possible positions. A CNN consists of three layers, including a convolution layer, a pooling layer and a fully connected layer which maps the extracted features into a final output [16], as shown in Figure 1.

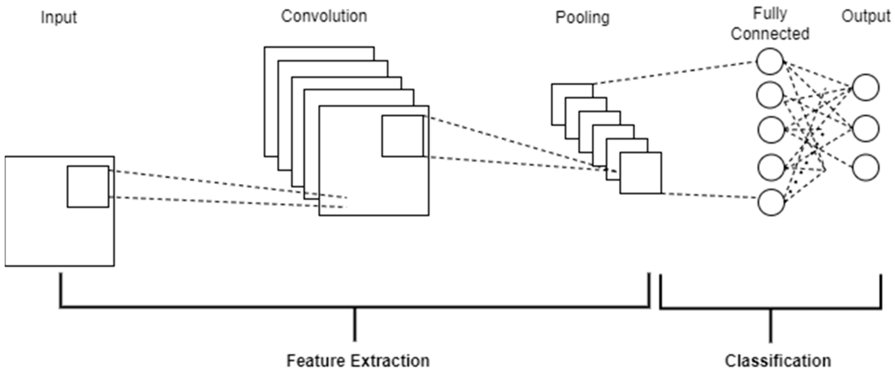

Figure 1. The three-layer architecture of CNN.

The convolutional layer performs most of the computational tasks in a Neural Network by applying a sliding window function, called a kernel or filter, to a 3D matrix of pixels representing an image. This process extracts features and recognises specific patterns, such as facial distortions [17]. Multiple kernels scan the image to generate new grids highlighting detected features. To introduce nonlinearity, Rectified Linear Unit (ReLU) transformations are applied, addressing the vanishing gradient problem by outputting zero for negative inputs and preserving positive values [18].

The pooling layer aims to extract significant features from the convoluted matrix while reducing dimensionality and the number of parameters in the model [19]. It slides a filter across the input and applies an aggregation function, with max pooling selecting the maximum pixel value and average pooling computing the average [20]. Max pooling is generally preferred for its noise suppression, while average pooling reduces noise through dimensionality reduction [21]. Both methods help mitigate overfitting by summarising outputs and decreasing spatial representation size [20]. The fully connected layer maps input representations to the output by connecting all neurons from the previous and output layers [22]. It computes nonlinear combinations through matrix multiplication and biases and then applies an activation function for classification. This process generates probabilities for final label predictions, with binary classification commonly used in deepfake detection to label inputs as real or deepfake [23].

2.3. Long Short-Term Memory (LSTM)

LSTM is a type of Recursive Neural Network (RNN) that effectively mitigates the vanishing gradient problem by storing information over long periods [24]. It outperforms standard feed-forward networks and RNNs due to its memory blocks and recurrent connections [25]. Each LSTM cell features three gates: the input gate, which controls information added to memory using the sigmoid function; the forget gate, which determines what to remove; and the output gate, which generates a filtered output vector using the tanh activation function [13]. Following the information flow through the cell gates, the output h_t is calculated based on the updated cell state and output gate activation. This is depicted in Figure 2.

Bidirectional RNNs enhance standard RNNs by utilising separate forward and backward states for neurons, allowing outputs to be processed independently [26]. When LSTM memory blocks replace the hidden states in a bidirectional RNN, it forms a bidirectional LSTM [27]. This architecture can be trained like a standard LSTM but requires two steps for the forward pass, processing input data through both states before activating output neurons. The backward pass calculates derivatives for the output and both states. Consequently, bidirectional LSTMs utilise both past and future contexts, improving performance in sequential data processing. Figure 3 illustrates these forward and backward passes.

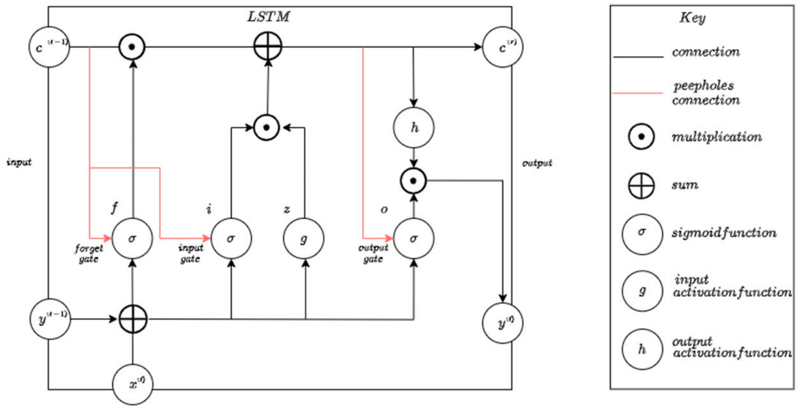

Figure 2. A standard LSTM block.

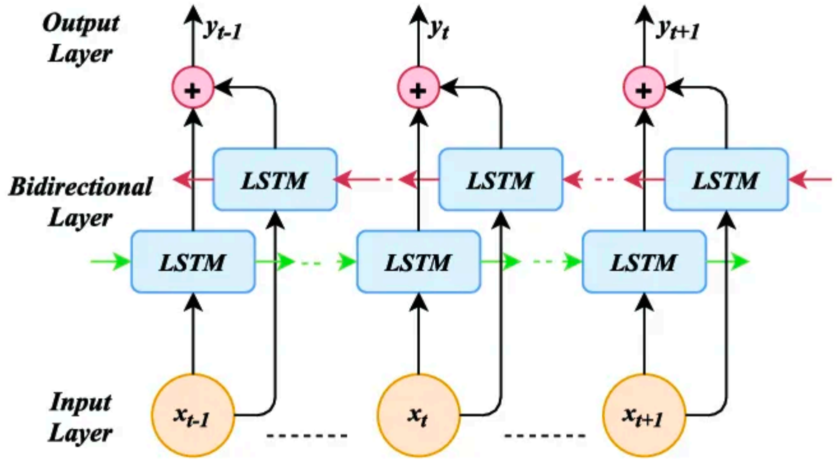

Figure 3. A bidirectional LSTM block.

3. Research Methodology

This systematic literature review (SLR) aims to define, analyse, and interpret all available research relevant to utilising CNN and LSTM for video deepfake detection. While video deepfake detection has become an area of growing interest, the complexity and evolving nature of deepfake technology necessitates an in-depth examination of current detection techniques. This SLR investigates the current landscape of video deepfake detection methods that leverage the hybrid model of CNN combined with LSTM. This review identifies the most common feature extraction techniques utilised in these models and evaluates the datasets frequently used for training and testing. By systematically reviewing state-of-the-art approaches, this paper aims to highlight key trends and offer insights into optimising CNN-LSTM models for enhanced detection accuracy.

To ensure transparency, reproducibility, and scientific rigour, this review follows the Preferred Reporting Items for Systematic Review (PRISMA) 2020 protocol developed by Page et al. [28]. PRISMA was first introduced in 2009 and is widely used across various disciplines. We selected PRISMA due to its comprehensive nature and its potential to promote consistency in systematic reviews. The PRISMA protocol outlines five key stages, as shown in Figure 4, for conducting an SLR, which were applied in this paper. The first stage involves formulating research questions that guide the review process. Following this, inclusion and exclusion criteria are established to ensure that the selected studies are relevant and aligned with the research objectives. In the third stage, relevant research databases are identified and searches are conducted to retrieve pertinent literature. The fourth stage focuses on analysing the findings from the reviewed studies, and in the fifth stage, the outcomes and the results are discussed.

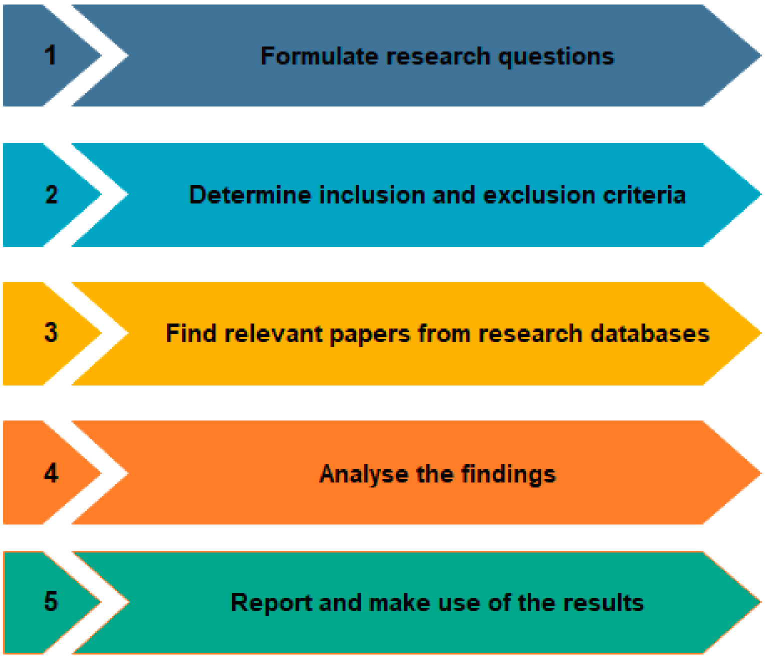
1
Formulate research questions
2
Determine inclusion and exclusion criteria
3
Find relevant papers from research databases
4
Analyse the findings
5
Report and make use of the results

Figure 4. The five stages of the systematic literature review.

Based on this strategy, 45 (out of 674) papers were selected, discussed, and criticised, leading to a section on open issues and further research. This allows researchers to explore the topic in depth to develop CNN-LSTM detection techniques and tools based on advanced findings.

Although this paper aims to provide detailed coverage of the topic, there may be limitations through constraining the scope such as the risk of overlooking important field contributions resulting from research outside of the inclusion and exclusion criteria. For example, limiting the scope to papers that were published in English biases the findings from non-English-speaking authors, despite their research following the same methodologies. Therefore, the source selection method may restrain the research availability, and the PRISMA statement is necessary. Additionally, publication bias could occur as the search strategy only identifies published research within known databases [29]. Although this research is regarded higher, studies show that results with positive findings are more likely to be published in high-impact journals [30], and therefore valid research is potentially excluded. However, the benefits of the peer reviewing of these journals ensure that the research is integral. Further limitations exist in the form of subjectivity, as prior knowledge on the larger topic may impact assumptions on what is regarded as ‘sufficient enough’ research to include within the SLR.

3.1. Research Questions

This paper seeks to address the following research questions:

- RQ1: What is the current landscape of state-of-the-art video deepfake detection studies that utilise CNN with LSTM?
- RQ2: What are the most common feature extraction techniques used in video deepfake detection tools that utilise CNN with LSTM?
- RQ3: What are the most common datasets used in the implementation of video deepfake detection tools that utilise CNN with LSTM?
- RQ4: What are the factors that have the strongest influence on detection accuracy for video deepfake detection when implementing CNN with LSTM?
- RQ5: Is using CNN with LSTM more effective for video deepfake detection compared to models that do not utilise LSTM?

- RQ6: What are the challenges involved in implementing video deepfake detection using CNN with LSTM?

3.2. Inclusion and Exclusion Criteria

Inclusion and exclusion criteria have been developed to ensure appropriate research papers are selected for answering the research questions.

The inclusion criteria include the following:

- Peer-reviewed journals and conference articles to ensure high-quality and credible sources.
- Relevant to the specific research questions.
- Topic mainly on video deepfake detection using CNN and LSTM.
- Full and available articles to allow for a comprehensive review of the content.
- English-language articles to maintain consistency in analysis.

The exclusion criteria were as follows:

- Articles concerning all other aspects of combining CNN and LSTM apart from video deepfake detection.
- Articles focused on video deepfake detection that do not discuss CNN and LSTM.
- Unpublished articles, non-peer-reviewed articles, and editorial articles to ensure credibility.
- Articles that are not fully available.
- Non-English articles to avoid translation issues and maintain analysis consistency.
- Duplicates of already included articles to avoid redundancy.

3.3. Data Sources

Recent studies have highlighted the growing importance of digital libraries in conducting comprehensive searches for systematic reviews. These electronic databases, selected based on their relevance and widespread recognition in current research, were instrumental in ensuring a thorough examination of the available literature. The digital libraries utilised in this SLR were chosen to align with the latest academic standards and recommendations. The electronic databases considered included the following:

- IEEE Xplore
- Google Scholar
- ACM Digital Library
- SpringerLink
- PubMed
- Elsevier ScienceDirect

3.4. Keywords

To gather the relevant information, the following keywords were used in searches, which were filtered to include research papers, journals, and conference proceedings:

- Video deepfake detection
- Convolutional Neural Network (CNN)
- CNNs in video deepfake detection
- LSTM in video deepfake detection
- Long Short-Term Memory (LSTM)
- Deepfake detection techniques
- CNN-LSTM hybrid models
- Deepfake detection datasets
- Temporal feature extraction
- Deepfake detection feature extraction techniques

Additionally, Boolean operations such as ′AND′, ′OR′, and ′NOT′ were used with the keyword search terms to obtain focused, relevant results.

3.5. Selection of Relevant Articles

Using the keyword-based search alongside the inclusion and exclusion criteria on the databases selected, articles were identified to fit the criteria. However, the publications needed to be refined to ensure that they contributed towards answering the research questions. The following three-phase selection process was used:

- Phase 1—Identification: Publications found during the search and those already in the collection were sorted using the inclusion and exclusion criteria. The scope of the search was narrowed to include only articles published recently.
- Phase 2—Screening: The titles and abstracts of the articles collected from several digital libraries were reviewed to determine how well they addressed the topic and the questions posed in this research work.
- Phase 3—Eligibility: During this stage, we focused on eliminating duplicates among the six digital libraries used for our publication collection.

4. Analysis of the Results

The inclusion and exclusion criteria were applied to the collected publications in three phases, according to the PRISMA 2020 statement [28]. In the first phase, a total of 674 articles were identified from six different databases: Google Scholar (210), IEEE Explore (38), PubMed (2), Elsevier ScienceDirect (90), ACM Digital Library (195), and SpringerLink (139). Then, in phase 2, the collected articles were screened based on the research questions where the articles that did not align with the research questions, were out of scope, or did not meet the inclusion criteria were excluded. This resulted in excluding 601 articles and moving forward with 73 articles. In phase 3, 28 duplicate articles were identified and removed from the 73 articles, leaving 45 articles that were included in this review. The flow diagram of the PRISMA process and the number of articles at each stage is shown in Figure 5.

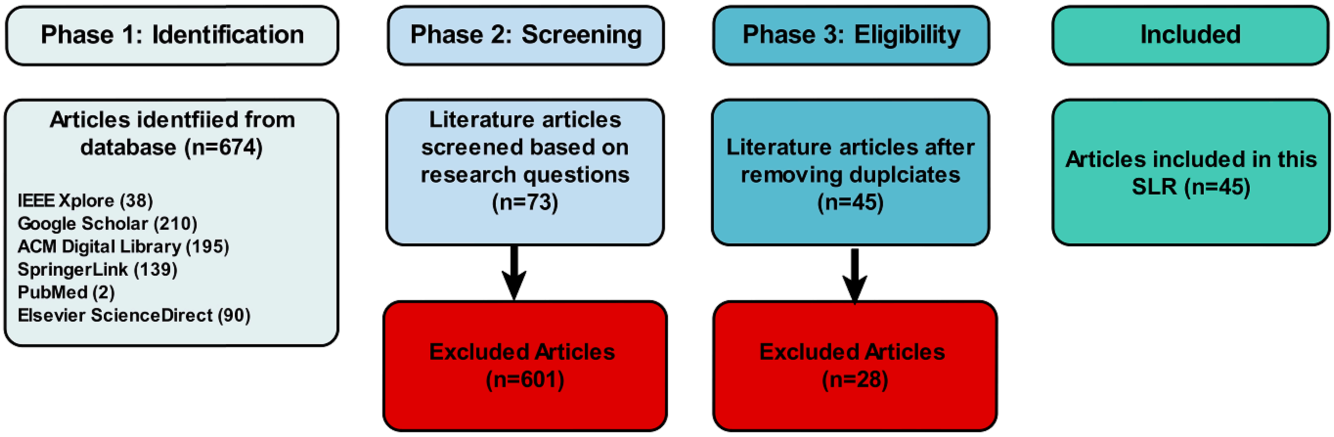
Phase 1: Identification
Phase 2: Screening
Phase 3: Eligibility
Included
Articles identfiied from database (n=674)
IEEE Xplore (38)
Google Scholar (210)
ACM Digital Library (195)
SpringerLink (139)
PubMed (2)
Elsevier ScienceDirect (90)
Literature articles screened based on research questions (n=73)
Literature articles after removing duplciates (n=45)
Articles included in this SLR (n=45)
Excluded Articles (n=601)
Excluded Articles (n=28)

Figure 5. The outcome of the three-phase selection process.

By applying the criteria to the online databases, the number of articles available was reduced to 45 throughout the three phases. The results within Table 1 show that Google Scholar provided the most relevant publications, while PubMed had the least publications related to using CNN with LSTM for video deepfake detection.

Figure 6 illustrates the number of publications per year. The graph suggests that the research focus on CNN and LSTM technologies for video deepfake detection particularly emerged from 2020 onward. The large spike in 2020 indicates a growing interest in this field, possibly due to advancements in technology and the rise in deepfake-related incidents. The 2022 peak aligns with the easing of the COVID-19 pandemic, suggesting that research activities resumed or increased following any pandemic-related delays. The consistency between 2022 and 2023 shows sustained interest in the area, with researchers continuing to explore these technologies for video deepfake detection.

Table 1. Selected publications from online databases.

Database	Phase 1	Phase 2	Phase 3
IEEE Xplore	38	20	13
Google Scholar	210	18	10
ACM Digital Library	195	12	7
SpringerLink	139	9	6
PubMed	2	1	1
Elsevier ScienceDirect	90	13	8
Total	674	73	45

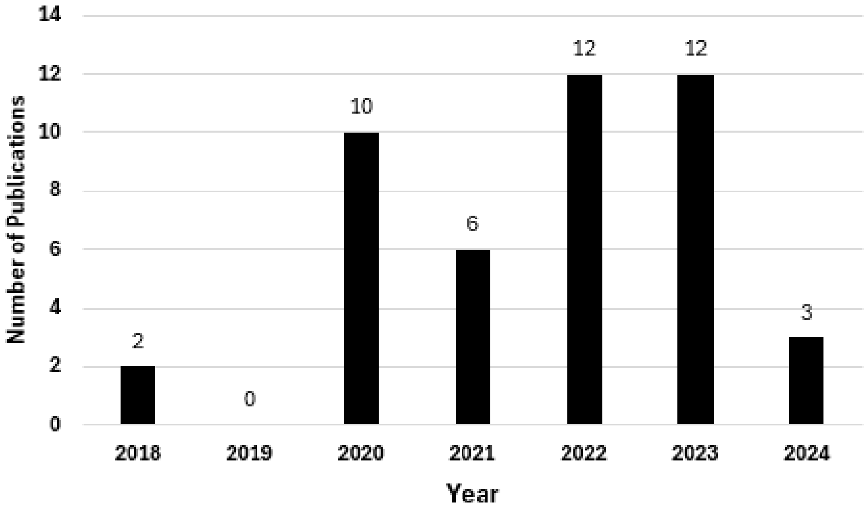

Figure 6. Number of publications per year.

The final publication list that was involved in this review is shown in Table 2, which assigns a publication ID to each publication, and includes the authors, year of publication, and publication type.

Table 2. Retrieved publications.

ID	Author	Year	Publication Type
1	Al-Adwan et al. [31]	2024	Journal
2	Al-Dhabi and Zhang [32]	2021	Conference proceedings
3	Al-Dulaimi and Kurnaz [33]	2024	Journal
4	Amerini and Caldelli [34]	2020	Conference proceedings
5	Chan et al. [35]	2020	Conference proceedings
6	Chen, Li, and Ding [36]	2022	Journal
7	Chinchalkar et al. [37]	2023	Conference proceedings
8	Chintha et al. [2]	2020	Journal
9	Fuad, Amin, and Ahsan [38]	2023	Conference proceedings
10	Gravina et al. [39]	2023	Conference proceedings
11	Guera and Delp [40]	2018	Conference proceedings
12	Hashmi et al. [41]	2020	Journal
13	Jaiswal et al. [42]	2021	Journal
14	Jalui et al. [43]	2022	Conference proceedings

Table 2. *Cont.*

ID	Author	Year	Publication Type
15	Jindal [44]	2023	Journal
16	John and Sherif [45]	2022	Conference proceedings
17	Jolly et al. [46]	2022	Conference proceedings
18	Jungare et al. [47]	2024	Journal
19	Kaur, Kumar, and Kumaraguru [48]	2020	Journal
20	Koshy and Mahmood [49]	2020	Journal
21	Kukanov et al. [50]	2020	Conference proceedings
22	Lai et al. [51]	2022	Journal
23	Li, Chang, and Lyu [52]	2018	Conference proceedings
24	Liang et al. [53]	2023	Journal
25	Malik et al. [54]	2023	Journal
26	Masi et al. [14]	2020	Conference proceedings
27	Masud et al. [38]	2023	Journal
28	Nawaz, Javed, and Irtaza [55]	2023	Journal
29	Parayil et al. [56]	2023	Journal
30	Patel, Chandra, and Jain [57]	2023	Conference proceedings
31	Ritter et al. [58]	2023	Conference proceedings
32	Saealal et al. [59]	2022	Journal
33	Saif et al. [60]	2022	Journal
34	Saikia et al. [61]	2022	Conference proceedings
35	Saraswathi et al. [25]	2022	Conference proceedings
36	Shende, Paliwal, and Mahay [62]	2021	Journal
37	Singh et al. [63]	2020	Journal
38	Sooda [64]	2022	Conference proceedings
39	Stanciu and Ionescu [65]	2021	Conference proceedings
40	Su et al. [66]	2021	Journal
41	Suratkar and Kazi [67]	2022	Journal
42	Taviti et al. [68]	2023	Conference proceedings
43	Wubet [69]	2020	Journal
44	Yadav et al. [70]	2021	Conference proceedings
45	Yesugade et al. [71]	2022	Book chapter

5. Results and Discussion

The growing interest in video deepfake detection using CNN and LSTM is driven by the increasing sophistication of AI-generated media and the rising concerns over misinformation, identity theft, and digital manipulation. As deepfake videos become more widespread, particularly in areas like politics, entertainment, and cybercrime, the demand for robust detection methods has surged. Traditional video analysis techniques struggle to detect subtle manipulations in deepfakes, making CNN and LSTM highly appealing due to their capabilities to handle complex patterns in image and video data. Researchers are therefore focused on developing and refining deep learning frameworks to enhance the accuracy and efficiency of deepfake detection.

The studies reviewed investigate the application of CNN and LSTM to detect deepfakes, showcasing their potential to improve detection accuracy by analysing both spatial and temporal features. Many studies propose innovative hybrid models that combine CNNs for frame-based analysis and LSTMs for temporal sequence analysis to identify the inconsistencies present in fake videos. However, a recurring limitation across these studies is the lack of comprehensive evaluations of diverse datasets and real-world environments, which raises concerns about the generalisability of the proposed solutions. While many papers introduce promising detection methods, they often focus on model design without providing sufficient empirical performance benchmarks, particularly regarding their ability to handle adversarial attacks or cross-dataset generalisation. Additionally, challenges such as computational efficiency, scalability, and the need for real-time detection in high-stakes environments are underexplored. Privacy concerns, as well as the ethical implications of training models on manipulated content, further complicate the development of deepfake detection systems. While the potential of CNNs and LSTMs in video deepfake detection is clear, the absence of extensive evaluations, real-world deployment, and discussion on practical limitations suggests that further research is required to fully address the challenges identified in the current body of work.

This section presents a comprehensive and detailed analysis to answer the research questions by integrating insights and contributions from various publications.

RQ1: What is the current landscape of state-of-the-art video deepfake detection studies that utilise CNN with LSTM?

The utilisation of CNN in conjunction with LSTM has significantly advanced the field of deepfake detection, pioneered by Guera and Delp [40]. Their model employed an InceptionV3 CNN, removing the fully connected layer at the top of the network whilst using the feature vectors as their LSTM input. The LSTM unit is given a 0.5 chance of dropout to recursively process the frame sequences in a meaningful manner, and the model is trained end-to-end without the need for loss functions. As manipulations can occur at any point in a deepfake model, a continuous subsequence of fixed frame lengths was used as input with classification results achieving 97% accuracy on video feeds less than 2 s. Conversely, the Xception CNN-LSTM model, due to its application of depthwise separable convolutions, has been observed to surpass the capabilities of InceptionV3 in handling spatial information [72]. Additionally, integrating bidirectional LSTMs has contributed substantially to enhancing temporal information modelling capabilities, as observed in several studies [2,14,55].

Investigating the state-of-the-art video deepfake detection studies reveals that these studies can be categorised into several classifications. The first category is the studies that use bi-LSTM to analyse the temporal dynamics of video frames in video deepfake detection [2]. Passing features from the XceptionNet module into bi-LSTM architecture achieved remarkable results, with a double pass achieving a detection accuracy of 99.7%, surpassing the one-directional LSTM with a forward pass in two prominent datasets, FaceForensics++ and Celeb-DF. Similarly, significant advancements were made by employing bi-LSTMs to fuse two bidirectional outputs with convolutional filters, effectively boosting the frame-level detection performance for videos with medium compression rates, increasing the Area Under the Curve (AUC) score from 92% to 99% [14]. Furthermore, the implementation of the Dense-Swish-121 feature descriptor in conjunction with bi-LSTMs showcased exceptional accuracy, surpassing five state-of-the-art CNN LSTM models on the DFDC dataset, reaching an impressive video deepfake detection accuracy of 99.3% [55]. However, using an Xception-based CNN-LSTM model to reduce feature dimensions before the LSTM cells results in the loss of valuable temporal correlations because Xception uses spatial learning, rendering it transparent to temporal correlations within the input sequences [36]. This results in the LSTM cells receiving features that have lost their temporal context and hinders the model's understanding of intrinsic patterns and temporal contexts.

In response, the first spatiotemporal attention mechanism with ConvLSTM for deepfake detection was developed to solve the challenges introduced by Xception-LSTM-based algorithms [36]. Incorporating convolutional operations within the LSTM cell allows the model to handle spatial–temporal information effectively. The model's superiority to other state-of-the-art models is due to the spatiotemporal attention mechanisms being introduced before dimensionality reduction and the ConvLSTM considering the structure information of features during temporal modelling by the LSTM. The model outperforms other state-of-the-art methods with 99% detection accuracy on the FaceForensics++ dataset.

The second category is the studies that discussed the blinking rate and its effect on the detection of video deepfakes. Several studies have researched using just the eye landmarks to determine whether a video is real or deepfake. The introduction of a long-term recurrent CNN captured the temporal relationship between consecutive eye-blinking frames from an open to a closed state [52]. The findings showed that deepfake videos have a blinking rate ten times lower than real videos, therefore acting as a threshold in conjunction with spatial–temporal detection. Training a ResNet-50 CNN on the MRL eye image dataset determined during the testing stage that the blinking rate was 4.3 times lower than the blinking rate in real videos on the UADFV dataset [69]. The overall detection accuracy on real videos was 92.23% and 98.3% on fake videos; however, a very small sample was used. An extension of this research determined a lack of, or too frequent, blinking can indicate deepfake content [59]. This model used a CNN for feature extraction and then standardised a frame rate for the blinking patterns to result in 255 eye-state probabilities where an eye is fully open or completely shut. The output was fed into an LSTM cell to determine the eye's state before a Fully Connected Network (FCN) denotes its classification. The model's accuracy was highest when trained when the batch size was set to 20 and trained for 105 epochs, achieving 95.57% on the FaceForensics++ dataset.

Another category discussed in the literature studies was emotional analysis and its role in video deepfake detection. For example, Gravina et al. [39] use a CNN trained to detect human faces with another trained to associate a human emotion within the frame before a three-layered bidirectional LSTM network captures sequential data and a fully connected layer classifies the video as real or deepfake. The model outperforms the performance of the DFDC competition winners using 403 million fewer parameters. The results concluded emotional analysis is a robust method of deepfake detection, although facial representations of emotion can differ amongst age, gender, and culture, therefore making a non-biased baseline hard to establish on the limited datasets available.

Prediction error rate and its effect on the effectiveness of CNN—LSTM models was also another category discussed in the literature. When dealing with compressed video feed, Amerini and Cadelli [34] use the prediction rate error between frames, averaging 94.29% detection accuracy on lossless videos in the FaceForensics++ dataset. This involves calculating the difference between the current macroblock and motion-compensated macroblock to look for residual errors in predicting the succeeding frame to determine whether a deepfake face has been inserted into the video and altered the sequence structure. The model uses a 2D convolutional layer and a max pooling layer to have time (t) distributed output, which is then input into an RNN with LSTM cells, with every LSTM cell considered input with its previous LSTM cell state at $t - 1$. Later research [61] outperforms Amerini and Cadelli [34], reaching 97.25% accuracy on compressed videos in the same dataset using ResNeXt-50 architecture with an LSTM layer.

Another category discussed in the literature studies was CNN-LSTM hybrid models. The use of CNN-LSTM hybrid models across various research papers is a prevalent approach to deepfake detection. Al-Adwan et al. [31] propose a model that combines CNN with LSTM, optimised using PSO, whereas Al-Dhabi and Zhang [32] use a ResNeXt-50 backbone alongside LSTM to bridge the gap between training and test accuracy. Both papers highlight the importance of optimising CNN-LSTM architectures, yet they differ in the optimisation techniques used (PSO vs. traditional gradient descent) and the datasets on which they are applied. Similarly, the work by Guera and Delp [40] also employs InceptionV3 CNN with LSTM for deepfake detection, achieving 97% accuracy with limited frames (40–80 frames). This is comparable to Hashmi et al. [41], who propose CNN-LSTM for efficient feature extraction without memory expense, but their model detects only key facial landmarks rather than additional features, such as noise or blur, which limits its robustness. In both cases, the emphasis on balancing spatial and temporal features is evident, but the strategies for handling data limitations diverge.

Al-Dulaimi and Kurnaz [33] and Kaur et al. [48] also introduce hybrid CNN-LSTM models, achieving high accuracy on benchmark datasets, such as DFDC and Ciplab. However, Al-Dulaimi and Kurnaz [33] focus on general deepfake detection, while Kaur et al. [48] specifically address face-swapping manipulations, which illustrates a common theme: while many papers rely on similar CNN-LSTM architectures, their focus areas and application domains differ significantly. Amerini and Caldelli [34], for instance, target video-based manipulations using prediction errors between frames but struggle with compressed video data, which limits its general applicability. An emerging trend is the integration of attention mechanisms to improve detection performance. For example, Chen, Li, and Ding [36] employ a spatiotemporal attention mechanism within a ConvLSTM framework, and Fuad, Amin, and Ahsan [38] introduce a multi-head attention layer for stronger video classification performance. These attention mechanisms focus on enhancing critical feature selection, further refining CNN-LSTM models for greater generalisability and efficiency. However, as noted by Chen et al. [36], increasing model complexity can lead to overfitting, a concern repeated by Masud et al. [38], whose lightweight model overfits after increasing the number of LSTM layer units beyond 128.

The last category of studies investigated the generalisability issue in CNN-LSTM models. A key limitation across many models is their generalisability to new or unseen datasets. For instance, Chintha et al. [2] achieve 99.7% accuracy using XceptionNet with bidirectional LSTM cells on the FF++ dataset, but the model's performance on other datasets is not evaluated. Jaiswal et al. [42] and Jalui et al. [43] also raise concerns about generalisability, as their models are tested on the same datasets used for training, limiting their real-world applicability. Kukanov et al. [50], by contrast, propose a maximal figure-of-merit (MFoM) framework that reduces equal error rate (EER) but fails to detect deepfake content on YouTube, a major source of online deepfakes. While some models, such as that of Patel, Chandra, and Jain [57], show promise with high accuracy on small frame sets, others like Saealal et al. [59] struggle with specific types of deepfake manipulations, such as neural textures. This discrepancy in handling diverse fake types is common across studies, with most models excelling in detecting face-swap or blinking anomalies (e.g., Li, Chang, and Lyu [52]) but underperforming on other forms of deepfake content. Table 3 provides a summary of each of the selected papers' contributions to video deepfake detection and their limitations.

Table 3. A summary of the contributions and limitations of each study of the selected publications.

Citation	Summary of Contribution	Limitations
Al-Adwan et al. [31]	The paper proposes a model that combines CNN with LSTM architecture, where the weight and bias values are initialised using Particle Swarm Optimisation (PSO), which is finetuned on a validation set to obtain the optimal parameters for performance.	The failure of accuracy to improve with increased iterations points to a limitation in the training process or model architecture.
Al-Dhabi and Zhang [32]	The paper proposes a model that uses ResNeXt-50 with LSTM and bridges the gap between training and test accuracy on sequential frames of a video using different datasets, with an extremely low training loss of 0.0053.	The proposed model has a low accuracy as the frame rate input is reduced, diminishing its effectiveness.
Al-Dulaimi and Kurnaz [33]	The paper introduces a hybrid CNN-LSTM model for deepfake image detection, achieving 98.21% precision on benchmark datasets (DFDC, Ciplab). This approach effectively combines spatial and temporal analysis to improve deepfake identification accuracy, addressing a critical concern in digital media	The proposed model's 0.26% error rate highlights the inherent difficulty of deepfake detection; further research is needed to improve robustness and generalisability.
Amerini and Caldelli [34]	The paper proposes a CNN-LSTM method that uses the prediction error from the current to the succeeding frame to determine whether deepfake content has been inserted into the video as it would modify the intrinsic structure of the sequence.	The proposed CNN-LSTM method performed very badly on strongly compressed data, reducing accuracy to 61%.
Chan et al. [35]	The paper provides a theoretical framework that enables proof of authenticity using LSTM as a deep encoder for creating unique discriminative features, which are compressed and hashed into a transaction. The content is validated and can then be traced back with a label verifying its authenticity as not deepfake.	The model makes the assumption that the uploaded content is not deepfake but could be integrated. Also, the maximum uploaded payload size is only 100 MB.
Chen, Li, and Ding [36]	The paper introduces a spatiotemporal attention mechanism using ConvLSTM to address the difficulties presented by Xception-LSTM-based algorithms in identifying deepfake videos.	The model runs the risk of overfitting and reduced generalisability from an increase in model complexity.
Chinchalkar et al. [37]	The paper proposes an RNN model that is trained on 150 frames per video to check for frame consistencies before using CNN ResNet architecture to retrieve frame-level characteristics, followed by RNN-LSTM to determine video authenticity.	The proposed model is trained on a high frame rate, reducing its generalisability to lower frame rate deepfake content.
Chintha et al. [2]	The paper uses a CNN to obtain a vector representation of the facial region within a frame before passing a sequence of frames to a bidirectional LSTM cell. The spatial features from the XceptionNet module are passed into bidirectional LSTM cells with a double pass to give 99.7% accuracy on the FF++ dataset.	The evaluation of the proposed detection method is based on specific datasets, potentially limiting the generalisability of the results.
Fuad, Amin, and Ahsan [38]	The paper proposes a model that uses a Wide CNN ResNet architecture with LSTM before inputting data into a multi-head attention layer, so the model selects essential features for stronger performance on video classification.	The generalisability of the findings to a broader range of deepfake manipulations is limited as the study focuses on a specific transfer learning approach.
Gravina et al. [39]	The paper proposes an InceptionNetV3 CNN with LSTM that shows that emotional analysis is a robust method of deepfake detection, aiming to identify how visual emotional artefacts in deepfake videos disrupt temporal coherence.	Facial expression and representation of emotion can vary amongst contexts and cultures, therefore making a non-biased baseline hard to establish.
Guera and Delp [40]	The paper proposes combining InceptionV3 CNN with LSTM for sequence processing without loss functions and performs with 97% accuracy on 40–80 frames.	Where a face is not fully present, such as a facial rotation, the model has a low accuracy of 40%.

Table 3. *Cont.*

Citation	Summary of Contribution	Limitations
Hashmi et al. [41]	The paper combines CNN for feature extraction and LSTM for storing feature vectors for frames instead of saving the frame itself to train large amounts of data without memory expense.	The proposed model's feature extraction only detects key facial landmarks, not additional features such as noise or blur.
Jaiswal et al. [42]	The paper proposes a hybrid approach to deepfake detection using GRU and LSTM showing that CNN has higher performance metrics than when using LSTM alone, with particular emphasis on placing the GRU layers before the LSTM layers on the DFDC dataset.	The proposed model uses the same dataset for training and testing, reducing generalisability.
Jalui et al. [43]	The paper combines ResNeXt-50 for feature extraction with LSTM to train the model for classification whilst intentionally reducing the number of frames fed into the network, resulting in very high detection accuracy on the DFDC dataset.	The proposed model trains on 300 frames per video and uses the same dataset for training and testing, which is computationally expensive.
Jindal [44]	The paper proposes a model that adopts ResNeXt-50-32×4d architecture with LSTM dimensions of 2048 and 2048 hidden layers, concluding the architecture has stronger detection accuracy at 100 frames per video analysed in comparison to any value of frames below that.	The proposed model is computationally expensive, limiting its applicability where high-speed processing is required, despite achieving high accuracy.
John and Sherif [45]	The paper compares the performance accuracy of using temporal-based detection and triplet loss detection. A combination of a temporal model with a triplet deep model is created to obtain higher training and testing accuracy than standalone models.	The proposed model is limited to using a frame rate of 100 frames per second (fps), with higher frames per second improving accuracy.
Jolly et al. [46]	The paper proposes a model that uses a CNN-LSTM followed by a Recycle-GAN with a two-stream fusion, merging the spatial and temporal data to evaluate the data whilst pushing its results back to the start of the network for continuous learning.	The proposed model failed to detect any neural textures in generated deepfakes, demonstrating a critical weakness.
Jungare et al. [47]	The paper proposes a novel deepfake detection model using a ResNeXt-50 CNN and LSTM. Its key contribution is achieving 83.21% accuracy while processing only 10 frames, significantly improving efficiency.	The diminishing returns after 100 frames significantly limit the model's ability to leverage longer video sequences for improved accuracy.
Kaur, Kumar, and Kumaraguru [48]	The paper proposes a C-LSTM model that uses CNN with LSTM to detect face-swapped politicians and achieves an accuracy of 98.21% for 0.99 precision and 0.93 recall values.	When the video subject consistently looks away, detection accuracy is compromised.
Koshy and Mahmood [49]	The paper proposes a CNN-LSTM model without a pre-processing stage, applying nonlinear diffusion to enhance liveness detection on video sequences as the CNN captures spatial information and the features obtained by the LSTM layer are classified as real or fake with a test accuracy of 96%.	The proposed model's applicability is limited by its lack of testing on non-GAN-generated fake faces, a significant class of deepfakes.
Kukanov et al. [50]	The paper proposes a maximal figure-of-merit (MFoM) framework, which uses the detection cost function as its performance measure and the equal error rate (EER) to optimise measures of performance and view video deepfake detection as a cost-sensitive objective. The model then follows a CNN-LSTM-based approach to determine a reduction in the EER of detection.	The proposed model fails to detect deepfake content from YouTube videos, which is viewed as the 'worst case', although it is the most readily available deepfake content online.

Table 3. *Cont.*

Citation	Summary of Contribution	Limitations
Lai et al. [51]	The paper proposes a feature fusion detection model where feature extraction from spatial, temporal, and frequency domains occur at the same time. The results show that the error rate is higher without feature fusion or LSTM.	The proposed model lacks generalisability to datasets it was not trained on, indicating a high variance and low bias in its predictive capabilities.
Li, Chang, and Lyu [52]	In the paper, a long-term recurrent CNN is employed with LSTM to detect deepfakes using a lack of blinking by detecting the number of frames an eye is open and its temporal correlation to its preceding frame. The results conclude that deepfake videos have a blinking rate ten times lower than real videos.	The threshold set to determine an open or closed eye impacts detection, affecting the precision and recall of the overall eye-state classification system.
Liang et al. [53]	The paper proposes a model that combines facial geometry feature maps with CNN-LSTMs to decode and detect manipulations at a pixel-wise level. The model outperforms XceptionNet and CNN-based models on a range of datasets.	The proposed model is not robust as it relies on facial standards, which can change depending on expression or if something's blocking the view.
Malik et al. [54]	The paper demonstrates that a CNN with LSTM requires a larger training set than a testing set compared to standalone CNN, with higher accuracy detected on the FF++ dataset with an 80/20 split than when a 70/30 split is used, outperforming previous works.	The proposed model lacks an analysis of how image distortions and noise can affect the accuracy of its deepfake detection.
Masi et al. [14]	The paper proposes a model that uses a CNN with bidirectional LSTM and a loss function to place real faces within an inner hypersphere and deepfake faces in an outer hypersphere. The model improves frame-level detection for videos with medium compression rates, increasing the AUC score from 92% to 99%.	The proposed model has a low ability to detect real faces that have facial hair or are poorly illuminated, resulting in a high rate of false negatives and reduced overall accuracy.
Masud et al. [38]	The paper proposes a lightweight VGG16 CNN-LSTM that is 152 times smaller than existing methods, significantly improving computational efficiency and reducing resource requirements.	Adding more than 128 LSTM layer units led to overfitting and reduced performance on the validation test set.
Nawaz, Javed, and Irtaza [55]	The paper introduces a new feature descriptor named Dense-Swish-121, which integrates convolutional layers, dense blocks, and transitional layers with a bi-LSTM approach. The model outperforms five CNN-LSTM models on the DFDC dataset, reaching 99.3% accuracy with Dense-Swish-Net121 despite having fewer parameters.	The choice of the ReLU activation function limited the model's ability to detect subtle visual cues, resulting in reduced detection accuracy.
Parayil et al. [56]	The paper uses XceptionNet with LSTM, achieving a significant reduction in computational time compared to existing methods, while maintaining comparable or even superior accuracy	There is a large discrepancy and fluctuations between training and validation accuracy, indicating overfitting.
Patel, Chandra, and Jain [57]	The paper uses ResNeXt-50-32x4d with LSTM to determine whether a video is a deepfake or real with 85% accuracy using only 10 frames, which is equivalent to less than a second.	The proposed model is computationally expensive to analyse 300 frames of a video.
Ritter et al. [58]	The paper compares baselines and concludes that EfficientNetB7 is the most effective CNN model for detecting whether a video is real or deepfake using binary classification.	The model's testing accuracy is lower than its training and validation accuracy, which is therefore indicative of overfitting.
Saealal et al. [59]	The paper detects deepfake content using eye-blinking patterns, where a lack of, or too frequent, blinking can indicate deepfake content. Using a batch size of 20 with 105 epochs, the model's optimal results occurred at 95.57% on the FF+ dataset.	The proposed model poorly detects neural texture deepfakes, resulting in a higher rate of missed detections for this specific type of deepfake.

Table 3. *Cont.*

Citation	Summary of Contribution	Limitations
Saif et al. [60]	The paper uses a multi-stream CNN-LSTM network with contrastive performing the best using EfficientNetB7 architecture on the FF++ dataset with 97.3%. Introducing contrastive loss allowed the model to learn features independent of the deepfake generation method, which influences the FF++ dataset.	The proposed model struggles to detect expression-swapping deepfakes compared to face-synthesis deepfakes due to the subtler visual artefacts present in expression swaps.
Saikia et al. [61]	The paper combines optical flow during feature extraction, VGG16 CNN, and LSTM; this model's performance fares 91% on the FF+ dataset with a reduced frame rate of 70 frames a second, indicating that optical flow can assist with the early detection of deepfakes.	The proposed model does not perform as well on the DFDC dataset, with low frame frames (20) producing a 0.5 accuracy.
Saraswathi et al. [25]	The paper combines ResNeXt-50-32x4d architecture with LSTM and concludes the model has higher performance metrics when trained for 40 epochs over 20 epochs when using a combination of FF+, DFDC, and Celeb-DF.	The proposed model selects the first 150 sequential frames; therefore, deepfake content after the 150th frame is ignored.
Shende, Paliwal, and Mahay [62]	The paper proposes a model that combines ResNeXt architecture with LSTM. The results demonstrate an accuracy of 94% on the Celeb-DF dataset.	The proposed model uses a small training and testing set so it is not very generalisable.
Singh et al. [63]	The paper compares backbone networks using the DFDC dataset and demonstrates that EfficientNet-B1 wrapped in a time-distributed layer followed by an LSTM layer resulted in the highest AUC sore and accuracy with 14 million fewer parameters than XceptionNet and InceptionNet backbones, with performances of 86% and 92%, respectively.	There is a trade-off between frame input size and computational power, meaning larger input sizes require significantly more processing power, limiting real-time applications.
Sooda [64]	The paper proposes a model that performed well with an accuracy of 97.25% on compressed videos (c = 23) at 13 epochs, where it became constant. The accuracy increased by 10% from 1 epoch to 13 epochs.	Because the model was trained using only a portion of the DFDC dataset, its accuracy decreased when merged with the FF++ dataset.
Stanciu and Ionescu [65]	The paper integrates Xception, LSTM, and the late fusion approach for combining outputs generated from facial regions to conclude that late fusion does not increase the AUC score using the Celeb-DF or FF++ datasets but provides a high-performing baseline for when only specific facial landmarks are deepfakes.	The model is computationally expensive, limiting its applicability to resource-constrained environments and real-time applications.
Su et al. [66]	The paper proposes adding a soft-attention mechanism based on weights to a CNN with LSTM which reduces the network's calculation costs and increases classification accuracy.	The effectiveness of this approach is highly dependent on the specific characteristics of the dataset and task.
Suratkar and Kazi [67]	The paper uses EfficientNet architecture with LSTM on the DFDC and FF++ datasets and concludes that training a model using residual image autoencoders to capture high-frequency details and improve reconstruction quality gives a higher classification accuracy of 94.75%.	The proposed model is computationally expensive due to the large input size and the long training time.
Taviti et al. [68]	The paper tests ResNeXt-50-32x4d CNN with 3 LSTM layers. The results showed that accuracy stabilised at 60 frames on the Celeb-DF and DFDC datasets at 97%, but when combined with FF++, performance capped at 93% at 100 frames.	Accuracy stabilised at 60 frames per second, meaning efficiency stopped increasing with longer sequences, suggesting a potential bottleneck in the model's architecture.
Wubet [69]	The paper proposed a VGG16 and ResNet-50-32x4d-based CNN-LSTM model that is trained on the MRL eye image dataset and tested on the UADFV dataset. The overall detection accuracy on real videos was over 90%.	The proposed model is unable to detect deepfakes from a diverse set of eyes, suggesting a bias in its training data or architecture.

Table 3. *Cont.*

Citation	Summary of Contribution	Limitations
Yadav et al. [70]	The paper uses transfer learning using InceptionResNetV2 CNN for feature extraction and an LSTM layer, where training for 40 epochs improved accuracy by 6.73% over 20 epochs.	The model's input is the deepfake portion of the clip; therefore, it is not representative of deepfake videos where a real video feed is used.
Yesugade et al. [71]	The paper presents a novel ResNeXt-50-32x4d and LSTM-based deepfake detection model achieving 88.54% accuracy on the DFDC dataset, setting a new benchmark for performance.	The model only considers the first 150 frames as input, significantly reducing its practical usability for analysing longer videos.

RQ2: What are the most common feature extraction techniques used in video deepfake detection tools that utilise CNNs with LSTM?

Many of the feature extraction techniques can be categorised based on the backbone networks they are built upon, where a pre-trained CNN that has been trained on a large dataset for image classification is employed for sequential data analysis. Subsequently, they are followed by an LSTM layer, as LSTM can sequentially process video frames by comparing frames at t seconds with the frame $t - \text{n}$ seconds, where n is any number of frames before t seconds [25].

The most common feature extraction baseline network utilised for video deepfake detection is based on ResNet architecture, which uses depth and width dimensions and residual blocks to enable shortcut 'skip' connections. This is when the gradient signal bypasses layers within the network by performing identity mapping and adding their outputs to the outputs of the stacked layers without adding extra parameters or computational complexity [73]. Therefore, when the network is trained via backpropagation, the vanishing gradient problem is mitigated and high-level features learned by the network are preserved. Hashmi et al. [41] adopt a transfer learning approach, using a pre-trained ResNet CNN as the foundation for feature extraction to identify the spatial locations of features within the frames. A dedicated feature extractor network is derived from identifying a layer within the CNN to dedicate for feature extraction and removing subsequent layers. Integrating this with LSTM, each video frame has its time and patterns memorised and the recurrent nature of the cell computes the temporal dependencies between the frames.

ResNet architecture has several forms, notably, 18-layer and 34-layer plain networks or 50-layer bottleneck models. The bottleneck models are considered more accurate due to their increased depth without the degradation problem [73]. While Jolly et al.'s [46] adoption of the ResNet18 architecture with Gaussian blur is creditable in its performance and ability to remove high-frequency noise to recognise more important characteristics, the choice of a shallower ResNet model may limit its capacity to capture intricate spatial features. Integrating Recycle-GAN to merge spatial and temporal data signifies a step towards addressing this limitation, enhancing the model's ability to learn as it feeds its results back through the network to update its weights and parameters [74], therefore potentially improving detection accuracy. Combining the Recycle-GAN with a deeper ResNet model could optimise its performance. In ResNet-50 architecture, each two-layer block in the 34-layer model is replaced with a three-layer bottleneck block, where the three layers consist of 1 × 1, 3 × 3, and 1 × 1 convolution, which are responsible for reducing and restoring input and output dimensions. Wubet [69] adopts this architecture to classify eye states as open or closed and then sets an eye aspect ratio to detect the eye-blinking rate within the frames followed by LSTM for sequence learning.

ResNeXt architecture is a successor of Resnet that solved the limitations of depth scalability issues whilst maintaining computational efficiency. It introduces a new dimension named cardinality—which refers to the number of parallel paths within a residual block—to improve classification accuracy by leveraging grouped convolutions to capture a more diverse set of features [75]. The widespread use of ResNeXt-50 (32 × 4d) with

LSTM in models [25,32,43,45,57,62,64,68,71,76] indicates its robustness as a baseline model for feature extraction. However, there are narrow architectural differences that present opportunities for comparison. For example, Taviti et al. [68] employ three LSTM layers and ReLU activation, diverging from the single LSTM layer and SoftMax activation used by others, introducing a deeper hierarchical representation of temporal dependencies. Similarly, John and Sherif [45] replace the SoftMax activation function with ReLU within the LSTM layer and use triplet loss detection to calculate the differences between two real images and one fake before updating the weights to reduce loss. These variations suggest there are different strategies for enhancing model performance and generalisation whilst using the same backbone network. Additionally, except for Jalui et al. [43] who use 0.5, the consistent use of a dropout rate of 0.4 at the LSTM layer across all studies emphasises its importance as a regularisation technique to prevent overfitting.

Other frequently used feature extraction techniques are the Xception and Inception networks. Interesting applications of the baseline for feature extraction include using an Xception network to extract the spatial, frequency, and Pattern of Local Gravitational Force (PLGF) features of the facial images [51]. The features are then spliced and fused to obtain 6144 dimensional features of frames that are used as input into a double-layer LSTM. Stanciu and Ionescu [65] also utilise a double-layer LSTM with Xception to extract the features of the mouth, eyes, and nose facial regions separately. Alternatively, variations in inception networks are used in studies [39,40,50,70] before passing feature vectors to LSTM cells, but at the cost of computational complexity.

Based on the residual blocks used in MobileNetV2, EfficientNet uses fixed compound scaling to scale all dimensions of depth, width, and resolution. The usage of EfficientNet for feature extraction varies throughout the studies. Suratkar and Kazi [67] use EfficientNet-B0 and then normalise the features before the feature vectors are passed to the LSTM. Saif et al. [60] utilise EfficientNet-B3 within a two-stream network for pairwise frame comparison, emphasising its effectiveness as a baseline network as the model learns features independent of the deepfake generation method, although this can be accredited to its use of contrastive loss to reduce interclass variations. Su et al. [66] extract faces using an MTCNN but extract the feature vectors using EfficientNet-B5 whilst a soft-attention mechanism is applied in conjunction to determine which of the feature vectors are the most significant. Ritter et al. [58] utilise EfficientNet-B7 with a single LSTM layer to leverage sequential data but it struggles with overfitting. Comparative analysis of the results from Suratkar and Kazi [67] reveal a significantly reduced training time of EfficientNet architecture compared to other techniques, highlighting its potential for real-time video deepfake detection deployment. There are several different feature extraction techniques utilised during the creation of deepfake detection techniques and tools, Table 4 summarises the selected paper's techniques with a description.

Table 4. Various feature extraction techniques used in the selected publications.

Citation	Feature Extraction Technique	Description of the Feature Extraction Technique
Al-Adwan et al. [31]	Hybrid CNN-LSTM with PSO algorithm	The CNN and LSTM are pre-trained on a dataset to extract facial landmarks before the PSO algorithm finetunes the weightings and biases of the CNN and LSTM architecture.
Al-Dhabi and Zhang [32]	Pre-trained CNN (ResNeXt-50) with LSTM	Utilises a ResNeXt-50 CNN model for feature extraction from video frames with an LSTM layer with a 2048 input vector shape and 0.4 ReLU dropout rate to capture temporal discrepancies between frames.
Al-Dulaimi and Kurnaz [33]	Hybrid CNN-LSTM architecture	Utilises three convolutional layers with LSTM with max pooling and the ReLU activation function.

Table 4. *Cont.*

Citation	Feature Extraction Technique	Description of the Feature Extraction Technique
Amerini and Caldelli [34]	Prediction error with convolutional layer and LSTM-RNN ensemble	A bounding box around the face (256 × 256 pixels) is used as input, which is extracted from select video frames. Utilises prediction to detect deepfake faces.
Chan et al. [35]	Triplet LSTM-RNN autoencoder with VGGNet-16 backbone	Utilises a triplet LSTM-RNN autoencoder in conjunction with a VGGNet-16 backbone to capture spatial information within each frame and temporal relationships.
Chen, Li, and Ding [36]	Spatiotemporal attention mechanism with ConvLSTM and Xception CNN	Integrates a spatiotemporal attention mechanism to capture correlations between frames. The video is processed through Xception architecture before ConvLSTM is employed to extract spatiotemporal inconsistency features.
Chinchalkar et al. [37]	RNN-CNN-LSTM hybrid model	Utilises an RNN model trained on 150 frames per video after pre-processing to ensure frame consistency. Frame-level characteristics are extracted using a CNN (ResNeXt), followed by an RNN-LSTM to determine video authenticity.
Chintha et al. [2]	Xception CNN and bidirectional LSTM fusion with Python Dlib	Employs Dlib to locate facial regions within frames followed by linear smoothing filters. Utilises an Xception CNN to extract vector representations of facial regions from frames.
Fuad, Amin, and Ahsan [38]	Wide ResNet-CNN and LSTM attention layer	A pre-trained wide ResNet architecture is used for feature extraction and an LSTM layer to capture temporal data. Data are then input into a multi-head attention layer, so the model selects essential features.
Gravina et al. [39]	InceptionV3 for textural feature extraction	Utilises InceptionV3 CNN pre-trained on the ImageNet dataset for textural feature extraction and implements a separate CNN for emotion feature extraction.
Guera and Delp [40]	Pre-trained CNN (InceptionV3) with LSTM	Each frame's features are extracted using the InceptionV3 model and the feature vectors are used as input into the LSTM.
Hashmi et al. [41]	Transfer learning with ResNet CNN and LSTM	Employs a pre-trained ResNet CNN where subsequent layers are removed to create a dedicated feature extractor network, which is input into an LSTM.
Jaiswal et al. [42]	Hybrid CNN-LSTM-GRU model with Gaussian blur pre-processing	The CNN learns spatial features by extracting frame-wise facial noise with Gaussian blur pre-processing. The extracted features are then fed into a multilayer LSTM-GRU model.
Jalui et al. [43]	Pre-trained CNN (ResNeXt) with LSTM	Adopts a 50-layer ResNeXt CNN architecture for feature extraction. The output of this CNN is a feature vector, which is passed into an LSTM network.
Jindal [44]	Pre-trained CNN (ResNeXt) with LSTM	Utilises ResNeXt-50, analysing the first 150 frames of each video and a 2048 dimensional LSTM.
John and Sherif [45]	ResNeXt CNN with deep triplet loss	ResNeXt architecture extracts temporal information whilst a deep model compares triplet images to distinguish between real or deepfake images.
Jolly et al. [46]	ResNet18 CNN with LSTM and Recycle-GAN fusion	Employs ResNet18 CNN followed by an LSTM layer and a Recycle-GAN two stream fusion to detect spatial–temporal inconsistencies.
Jungare et al. [47]	Pre-trained CNN (ResNeXt) with LSTM	Utilises ResNeXt-50 architecture with LSTM with an input of 150 frames for extraction.
Kaur, Kumar, and Kumaraguru [48]	CNN-LSTM	Utilises three convolutional layers and three max pooling layers for high-level feature extraction from each sequential video frame. The output is flattened and used as input to an LSTM.
Koshy and Mahmood [49]	Nonlinear anisotropic diffusion pre-processing with CNN	Applies nonlinear anisotropic diffusion to the frames of each input sequence for noise reduction, which is then fed into a CNN to capture spatial information, generating an output of 50 features per frame.

Table 4. *Cont.*

Citation	Feature Extraction Technique	Description of the Feature Extraction Technique
Kukanov et al. [50]	Pre-trained CNN (InceptionV3) with LSTM	A pre-trained InceptionV3 CNN is used for feature extraction and LSTM for temporal analysis of 20 frames.
Lai et al. [51]	Xception CNN with double-layer LSTM	Utilises Xception architecture for extracting spatial, frequency, and Pattern of Local Gravitational Force (PLGF) features from facial images.
Li, Chang, and Lyu [52]	Long-term recurrent CNN (LRCRN) with LSTM	Features are extracted using VGG-16 LRCN to find the temporal correlation between frames to detect the duration of the eyes remaining open.
Liang et al. [53]	Facial geometry landmarks module (FGLM) and U-Net	An FGLM acts as an encoder to extract facial landmark information, followed by feature map extraction using a U-Net network.
Malik et al. [54]	CNN-LSTM	Utilises a CNN model where pooling layers extract horizontal and diagonal edge features, and the fully connected layers map these features to the final output before inputting them into an LSTM network.
Masi et al. [14]	Pointwise CNN with bi-LSTM	A bidirectional LSTM processes facial sequences, followed by feature extraction using a pre-trained ImageNet network.
Masud et al. [38]	CNN (VGG-16) wrapped in a time-distributed layer	Utilises a time-distributed layer to wrap the VGG-16 encoder, generating an equal number of feature matrices to the processed frames, using one layer to learn from different frames.
Nawaz, Javed, and Irtaza [55]	Dense-Swish-Net-121 feature descriptor	Proposes a feature descriptor that integrates convolutional layers, dense blocks, and transitional layers with a bi-LSTM approach.
Parayil et al. [56]	Pre-trained CNN (Xception) with LSTM	Uses an Xception CNN for feature extraction using the 2048 dimensional feature vectors after the last pooling layer as sequential LSTM input.
Patel, Chandra, and Jain [57]	Pre-trained CNN (ResNeXt) architecture	Frame-level characteristics are extracted using a ResNeXt CNN model for feature extraction.
Ritter et al. [58]	Pre-trained CNN (EfficientNetB7) with LSTM	This experiment uses an EfficientNetB7 baseline accompanied by one LSTM layer to leverage temporal and sequential information.
Saealal et al. [59]	Pre-trained CNN (VGG-16) with LSTM	Utilises a pre-trained VGG-16 CNN to extract spatial information for each eye, followed by an LSTM for temporal feature extraction from the frame sequence.
Saif et al. [60]	Multi-stream CNN (EfficientNet-B3) with LSTM	Utilises a two-stream network to produce pairwise outputs before extracting features using a time-distributed EfficientNet-B3 CNN and LSTM.
Saikia et al. [61]	Optical flow analysis with CNN-LSTM	The optical flow for pairs of frames, represented as a three-channel image indicating magnitude and direction of motion, is used as input to a CNN-LSTM model with two LSTM layers following convolutional layers for temporal analysis.
Saraswathi et al. [25]	Pre-trained CNN (ResNeXt-50) with LSTM	ResNeXt-50 CNN retrieves the features from the frames, and after the final pooling layer, LSTM is the next sequential layer with 0.4 dropout.
Shende, Paliwal, and Mahay [62]	Pre-trained CNN (ResNeXt-50) with LSTM	Employs a ResNeXt-50 CNN followed by a single layer of LSTM for feature extraction and temporal analysis of frames.
Singh et al. [63]	MobileNet-Single shot detection (SSD)	Integrates MobileNet with SSD, a feed-forward convolutional network for object detection, to extract feature maps and apply convolutional filters.
Sooda [64]	CNN (ResNeXt-50)—GAN with LSTM	A GAN comprises a generator and discriminator, trained with a ResNeXt-50 CNN and LSTM to classify videos.

Table 4. *Cont.*

Citation	Feature Extraction Technique	Description of the Feature Extraction Technique
Stanciu and Ionescu [65]	Xception CNN with LSTM fusion	The Xception CNN outputs 60 feature vectors, which are input into a two-layer LSTM network.
Su et al. [66]	Multitask Cascaded Neural Network (MCNN) with EfficientNet-B5 and LSTM	An MCNN extracts face in pre-processing before an EfficientNet-B5 CNN extracts feature vectors. Subsequently, LSTM detects sequential inconsistencies.
Suratkar and Kazi [67]	EfficientNet CNN with LSTM	Frames are extracted using EfficientNet CNN, and the faces are saved instead of the full frame. The LSTM layer is used as input and is followed by two dense layers to classify the output.
Taviti et al. [68]	Pre-trained CNN (ResNeXt-50) with three LSTM layers	Extracts features using ResNeXt-50 architecture with ReLU to activate the convolutional layers followed by three LSTM layers.
Wubet [69]	VGG-16 and ResNet-50-based CNN with LSTM	A VGG16 and ResNet-50-based CNN model extracts features followed by an LSTM layer.
Yadav et al. [70]	InceptionResNetV2 CNN with LSTM	InceptionResNetV2 CNN is used for feature extraction followed by LSTM for temporal analysis.
Yesugade et al. [71]	Pre-trained CNN (ResNeXt-50) with LSTM	Utilises ResNeXt-50 architecture with LSTM for feature extraction and temporal analysis, with a dropout rate of 0.4 applied to the LSTM layer for regularisation.

There are several similarities between various feature extraction techniques adopted by various researchers. One prominent similarity is the use of pre-trained CNNs, such as ResNeXt, InceptionV3, and Xception, to extract spatial features from individual frames. Many studies, including those by Al-Dhabi and Zhang [32], Jalui et al. [43], and Wubet [69], utilise ResNeXt-50 due to its modular architecture, which makes it adaptable to various tasks and effective in high-level spatial feature extraction. Similarly, the Xception CNN is frequently used by Chen, Li, and Ding [36] and Stanciu and Ionescu [65] for spatial feature extraction, particularly because its depthwise separable convolutions improve computational efficiency without sacrificing accuracy.

Another shared approach across these studies is the integration of LSTM or other RNNs for temporal feature extraction. LSTMs are particularly adept at capturing long-range dependencies between frames, as utilised by Guera and Delp [40], Fuad et al. [38], and others. This combination of CNN for spatial extraction and LSTM for temporal analysis has become a standard practice in the field, given its effectiveness in processing sequential data like videos. Several studies also implement additional layers to enhance feature extraction. For instance, Saikia et al. [61] incorporate optical flow analysis into a CNN-LSTM model to capture motion between frames, adding depth to the temporal analysis. Fuad et al. [38] take this further by integrating a multi-head attention layer, allowing the model to focus on the most relevant temporal features. This highlights the growing trend of using attention mechanisms to improve the performance of hybrid CNN-LSTM models, especially for complex sequential tasks.

While many studies share architectural similarities, there are key differences in the specific models and techniques employed. One notable difference lies in the choice of the CNN backbone. While ResNeXt and Xception are popular, some works like Gravina et al. [39] opt for InceptionV3, focusing on textural feature extraction, which highlights a different aspect of video frames. Additionally, Saif et al. [60] introduced a multi-stream network with EfficientNet-B3, showing that different CNN architectures bring varying degrees of efficiency and accuracy depending on the task requirements. There is also considerable variation in how the models process the frames. Some approaches, like Al-Adwan et al. [31], apply hybrid CNN-LSTM models with PSO for fine-tuning, adding an optimisation layer that can enhance the accuracy of the predictions. This contrasts with models that rely on more straightforward CNN-LSTM architectures without such optimisation techniques, as seen in studies like Patel, Chandra, and Jain [57]. Furthermore, methods like those by Chan et al. [35], which employ a triplet LSTM-RNN autoencoder with a VGGNet-16 backbone, show a shift towards combining autoencoders to capture both spatial and temporal dynamics simultaneously.

Another point of divergence is the pre-processing step before feature extraction. Jaiswal et al. [42] use Gaussian blur to extract facial noise, while Saraswathi et al. [25] rely on frame consistency, showing that pre-processing methods differ greatly depending on the dataset and task focus. Moreover, temporal feature extraction techniques vary beyond LSTM usage. Some approaches incorporate more complex recurrent architectures, such as bidirectional LSTMs (Chintha et al. [2]) or GRU layers (Jaiswal et al. [42]), while others use simpler models. The choice between using unidirectional or bidirectional LSTMs and whether to augment the model with GRUs depends on the depth of temporal analysis required. Lastly, there are subtle variations in the regularisation and optimisation techniques applied. For example, Taviti et al. [68] employ a ReLU activation function and dropout layer, while others, such as Yesugade et al. [71], adopt a dropout rate of 0.4 to prevent overfitting in their LSTM layers. Such differences, though subtle, can significantly impact model performance, depending on the complexity of the dataset. Table 5 summarises the feature extraction methods within the selected papers, categorised based on a baseline network, where the 'original' column signifies a new method.

Table 5. Feature extraction techniques used in the selected publications.

Citation	ResNet	ResNeXt	Xception	InceptionNet	VGG-16	EfficientNet	MobileNet	Image Net	ConvLSTM	Original
Al-Adwan et al. [31]	x	x	x	x	x	x	x	x	x	✓
Al-Dhabi and Zhang [32]	x	✓	x	x	x	x	x	x	x	x
Al-Dulaimi and Kurnaz [33]	x	x	x	x	x	x	x	x	x	✓
Amerini and Caldelli [34]	x	x	x	x	x	x	x	x	x	✓
Chan et al. [35]	x	x	x	x	✓	x	x	x	x	x
Chen, Li, and Ding [36]	x	x	x	x	x	x	x	x	✓	x
Chinchalkar et al. [37]	x	✓	x	x	x	x	x	x	x	x
Chintha et al. [2]	x	x	✓	x	x	x	x	x	✓	x
Fuad, Amin, and Ahsan [38]	✓	x	x	x	x	x	x	x	x	x
Gravina et al. [39]	x	x	x	✓	x	x	x	x	x	x
Guera and Delp [40]	x	x	x	✓	x	x	x	x	x	x
Hashmi et al. [41]	✓	x	x	x	x	x	x	x	✓	x
Jaiswal et al. [42]	x	x	x	x	x	x	x	x	x	✓
Jalui et al. [43]	x	✓	x	x	x	x	x	x	x	x
Jindal [44]	x	✓	x	x	x	x	x	x	x	x
John and Sherif [45]	x	✓	x	x	x	x	x	x	x	x
Jolly et al. [46]	✓	x	x	x	x	x	x	x	x	x
Jungare et al. [47]	x	✓	x	x	x	x	x	x	x	x
Kaur, Kumar, and Kumaraguru [48]	x	x	x	x	x	x	x	x	x	✓
Koshy and Mahmood [49]	x	x	x	✓	x	x	x	x	x	✓
Kukanov et al. [50]	x	x	x	✓	x	x	x	x	x	x
Lai et al. [51]	x	x	✓	x	x	x	x	x	x	x
Li, Chang, and Lyu [52]	x	x	x	x	✓	x	x	x	x	x
Liang et al. [53]	x	x	x	x	x	x	x	x	x	✓
Malik et al. [54]	x	x	x	x	x	x	x	x	x	✓

Table 5. *Cont.*

Citation	ResNet	ResNeXt	Xception	InceptionNet	VGG-16	EfficientNet	MobileNet	Image Net	ConvLSTM	Original
Masi et al. [14]	x	x	x	x	x	x	x	✓	x	x
Masud et al. [38]	x	x	x	x	✓	x	x	x	x	x
Nawaz, Javed, and Irtaza [55]	x	x	x	x	x	x	x	x	x	✓
Parayil et al. [56]	x	x	✓	x	x	x	x	x	x	x
Patel, Chandra, and Jain [57]	x	✓	x	x	x	x	x	x	x	x
Ritter et al. [58]	x	x	x	x	x	✓	x	x	x	x
Saealal et al. [59]	x	x	x	x	✓		x	x	x	x
Saif et al. [60]	x	x	x	x	x	✓	x	✓	x	x
Saikia et al. [61]	x	x	x	x	x	x	x	x	x	✓
Saraswathi et al. [25]	x	✓	x	x	x	x	x	x	x	x
Shende, Paliwal, and Mahay [62]	x	✓	x	x	x	x	x	x	x	x
Singh et al. [63]	x	x	x	x	x	x	✓	x	x	x
Sooda [64]	x	✓	x	x	x	x	x	x	x	x
Stanciu and Ionescu [65]	x	x	✓	x	x	x	x	x	x	x
Su et al. [66]	x	x	x	x	x	✓	x	x	x	x
Suratkar and Kazi [67]	x	x	x	x	x	✓	x	x	x	x
Taviti et al. [68]	x	✓	x	x	x	x	x	x	x	x
Wubet [69]	✓	x	x	x	✓	x	x	x	x	x
Yadav et al. [70]	x	x	x	✓	x	x	x	x	x	x
Yesugade et al. [71]	x	✓	x	x	x	x	x	x	x	x

RQ3: What are the most common datasets used in the implementation of video deepfake detection tools that utilise CNN with LSTM?

Datasets play a crucial role in training ML models to identify altered or synthesised content. High-quality and diverse datasets are essential because deepfake videos often involve subtle manipulations that can easily deceive the human eye. These manipulations may include alterations in facial expressions, speech, or body movements, which are challenging to detect without robust data-driven models. By exposing detection algorithms to a wide range of fake and authentic videos, the models can learn to differentiate between genuine and manipulated content, improving the overall accuracy of deepfake detection. The importance of datasets also extends to their ability to generalise across different types of deepfakes [77]. As deepfake generation techniques evolve, new and more sophisticated methods emerge, making it vital for detection models to remain adaptable. Datasets that capture diverse deepfake techniques allow for a more comprehensive training process, enabling the model to detect deepfakes created using various algorithms and approaches. This adaptability is critical in the real-world application of these technologies, where deepfakes may be used in malicious contexts such as misinformation, identity fraud, or cybercrime. Moreover, the continuous expansion and updating of datasets ensure that deepfake detection systems stay relevant and effective. As deepfake creation methods become more advanced, detection systems must keep pace by learning from up-to-date datasets. Figure 7 illustrates the frequency of the common datasets used across the literature.

The most prominent dataset used within the selected papers was the DFDC dataset, which was used 25 times. DFDC is a deepfake dataset created by Facebook for a deepfake detection challenge with a partial set containing 4113 deepfake videos and is based on an unknown creation algorithm [78]. All the faces collected are actors with informed consent, which the Celeb-DF and FaceForensics++ datasets lack. The results from Jalui et al. [43] determine that CNN-LSTM models can achieve strong accuracy on the DFDC dataset alone, at 96% on unseen data, but it uses a very small sample size and therefore is not generalisable.

The FaceForensics++ (FF++) dataset [79] contains 1000 videos split into 509,914 images (frames) sourced from YouTube videos and were manipulated using Face2Face, FaceSwap, and neural texture and deepfake tools to create the deepfake content. This was the second most prominent dataset utilised by authors, with 24 uses. The subjects within the frames are forward-facing and do not contain occlusions to maximise detection. The results from Saealal et al. [59] and Saif et al. [60] both concluded that their models performed the lowest on the Neural Textures subset of the FF++ dataset, indicating that the models perform [80] strongest on GAN-generated deepfakes. As this is the primary creation method of deepfakes, this causes little concern. Taviti et al. [68] determined that their model had the strongest performance using 100 frames on the FF++ dataset at 97.89%. Combining the dataset with Celeb-DF and DFDC during testing caused performance to drop to 93%.

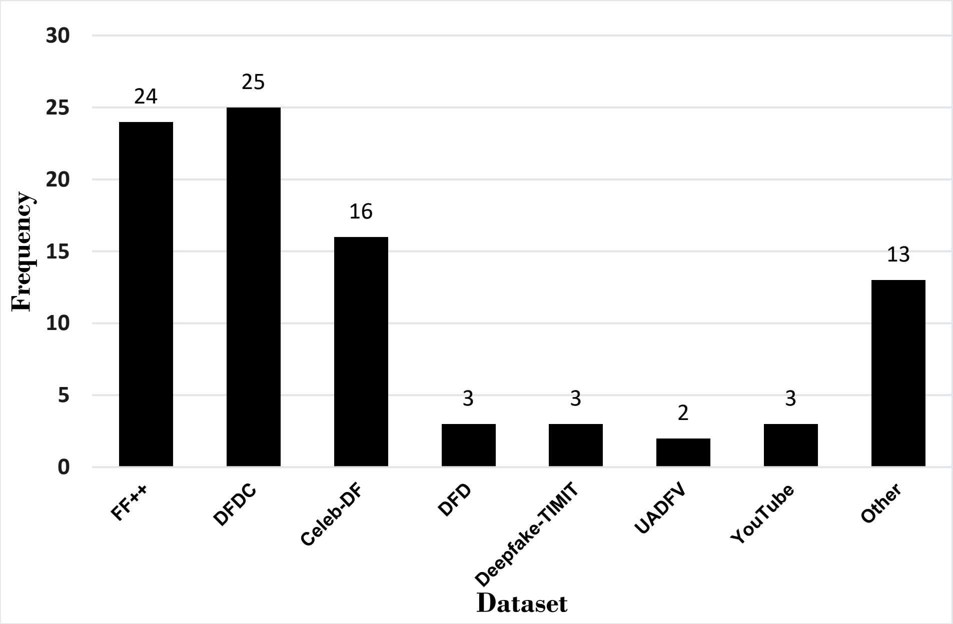

Figure 7. Common datasets used in video deepfake detection.

The Celeb-DF dataset contains 590 videos from YouTube and 5639 deepfake videos of celebrities split into over 2 million frames. The dataset was utilised 16 times within the training and testing phases in the selected papers. The dataset contains 59 celebrities of various ages, genders, and ethnicities to maximise diversity and detection. The deepfakes are generated using a deepfake synthesis algorithm with a high resolution, reduced colour mismatch, and reduced temporal flickering using a Kalman smoothing algorithm [81]. Research from Masud et al. [38] explores how the Celeb-DF dataset has a large class imbalance, with the majority of the content being deepfake, so augmentation and re-sampling has to occur to ensure an equal balance between real and deepfake videos during training. Therefore, studies that use this dataset alone will learn fewer deepfake features during the feature extraction process. This is reinforced by Shende, Paliwal, and Mahay [62], who although 94.21% detection accuracy was reached, concluded that the dataset is not very generalisable.

Deepfake-TIMIT is a dataset of deepfakes generated from GANs, consisting of 10 videos of 32 people, with low-quality and high-quality options [82]. The dataset has no manipulation of the audio channel; therefore, it is not a good dataset if an audio–visual detection method is desired. However, as most deepfakes are generated using GANs, the dataset is a good baseline for video detection. This dataset appeared three times within the papers. Experimental results from Lai et al. [51] on different baselines determined that the Deepfake-TIMIT dataset produced a consistently high false alarm rate and missed detection rate compared to the DFD and FF++ datasets. A robust model must not have a high false-positive rate; therefore, this dataset is not suitable for training models alone.

The UADFV dataset has 98 videos, with 49 real and 49 deepfake videos [83], which is not computationally expensive but is not a large dataset to train from. The real videos are collected from YouTube and the deepfakes are generated using FakeAPP [84], which were utilised twice within the papers. This dataset is small and therefore will learn intricate feature patterns that are not generalisable to unseen content. Authors have also created original datasets to use for tests, with several authors collecting YouTube videos. Although this is good for gathering compressed data, which applies to the general use of deepfakes, there may be concerns surrounding data collection consent. The results from Kukanov et al. [50] failed to detect deepfake content from YouTube videos, which is viewed as the 'worst case', although it is the most readily available deepfake content online. Figure 8 is a Sankey diagram, demonstrating the usage of each dataset and the correlation of the datasets each paper used. Several datasets were used per model in many cases, as they tested and trained on different datasets.

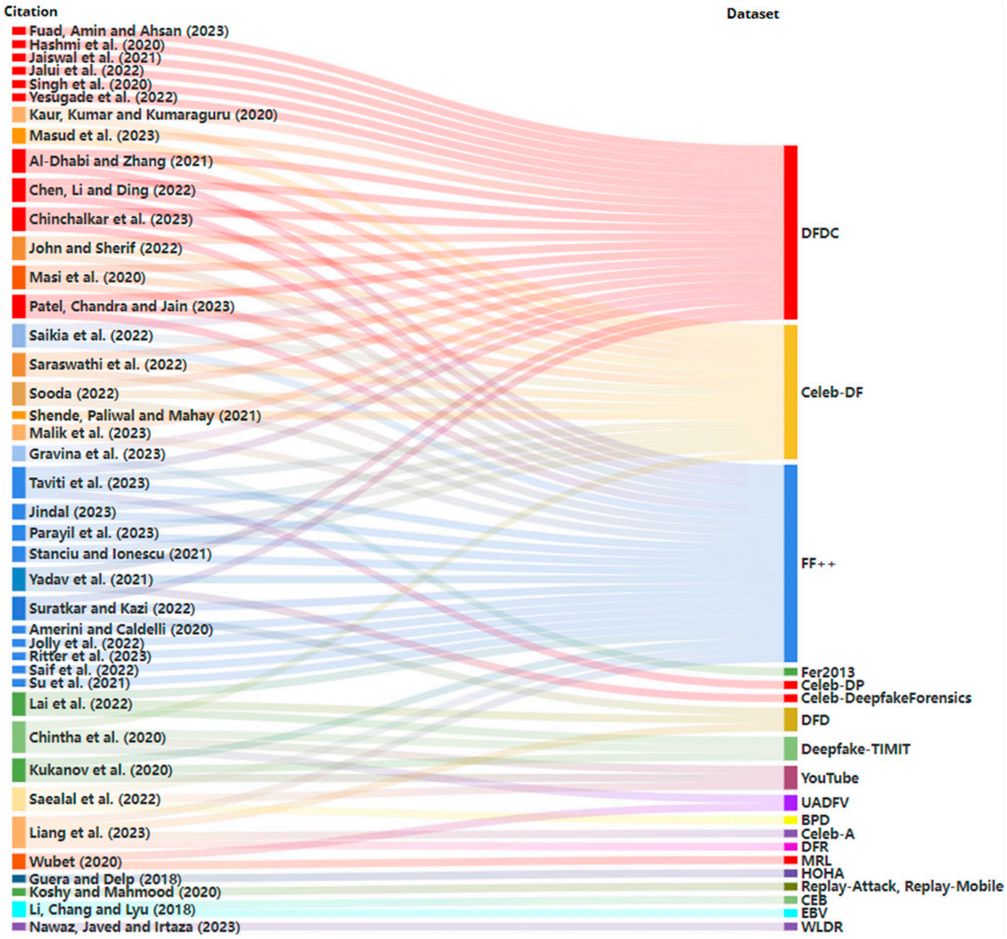

Figure 8. Video deepfake datasets used within the selected publications [2,14,25,32,34,36–46,48–71].

The use of different datasets causes limitations for model performance and generalisability. As observed, datasets like DFDC and FF++ are widely used due to their size and diversity but they differ in manipulation types used to generate deepfake content, therefore impacting the quality of the videos that the model trains on and their detection. These variations make it difficult to compare results across models that have been trained on different datasets, as each model learns a specific subset of deepfake techniques, for example, Saikia et al. [61] achieve 91% accuracy on the FF++ dataset, but 61% on DFDC. If models are only trained on one dataset, they can struggle to generalise to others during testing, particularly when those datasets contain deepfakes generated using unfamiliar methods (VAE, GAN) or contain elements such as occlusions, facial rotations, or a change in resolution. Combining datasets to enhance generalisability exposes the CNN-LSTM model to a larger set of real and deepfake videos to extract features from but could introduce overfitting or a class imbalance to a specific type of deepfake generation method, therefore requiring even sets of generation methods to be input into the model.

A primary reason for the lack of generalisability is that these models often overfit the specific features of the datasets they are trained on. For instance, models might learn to recognise deepfakes based on dataset-specific artefacts, such as particular lighting conditions or facial attributes, rather than general patterns that could apply across all types of deepfakes. Additionally, the scarcity of large, diverse, and standardised benchmark datasets limits the robustness of these models. Addressing this challenge requires expanding the diversity of training datasets and employing techniques like transfer learning and domain adaptation. These approaches could improve the model's ability to adapt to new data without requiring extensive retraining on each new dataset. Furthermore, incorporating multimodal approaches, which analyse both visual and audio cues, could enhance the generalisation of CNN-LSTM models, making them more robust in real-world applications where deepfake content varies widely.

RQ4: What are the factors that have the strongest influence on detection accuracy for video deepfake detection when implementing CNN with LSTM?

After reviewing the selected research papers, the strongest factors influencing detection accuracy for video deepfake detection when implementing CNN with LSTM include the LSTM architecture, training rate, frame quality, and the presence of facial landmarks.

LSTM architecture has a significant influence on detection accuracy for video deepfake detection because it excels in capturing temporal dependencies in sequential data. Since deepfakes often involve dynamic, frame-by-frame manipulations of videos, LSTM models are particularly well suited for detecting inconsistencies that occur over time, such as unnatural movements or frame transitions. Unlike traditional convolutional networks that focus on spatial features within individual frames, LSTMs analyse the video as a sequence, allowing the model to learn the patterns of both natural and manipulated content over time. By doing so, LSTM networks can detect subtle anomalies in facial expressions, lip movements, or eye blinks that may be temporally inconsistent with natural human behaviour, leading to higher detection accuracy. The integration of bidirectional LSTM with CNN has a significant influence on improving the detection accuracy of video deepfake detection models. The use of bidirectional LSTM layers, which extract temporal data through a backward pass, resulted in higher detection accuracy compared to unidirectional LSTM layers [2]. For example, on the FF++ and Celeb-DF datasets, the bidirectional LSTM approach achieved detection accuracy values of 99.7% and 97%, respectively, outperforming the unidirectional LSTM which only attained 92% and 84%. Similar findings by Gravina et al. [39] stated that utilising bidirectional LSTM layers to represent the sequential nature of video data mitigates overfitting and leads to a higher prediction accuracy than traditional LSTM approaches. Although their model did not outperform all state-of-the-art methods, it achieved robust results with a significantly smaller number of trainable parameters, at around 59 million whilst incorporating emotional features. Furthermore, the findings of Nawaz, Javed, and Irtaza [55] leverage the ability of dense connections to propagate negative scores during feature extraction to extract a more dense and informative set of visual characteristics, which, when coupled with bi-LSTM for temporal correlation, results in a substantial increase in classification accuracy compared to the VGG16-based LSTM approach. The DenseNet121 and GoogleNet models also benefited from the integration of bi-LSTM, achieving detection accuracies of 98.11% and 97.91%, respectively. Therefore, the combination of CNN and Bi-LSTM can increase detection accuracy for video deepfake detection despite its high computational cost.

Training rate, often referred to as the learning rate, also has a great influence on detection accuracy for video deepfake detection because it directly affects the model's ability to learn patterns effectively during the training process. A well-tuned training rate ensures that the model converges at an optimal pace, striking a balance between learning from the data and avoiding overfitting. If the rate is too high, the model may miss important nuances and fail to capture subtle differences between real and fake video frames, leading to poor detection accuracy. Conversely, if the rate is too low, the model may converge too slowly or get stuck in local minima, resulting in underperformance. Therefore, finding the optimal training rate is essential for improving the model's accuracy, as it allows the network to efficiently learn the complex patterns and temporal dependencies needed to distinguish between genuine and deepfake videos. The training–test split and the number of epochs used during training were found to impact the detection accuracy of the LSTM-CNN models. An epoch is a complete pass of the algorithm through an entire training set, whilst a batch size is the number of samples processed before the model updates [85]. Patel, Chandra, and Jain [57] observed that as the number of epochs and frame sequences increased, the model's accuracy improved while the training loss decreased. Similar results by Sooda [64] found that increasing the number of epochs from 5 to 13 led to a boost in accuracy, from 91.35% to 97.25%. Findings from Saraswathi et al. [25] illustrated that the LSTM-based model showed higher accuracy when trained for 40 epochs (90.37%) compared to 20 epochs (84.18%), though there was a fairly high false negative rate in both cases. Further research from Yadav et al. [70] confirmed this finding, with the 40-epoch model achieving 91.48% accuracy versus 84.75% for the 20-epoch model. This demonstrates that more training time allows the model to better learn the spatial and temporal features necessary for robust video deepfake detection.

Frame rate plays a critical role in influencing detection accuracy for video deepfake detection because it determines the temporal resolution of the video, which is essential for identifying inconsistencies across frames. A higher frame rate provides more data points for the model to analyse, allowing it to capture subtle temporal anomalies such as unnatural transitions, mismatched lip movements, or irregular blinking patterns that are often indicative of deepfake videos. The existing research demonstrates that the number of video frames used as input to the CNN-LSTM model has a significant impact on its accuracy. The results from Al-Dhabi and Zhang [32] showed that using 100 frames resulted in a high training accuracy of 99.93% and validation accuracy of 95%, but when the number of frames was reduced to just 10, the training accuracy dropped to 86.75% and test accuracy to 84%. Similar research by Jindal [44] deduced that detection accuracy increased as the number of frames used increased, with 100 frames yielding 93.5% accuracy compared to 84.2% for 10 frames. This suggests that the model cannot determine sufficient temporal discrepancies with a smaller frame rate, which is reinforced by Saikia et al. [61] who achieved an accuracy of 0.5 across all datasets when using 20 frames. The findings of the research by Chan et al. [35] also show that using too few frames leads to insufficient temporal correlations being captured, while too many frames cause feature redundancy with bias patterns being learned. Their research indicated that 30 frames were ideal, with overfitting and performance loss occurring below that threshold. Likewise, Guera and Delp [40] reported that higher frame rates improved accuracy, as the model was able to learn temporal relationships in a sequential, end-to-end manner, concluding 40 frames to be sufficient for video deepfake. However, using only 10 frames, Jungare et al. [47] achieved an accuracy of 84.21%, which increased to 93.58% using 100 frames. Overall, the evidence suggests that the number of frames used as input impacts the detection accuracy as the model can better detect temporal variances for classification, with the ideal range of frames for detection being between 30 and 100, depending on the available resources.

Another factor that impacted detection accuracy was the quality of the frames taken from the video, as high-quality frames provide clear and detailed visual data, which is critical for identifying subtle manipulations. Deepfakes often involve minor distortions in facial features, skin texture, or lighting conditions that may be imperceptible in low-resolution or poor-quality frames. High-quality frames preserve the fine details necessary for detection models, especially CNNs, to capture these anomalies and make accurate distinctions between real and fake content. Masi et al. [14] discovered their model performed 10% better on high-quality videos from the DFDC dataset compared to low-quality videos and also achieved higher AUC scores for medium compression levels versus high compression. They attribute these improvements to incorporating minor training optimisations, including assigning different update rates per network layer and using dropout, as well as their proposed loss function. Similarly, Kukanov et al. [50]

observed that their detection methods were able to correctly classify videos with lower deepfake quality but struggled with higher-quality manipulations. Fuad, Amin, and Ahsan [38] also reported that their model achieved better accuracy and precision when operating on cropped and resized face frames at 512 × 512 resolution, rather than full video frames. Collectively, these findings underscore how CNN-LSTM models require input pre-processing to reach their optimal detection accuracy. Additionally, they are robust to higher-quality images for detection but require further work to tune the models to have a stronger performance on lower-quality deepfakes. The findings by Gravina et al. [39] suggest applying Contrast Limited Adaptive Histogram Equalization (CLAHE) during pre-processing to improve the contrast in images, enhancing the definitions of edges in each region, meaning the CNN can detect facial features more effectively in the feature extraction stage.

Facial landmarks also have a strong influence on detection accuracy for video deepfake detection because they serve as key reference points for identifying natural facial movements and expressions. Deepfake algorithms often struggle to perfectly replicate the subtle dynamics of facial features, such as the alignment of eyes, nose, mouth, and jawline during speech or expression changes. By focusing on these landmarks, detection models can identify irregularities in how these points shift or interact over time, exposing unnatural distortions that are indicative of manipulation. The precise tracking of facial landmarks allows the model to detect even minor inconsistencies in facial behaviour, significantly boosting the accuracy of deepfake detection systems. Hashmi [41] determines that detection accuracy increases for every facial landmark introduced, only performing with 26.8% accuracy using eye blinking when using an MTCNN with LSTM but 94% accuracy when the mouth movement is included. Other studies reinforce these conclusions [65], establishing that training individual facial regions and applying late fusion does not yield higher accuracy in comparison to training on the full face, although detection of individual facial features on the FF++ dataset resulted in 0.98 AUC scores. Therefore, this detection method is not robust for entire face synthesis or identity swap deepfake methods but could be applied to detecting attribute manipulation. Comparably, other research by Su et al. [66] deduced the mouth and eyes to be the most important features in detection, concluding that performance increases significantly when a particular feature of the facial data is strengthened through the removal of other features. Furthermore, applying changes such as Gaussian blur and Wavelet transform to the pre-processed inputs improves classification accuracy when using a ConvLSTM compared to its original input as image noise is reduced, improving performance for low-quality deepfakes. Also, research by Chinchalkar et al. [37] deduces noise as an identifier of deepfake content, concluding that reconstructed deepfake videos have a low Signal-To-Noise ratio. Therefore, further research into using noise as a detection measure could improve the accuracy of deepfake detection models. Table 6 presents the factors that influenced the detection accuracy the most for each of the selected publications. Where 'Other' is selected, the factor is either minimal or applicable to a specific type of CNN-LSTM architecture.

RQ5: Is CNN with LSTM more effective for video deepfake detection compared to models that do not utilise LSTM?

Researchers utilise many performance metrics to determine how successfully a model can correctly classify unseen input as real or deepfake. The most commonly used metric is accuracy to determine how often the outcome is predicted correctly.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Number of total predictions}} \times 100$$

A robust model should be accurate, detecting faces with no false positives or negatives. Another measure is the Area Under the Receiver Operating Characteristics Curve (AUC), which combines the true positive rate and false positive rate using logistic regression. A true positive rate refers to the number of positive class samples the model correctly predicted, whilst a false positive rate refers to the number of negative class samples the model predicted incorrectly. A true negative rate refers to the number of negative class samples the model correctly predicted, and a false negative rate refers to the number of positive class samples the model incorrectly predicted. These values are often used in confusion matrices, as seen in Figure 9, to represent how correctly data are classified. With these values, further calculations such as precision, recall, and the F1 score can be computed.

Table 6. Factors that influence video deepfake detection accuracy.

Citation	LSTM Architecture	Training Rate	Frame Rate	Frame Quality	Facial Landmark
Al-Adwan et al. [31]	x	✓	x	x	x
Al-Dhabi and Zhang [32]	x	x	✓	x	x
Al-Dulaimi and Kurnaz [33]	✓	x	x	x	x
Amerini and Caldelli [34]	x	x	x	✓	x
Chan et al. [35]	x	x	x	x	x
Chen, Li, and Ding [36]	x	x	✓	x	x
Chinchalkar et al. [37]	x	x	x	x	✓
Chintha et al. [2]	✓	x	x	x	x
Fuad, Amin, and Ahsan [38]	x	x	x	✓	x
Gravina et al. [39]	✓	x	x	x	x
Guera and Delp [40]	x	x	✓	x	x
Hashmi et al. [41]	x	x	x	x	✓
Jalui et al. [43]	x	x	✓	x	x
Jindal [44]	x	x	✓	x	x
John and Sherif [45]	x	x	✓	x	x
Jungare et al. [47]	x	x	✓	x	x
Kaur, Kumar, and Kumaraguru [48]	x	x	x	x	✓
Kukanov et al. [50]	x	x	x	✓	x
Lai et al. [51]	✓	x	x	x	x
Li, Chang, and Lyu [52]	x	x	x	x	✓
Liang et al. [53]	x	x	x	x	✓
Malik et al. [54]	x	✓	x	x	x
Masi et al. [14]	x	x	x	✓	x
Masud et al. [38]	✓	x	x	x	x
Nawaz, Javed, and Irtaza [55]	✓	x	x	x	x
Parayil et al. [56]	x	x	✓	x	x
Patel, Chandra, and Jain [57]	x	✓	x	x	x
Ritter et al. [58]	x	✓	x	x	x
Saealal et al. [59]	x	✓	x	x	x
Saikia et al. [61]	x	x	✓	x	x
Saraswathi et al. [25]	x	✓	✓	x	x
Shende, Paliwal, and Mahay [62]	x	x	✓	x	x
Singh et al. [63]	x	x	✓	x	x
Sooda [64]	x	✓	x	x	x
Stanciu and Ionescu [65]	x	x	x	x	✓
Su et al. (2021) [66]	x	x	✓	x	x
Taviti et al. [68]	x	x	✓	x	x
Wubet [69]	x	x	x	✓	✓
Yadav et al. [70]	x	✓	x	x	x
Yesugade et al. [71]	x	x	✓	x	x

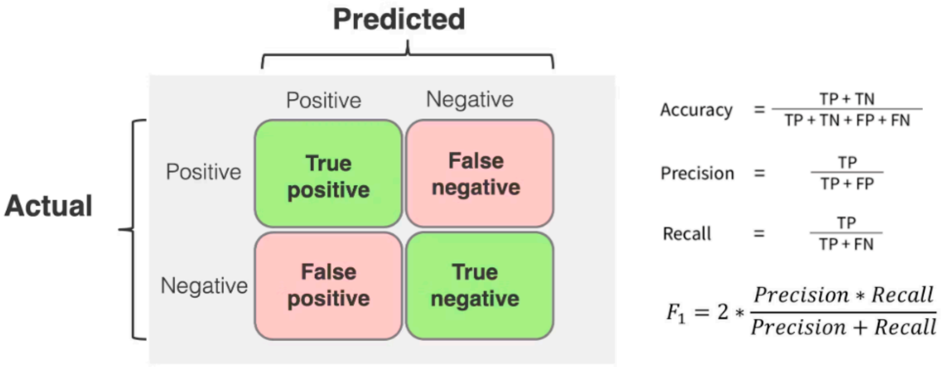

Figure 9. Confusion matrix.

Some researchers utilise the Half Total Error Rate (HTER), which is the average value of the false alarm rate and missed detection rate under a decision threshold. A lower HTER score indicates a stronger model performance. Similarly, the detection cost function (DFC) is a weighted sum of the missed detection rate and false alarm rate [50]. Using a range of measures to evaluate performance can provide a more nuanced understanding of the model's abilities to classify the data correctly. However, using multiple metrics can make direct comparisons between models more challenging as they determine different things. For example, AUC measures a model's sensitivity and specificity, but accuracy does not distinguish between these variables. Furthermore, accuracy is a weak measure for imbalanced datasets. Table 7 compares the performance of the selected models, with a comparison to a CNN model alone if provided by the author. The dataset used during testing, the performance metric, and the network backbone are provided. Where a range of backbones were tested, the highest performance metric is given. The frame rate also produced standardisation problems when incorporating LSTM, as some frames were reported as a sequence length whilst others were reported as 'frames per second', which was defaulted to if available.

Table 7. Performance comparison of selected publications.

Citation	Backbone	CNN	CNN-LSTM	Metric	Dataset Used	Frame Rate
Al-Adwan et al. [31]	Original	- -	97.40% 94.47%	ACC	Celeb-DF DFDC	-
Al-Dhabi and Zhang [32]	ResNeXt	-	95.54%	ACC	FF++, DFDC, Celeb-DF	80
Al-Dulaimi and Kurnaz [33]	Original	-	98.24%	ACC	DFDC	-
Amerini and Caldelli [34]	Original	91.72% 79.41%	94.29% 85.27%	ACC ACC	FF++ (lossless) FF++ (c23)	- -
Chen, Li, and Ding [36]	ConvLSTM	- - -	99% 99.93% 92.43%	ACC ACC ACC	FF++ Celeb-DF DFDC	30
Chinchalkar et al. [37]	ResNeXt	-	98%	ACC	FF++, DFDC, Celeb-DF	150
Chintha et al. [2]	Xception	96.71% 89.91%	100% 99.16%	ACC ACC	FF++ Celeb-DF	-
Fuad, Amin, and Ahsan [38]	ResNet	70%	82.4%	ACC	DFDC	150
Gravina et al. [39]	InceptionNet	-	84.26%	ACC	DFDC	-

Table 7. *Cont.*

Citation	Backbone	CNN	CNN-LSTM	Metric	Dataset Used	Frame Rate
Guera and Delp [40]	InceptionNet	-	97.1%	ACC	HOHA	80
Hashmi et al. [41]	ResNet, ConvLSTM	-	97.5%	ACC	DFDC, Celeb-DF, YT	-
Jaiswal et al. [42]	Original	-	80%	ACC	DFDC	-
Jalui et al. [43]	ResNeXt	-	97.27%	ACC	DFDC	-
Jindal [44]	ResNeXt	-	93.59%	ACC	FF++, Celeb-DF	100
John and Sherif [45]	ResNeXt	-	94.31%	ACC	DFDC	100
Jolly et al. [46]	ResNet	-	95.73%	ACC	FF++(HQ), YT	-
Jungare et al. [47]	ResNeXt	-	93.58%	ACC	DFDC, FF++, Celeb-DF	100
Kaur, Kumar, and Kumaraguru [48]	Original	97.06%	98.21%	ACC	DFDC	-
Koshy and Mahmood [49]	InceptionNet	- -	98.71% 95.41%	ACC ACC	Replay Attack Replay Mobile	20
Kukanov et al. [50]	InceptionNet	-	38.90	DCF	FF++, Deepfake-TIMIT, YT	20
Lai et al. [51]	Xception	18.84 22.31 4.38 10.60	14.17 18.32 2.91 15.34	HTER HTER HTER HTER	FF++ FF++(c23) DFD (c23) Deepfake-TIMIT	10
Li, Chang, and Lyu [52]	VGG-16	0.98	0.99	AUC	EBV, CEW	30
Liang et al. [53]	Original	98% 91% 63%	99% 94% 75%	ACC ACC ACC	FF++ DFD Celeb-DF	-
Malik et al. [54]	Original	82% 75%	66% 72%	ACC ACC	FF++ DFDC	-
Masi et al. [14]	Image Net	0.92 0.86	0.99 0.91	AUC AUC	FF++ (c23) FF++ (c40)	110
Masud et al. [38]	VGG-16	-	99.24%	ACC	Celeb-DF	20
Nawaz, Javed, and Irtaza [50]	Original	-	99.31%	ACC	DFDC	-
Parayil et al. [56]	Xception	-	40%	ACC	DFDC	-
Patel, Chandra, and Jain [57]	ResNeXt	-	92.12%	ACC	FF++, DFDC, Celeb-DF	100
Ritter et al. [58]	EfficientNet	-	75%	ACC	FF++	-
Saealal et al. [59]	VGG-16	-	95.57%	ACC	FF++	-
Saif et al. [60]	EfficientNet, ImageNet	-	99.6%	ACC	FF++	5
Saikia et al. [61]	Original	0.89 0.69 0.64	0.91 0.83 0.68	AUC AUC AUC	FF++ Celeb-DF DFDC	30 50 20
Saraswathi et al. [25]	ResNeXt	-	90.37%	ACC	FF++, DFD, Celeb-DF	30
Shende, Paliwal, and Mahay [62]	ResNeXt	-	94.21%	ACC	Celeb-DF	30

Table 7. *Cont.*

Citation	Backbone	CNN	CNN-LSTM	Metric	Dataset Used	Frame Rate
Singh et al. [63]	MobileNet	-	97.63%	ACC	DFDC	30
Stanciu and Ionescu [65]	Xception	99.40 83.60	99.95 97.06	AUC AUC	FF++ Celeb-DF	60
Su et al. [86]	EfficientNet	-	99.57%	ACC	FF++	15
Suratkar and Kazi [67]	EfficientNet	98.69% 85.84%	97.56% 81.23%	ACC ACC	DFDC FF++	20
Taviti et al. [68]	ResNeXt	- - - -	97.89% 97.79% 97.75% 93.29%	ACC ACC ACC ACC	FF++ Celeb-DF DFDC FF++, Celeb-DF, DFDC	100
Wubet [69]	ResNet	- -	93.23% 98.30%	ACC ACC	UADFV (real) UADFV(fake)	28
Yadav et al. [70]	InceptionNet	-	91.48%	ACC	FF++, Celeb-DF, DFDC	-
Yesugade et al. [71]	ResNeXt	-	88.55%	ACC	DFDC	30

Numerous results indicate that the combined CNN-LSTM architecture can outperform CNN-only models. Amereni and Caldelli [34] found that the classification accuracy was highest (94.29%) when using an LSTM-based model compared to 91.72% for a CNN-only model, demonstrating the benefits of exploiting sequence correlations across frames. Similarly, Chen, Li, and Ding [36] found that combining a spatial attention mechanism with an Xception backbone and ConvLSTM achieved an impressive 0.99 AUC score, outperforming the Xception CNN without LSTM, which achieved 70.10% accuracy [79]. Additionally, the C-LSTM model from Kaur, Kumar, and Kumaraguru [48] outperforms MesoNet [87] and CNN state-of-the-art models, achieving an accuracy of 98.21% in a lower training time than the CNN model whilst also having the lowest loss value.

The findings of Fuad et al. [38] combining wide ResNet architecture and LSTM achieved the same accuracy as ResNet alone on frames of a 256 × 256 resolution but performed 9% better when 512 × 512 resolution frames were used as input and 12% better on the DFDC test set, achieving 82.4%. The results from Masi et al. [14] achieved an AUC score of 81.53 on the FF++ with strong compression (c40) enabled, while achieving 76.04 without LSTM, therefore demonstrating that LSTM improves detection when other fine-tuning methods are added, including updating rate per layer and adding dropout. In particular, almost all frame-level performance improved for the medium compression (c23) videos, increasing the video-level AUC score from 92% to 99% on the overall FF++ dataset. Achieving a 97.26 AUC score using an EfficientNet CNN-LSTM architecture with transfer learning on the DFDC dataset, the results of Suratkar and Kazi [67] slightly improve the 97.18% AUC score achieved by the DFDC competition winner [88], who did not utilise LSTM in their EfficientNet CNN implementation.

Saika et al. [61] compare the performance of the base models with optical flow, LSTM, and CNN on the Celeb-DF, DFDC, and FF++ datasets. When optical flow and LSTM are combined, the model cannot differentiate well and produces its worst performance with an average AUC score of 0.5 on all datasets; therefore, the classifier cannot distinguish between positive and negative class points effectively. When using optical flow as input to the CNN and training without LSTM, the model performs better on the Celeb-DF and DFDC datasets with 0.83 and 0.68 AUC scores, respectively. However, combining optical flow, LSTM, and CNN produces the strongest performance for the FF++ dataset with an AUC score of 0.91. When detecting specific facial landmarks, such as the eyes, using a standalone CN to train the pixel pattern of the eyes is generally unable to determine whether the eyes are in an open or closed state; therefore, using an RNN for temporal features can improve the model. The results in this domain by Saealal et al. [59] achieved up to 95.57% accuracy. Other findings by Li et al. [52] reinforce this, as the LRCN performs at 0.99 AUC compared to 0.98 AUC using a CNN, as the long-term dynamics improve the detection of the state.

The studies discussed highlight the advantages of CNN-LSTM models over CNN-only approaches, particularly in exploiting temporal dependencies for improved accuracy. Amereni and Caldelli [34] and Chen, Li, and Ding [36] both report higher classification accuracy and AUC scores when combining CNN with LSTM, demonstrating that LSTM enhances temporal feature learning across frames. Similarly, Kaur, Kumar, and Kumaraguru [48] found that CNN-LSTM outperforms state-of-the-art CNN models with lower loss and faster training times. Fuad et al. [38] further emphasise that LSTM's performance gains are more significant with higher-resolution input data. However, Saika et al. [61] show that while CNN-LSTM models perform well on specific datasets like FF++, combining LSTM with optical flow can reduce performance on others. Saealal et al. [59] and Li et al. [52] reinforce LSTM's strengths, particularly in tasks involving temporal features like facial recognition, where RNN-based models achieve superior results. Overall, CNN-LSTM models generally outperform CNN-only models but are dataset-dependent.

RQ6: What are the challenges involved in implementing video deepfake detection tools that utilise CNN with LSTM?

One of the key challenges in deepfake detection models that combine CNN and LSTM architectures is the trade-off between computational efficiency and model performance. Chen, Li, and Ding [36] found that their CNN-LSTM model had an increased runtime compared to other algorithms due to the non-parallelisable convolutional operations of ConvLSTM, resulting in a worst-case time–space complexity of O(n^2). This complexity implies that as the input size or model complexity grows, computational costs can increase quadratically. Similarly, other studies [37,44,63] highlighted the burden of large processing times when networks are required to learn from thousands of input images, creating a challenge in deploying these models in resource-constrained or real-time environments. Addressing this issue by optimising the architecture is essential to maintain a balance between performance and computational feasibility. Although feature extraction learns more from detailed imagery, testing the parameters and reducing the input size by using smaller images and a lower frame rate helps equilibrate the trade-off when access to larger computational power is not available

The computational trade-offs in CNN-LSTM models for deepfake detection are critical, particularly for real-time applications. These models combine spatial and temporal analysis, requiring significant computational resources, which can hinder their ability to function efficiently in real-time scenarios. High computational costs result from processing large video datasets, the complexity of CNN layers for feature extraction, and the LSTM's sequential frame analysis. To enable real-time detection, there is often a need to simplify the model, reduce the input data, or limit the depth of Neural Networks. However, these optimisations can negatively impact the accuracy of deepfake detection, as reducing model complexity typically leads to less effective feature extraction and pattern recognition. Balancing accuracy with speed is thus a key challenge, especially when considering deployment in environments with limited computational resources, such as mobile devices or real-time video streaming platforms.

Furthermore, real-time deepfake detection leveraging CNN-LSTM architectures is often constrained by the availability of hardware acceleration, such as GPUs, which can significantly influence detection viability. Hardware acceleration enhances real-time speed whilst reducing latency and power consumption, which is ideal for real-time detection [61]. Where this is not available, optimising models to function on less powerful hardware using lightweight CNNs such as MobileNet [62] or Efficient [63] is effective. The computational strain of CNNs can be reduced using model compression through techniques like model pruning or quantisation, which lower the precision of weights and reduce memory usage. Compressing the CNN-LSTM model refers to reducing the size of the model without sacrificing performance. Pruning the model uses deep learning to remove less important weights and connections from the network to reduce the number of operations [64]. Quantisation involves reducing the precision of the weights to reduce computational power and reduce memory between nodes, therefore also enhancing processing speeds [64]. However, aggressive compression can lead to a reduction in the models ability to detect fine-grain details. In time-sensitive scenarios, such as real-time live streams, these trade-offs can make the distinction between effective and ineffective detection systems.

Another significant challenge relates to the limitations of frame selection in deepfake detection models. One of the main issues with frame selection is the inherent variability in deepfake videos. These videos often employ sophisticated techniques to manipulate specific frames, making certain frames more susceptible to detection than others. Consequently, if the selected frames do not adequately represent the video as a whole, the model may struggle to identify subtle anomalies that indicate manipulation. For instance, a deepfake might appear convincing in a few selected frames but exhibit detectable artefacts in others, such as unnatural facial movements or inconsistencies in lighting and shadows. If the model only trains on frames that do not capture these discrepancies, it may fail to generalise to new instances of deepfakes. Al-Dhabi and Zhang [32] attempted to reduce computational costs by limiting their CNN-LSTM model to the first 100 sequential frames, which can result in missed deepfake instances occurring later in the video. Similar frame rate restrictions were observed in research by Yadav et al. [70], where models trained only on deepfake frames struggled to generalise to real-world videos containing a mix of authentic and manipulated content. As deepfake technology advances, these hybrid content videos may become more prevalent, and models that do not account for them risk incomplete or inaccurate detection. Other researchers, such as John and Sherif [45], restricted their models to 100 fps input to limit computational expenses, but doing so can limit the model's ability to detect temporal inconsistencies, reducing overall robustness.

A related issue is the reduction in frame rates during the pre-processing stage to save computational resources. Some studies employed techniques that sample every fifth frame, which, while efficient, can negatively impact the ability to capture critical temporal dynamics necessary for accurate detection [65]. A reduction in temporal resolution may cause the model to miss subtle manipulations or patterns needed for deepfake detection. This challenge becomes more apparent when dealing with deepfakes that feature fast transitions or nuanced alterations, suggesting that while frame sampling is useful for reducing computational load, it could compromise detection accuracy in more complex scenarios. Al-Dhabi and Zhang [32] reduced the frame rate input into the CNN-LSTM model, limiting the model to utilise the first 100 sequential frames. Similarly, researchers [25,57,71] employed models that only considered the first 150 sequential frames, therefore risking the possibility of not detecting deepfake instances in the remaining parts of the video. Research from Yadav et al. [70] suffered from a similar limitation, where the model input consisted solely of the deepfake frames of the video. This approach is not representative of all real-world scenarios, where videos can contain a mix of both authentic and manipulated content. As deepfake technology advances, more videos may feature a hybrid of real and deepfake elements to mislead the viewer. Furthermore, John and Sherif [45] restricted their model to accept a maximum frame rate input of 100 fps to limit computational expenses. Although this is still considered a high frame rate, placing limitations on the frame rate can limit the model's ability to capture temporal inconsistencies. This is reinforced by other findings by Saikia et al. [61], which only achieved an accuracy of 50% when using 20 fps as input. This continues in the findings of Stanciu and Ionescu [65] where the frame rates are reduced within the pre-processing stage by sampling every fifth frame. Although this reduces computational expenses, skipping intermediary frames can affect the model's ability to accurately capture temporal dynamics. A reduction in the temporal resolution of the frames from lower frequency sampling could cause fine details required for detecting patterns or anomalies to be missed. Therefore, the robustness of the model could be reduced when applied to deepfake videos with fast or subtle manipulations. An investigation into the optimal frame rate input for CNN-LSTM architecture can support future researchers by allowing their models to be less computationally expensive and finetune the pre-processing stage; however, as models operate on different hardware and frames are of varying quality, the optimal baseline may vary.

The ability to handle hybrid content is vital for the development of robust and reliable video deepfake detection systems. Addressing this limitation by analysing the entire video context in a computationally efficient manner, rather than a small sample of individual frames, could lead to more effective detection systems. Jalui et al. [43] introduce finding the correlation coefficient between subsequent frames as a solution to the computational expense of a high frame rate, where preceding frames are discarded from being fed into the network if they have a threshold difference greater than 0.005. This keeps input small and selects essential frames for the highest detection accuracy during the pre-processing stage. The results of Taviti et al. [68] show the CNN-LSTM model's performance plateaued at a 60 fps input. Increasing the frame rate to 80 fps and 100 fps did not yield any significant improvement in the accuracy measures, which remained consistent at 97% across the two datasets, with a variance of only 0.3%. This implies that beyond a threshold of 60 frames per second, additional temporal information does not contribute to improving the model's performance. The performance plateau observed suggests that the LSTM cells do not leverage long-range temporal information or the CNN layers in the CNN-LSTM architecture may not be fully effective in extracting the necessary spatial and temporal features required for accurate detection. Therefore, to ensure robust models are created, authors should test their models under a range of conditions to identify which architectural features or parameters hinder performance. However, when finetuning the CNN and LSTM bias and weight parameters using the PSO algorithm, an increased number of iterations did not lead to an increase in accuracy [31] An alternative solution is to integrate existing video codec information into the process instead of selecting frames. Information masks from the H.264 video codec are used to detect temporal inconsistencies using the preceding frames to map changes in the motion vector's spatial location [89].

Compression of video data further complicates deepfake detection, as CNN-LSTM models are sensitive to compression artefacts. Video compression can have a significant impact on the performance of CNN-LSTM-based video deepfake detection models. Amerini and Caldelli [34] found that their CNN-LSTM model demonstrated poor performance on strongly compressed data, indicating limitations in detecting deepfakes in highly compressed videos. However, the classification accuracy was better on lossless and level 23 and 40 compressed videos with LSTM employed than using a CNN alone, demonstrating that exploiting sequence correlations can help in distinguishing deepfake videos. Similarly, Chintha et al. [2] deduced that video compression adversely affects detection accuracy, particularly for the FaceSwap subset of the FaceForensics++ dataset. This model outperformed the Xception network on compressed MPEG and JPEG videos, but Xception was significantly better at detecting deepfakes in the FaceSwap subset, which was more heavily impacted by compression. Regarding audio channels, research [2] also found that increased audio compression made it harder to detect real from spoofed audio, emphasising the need for deepfake detection models to be robust to both video and audio compression. Also, the results of Masi et al. [14] demonstrated that a CNN-bi-LSTM model with a newly proposed loss function could achieve better frame-level detection performance for videos with medium compression rates, increasing the AUC score from 0.92 to 0.99. Additionally, Sooda et al.'s [64] ResNeXt-50-based model achieved 97.5% accuracy on compressed videos (c = 23) when training for 13 epochs, showing that intricate feature extraction and early stopping during training can overcome the challenges posed by video compression. Overall, the findings suggest that incorporating techniques to handle compression, such as pre-processing data with a fusion of frequency enhancements and colour domains alongside architecture-specific loss functions, can enhance the robustness of CNN-LSTM-based video deepfake detection models.

Facial occlusions remain a challenge for video deepfake detection models, requiring a larger frame rate to compensate for the loss of facial information [41]. The findings concluded that facial rotations limit the model's ability to classify without a full face showing, with testing accuracy only reaching 40.2%. This is reinforced by researchers [48] who observed that when the video subject looks away, detection can be compromised. To mitigate this challenge, the detection model should adapt its pre-processing and feature extraction techniques to detect and learn the key landmarks for facial rotations ranging from 45 to 90 degrees. This can be achieved by augmenting the training data with samples depicting various degrees of facial rotation. Furthermore, the CNN-LSTM architecture of the model should be designed to effectively track the dynamics of the face throughout the video sequence. The convolutional layers can be optimised to extract spatial features that are invariant to facial rotations, whilst the LSTM component learns the temporal patterns associated with real and deepfake manipulations in the presence of differing facial orientations to improve the overall robustness and real-life applications of video deepfake detection.

Masud et al. [38] deduced that the illumination within the frame can influence the detection accuracy. These findings highlight the requirement for the development of a new dataset that contains a range of lighting changes within the frames alongside a CNN-LSTM model that can learn under these conditions. A CNN model could be used to extract the spatial features from each frame, whilst the LSTM layers capture the temporal dependencies to allow the model to maintain a coherent understanding of the face and retain facial features that were present in the previous frames despite the lighting fluctuations. By aggregating the probabilities from multiple frames under different lighting conditions, the final computed classification can be more representative of dynamic conditions in which deepfake videos are often filmed.

One challenge that emerges in deepfake detection is the performance plateau due to parameter tuning, as discussed by Al-Adwan et al. [31]. In their research, they employed the PSO algorithm to finetune parameters iteratively, expecting continuous improvements. However, their findings showed that accuracy did not always increase as expected. This plateau suggests that certain optimisation algorithms might not be sufficient to boost performance after a certain threshold, highlighting the need for alternative methods or more sophisticated techniques to improve model tuning and avoid diminishing returns.

The computational expense of deepfake detection models is another prominent challenge, as highlighted by Al-Dulaimi and Kurnaz [33]. These models often require extensive resources to achieve high accuracy, resulting in trade-offs between computational efficiency and model performance. Chen, Li, and Ding [36] similarly reported that their model's non-parallelisable convolutional operations caused an increase in runtime, exacerbating the time–space complexity. These findings illustrate that the demand for computational power can impede the scalability and practical deployment of deepfake detection systems, particularly for real-time or large-scale applications.

Also, generalisation against unseen deepfake techniques remains a persistent issue for CNN-LSTM models. Masi et al. [14] concluded that the failure to detect real faces in their model could be accredited to characteristics such as a lack of facial hair in the training sets, which is a limitation when attribute manipulation is the deepfake creation method. This limitation in the training data can impact the model's performance on test sets where such features are present. When a small or undiversified dataset is used, the model learns patterns relating to specific facial characteristics, which can then cause real faces to be misclassified. Furthermore, Liang et al. [53] explored the use of a Facial Geometry Prior Module (FGPM) to extract facial landmarks for deepfake detection before passing the input to the CNN-LSTM architecture. However, the researchers found that this approach may be ineffective in detecting deepfake techniques that involve attribute manipulation, such as changes to hair, colouring, and skin retouching, as these do not significantly affect the facial structure. The evolving nature of deepfake generation techniques, such as the emergence of the Neural Textures subset within the FF++ dataset, further highlights the need for more adaptable detection models. Multiple studies [59,60] demonstrated that their CNN-LSTM models had the lowest performance metrics on neural texture deepfakes, therefore indicating that these models have a stronger performance on GAN-generated deepfakes.

Adversarial attacks pose a unique challenge to CNN-LSTM models due to their ability to target both spatial features, handled by CNNs, and temporal coherence, processed by LSTMs. Adversarial examples can be crafted to disrupt the temporal patterns between video frames or introduce pixel-level distortions that evade detection by the spatial filters of CNNs. Adversarial training of CNN-LSTM models using high-quality, imperceptible adversarial deepfakes will strengthen the robustness of the detection system as it learns to be vigorous against adversarial attacks. By intentionally adding Gaussian noise to deepfakes to misclassify them, the model continuously optimises as it learns to accurately reconstruct and classify the images. Table 8 summarises the challenges encountered when implementing the CNN-LSTM models in video deepfake detection in the selected publications.

Table 8. Challenges of video deepfake detection tools in the selected publications.

Author	Challenge	Description of the Challenge
Al-Adwan et al. [31]	Performance plateau due to parameter tuning	Fine-tuning the parameters through iteration using the PSO algorithm did not always lead to an increase in accuracy.
Al-Dhabi and Zhang [32]	Discarding frames	This model only uses the first 100 sequential frames, which could discard detecting a deepfake in the later portion of the clip.
Al-Dulaimi and Kurnaz [33]	Computationally expensive	This model has a trade-off between accuracy and computational efficiency
Amerini and Caldelli [34]	Low performance metrics on highly compressed videos	The CNN-LSTM model demonstrated poor performance on strongly compressed data, indicating limitations in detecting deepfakes in highly compressed videos.
Chan et al. [35]	Default assumption that the content is not deepfake	The model operates under the default assumption that the content is not deepfake, which could then potentially overlook key features of a deepfake.
Chen, Li, and Ding [36]	Computationally expensive and time consuming	The model has an increased runtime compared to other algorithms due to the non-parallelisable convolutional operations of ConvLSTM, resulting in a worst-case time–space complexity of $O(n^2)$.
Chinchalkar et al. [37]	Computationally expensive	This model requires a high number of frames to be fed into the network, which is computationally expensive.
Chintha et al. [2]	The impact of compression	The authors deduce that video compression adversely affects detection accuracy, in particular the FaceSwap subset of FF++.
Fuad, Amin, and Ahsan [38]	Poor illumination within the frame	Qualitative observations state that the illumination within the frame influences detection accuracy.
Gravina et al. [39]	Does not consider the audio stream within the deepfake	The model accounts for only one source of inconsistency in deepfake videos, not accounting for audio.
Guera and Delp [40]	Generalisability	The model is trained on a very small dataset, limiting its applicability to unseen data.
Hashmi et al. [41]	Occlusions	Facial occlusions impact accuracy and result in a larger frame rate being required.
Jaiswal et al. [42]	Hybrid model complexity	Combining the CNN-LSTM with GRU increases the training process time and complexity.
Jalui et al. [43]	Same training and testing dataset	Using limited test data of the same dataset limits generalisability.

Table 8. *Cont.*

Author	Challenge	Description of the Challenge
Jindal [44]	Computationally expensive	This model has a trade-off between accuracy and computational efficiency
John and Sherif [45]	Frame rate limitation	This model is limited to accepting a frame rate of 100 fps due to the computational expense.
Jolly et al. [46]	Increased data pre-processing	Due to the complexity of the feature extraction stage, the data are heavily pre-processed and then re-processed manually, which requires a lot of extra time and input.
Jungare et al. [47]	Diminishing returns	After 100 frames, the model's performance plateaus.
Kaur, Kumar, and Kumaraguru [48]	Facial rotations limiting detection	The video subject looking away causes detection to be compromised.
Koshy and Mahmood [49]	Loss of spatial information and lack of interpretability	Applying anisotropic diffusion can lead to the loss of fine spatial details whilst making it difficult to distinguish which specific features are affected by the diffusion.
Kukanov et al. [50]	Failure to detect online content	This model could not detect content collected from YouTube videos, which hosts a large portion of deepfake content.
Lai et al. [51]	Complexity	The involvement of spatial, frequency, time domain feature extraction, and LSTM is complex and computationally expensive.
Li, Chang, and Lyu [52]	Subjective threshold	The threshold set to what determines an open or closed eye impacts detection.
Liang et al. [53]	Reliance on Facial Geometry Prior Module (FGPM)	Relying on FPGM to extract facial landmarks may be ineffective in detecting deepfake techniques such as attribute manipulation, where changes in features such as hair, colours, and skin retouching do not affect the facial structure.
Malik et al. [54]	Pre-processing is sensitive to illumination and occlusions	The usage of the Viola–Jones algorithm is sensitive to occlusions and under/over light exposure.
Masi et al. [14]	Facial occlusions and poor illumination	Failure to detect real faces deduced from poor illumination and a lack of facial hair in training sets can impact results from test sets where this is present.
Masud et al. [38]	Poor illumination within the frame	The model struggled with low lighting, obscurity, and shadows.
Nawaz, Javed, and Irtaza [55]	ReLU activation function limitations	The ReLU activation function is liable to remove negative values during computation, which can potentially result in the loss of some subtle visual characteristics in the deeper layers.
Parayil et al. [56]	Generalisability	The model is trained on a very small dataset, limiting its applicability to unseen data.
Patel, Chandra, and Jain [57]	Discarding frames	This model only uses the first 150 sequential frames, which could discard the detection of a deepfake in the later portion of the clip.
Ritter et al. [58]	Overfitting	The model's testing accuracy is lower than its training and validation accuracy, therefore indicative of overfitting.
Saealal et al. [59]	Low detection accuracy on non-GAN generated deepfakes	Deepfakes generated using neural texture architecture had a low detection accuracy.
Saif et al. [60]	Low detection accuracy for attribute manipulation	The model struggled to detect deepfakes where the expressions were swapped.
Saikia et al. [61]	Frame rate limitation	A frame rate input of 20 fps produced a 0.5 accuracy measure.

Table 8. *Cont.*

Author	Challenge	Description of the Challenge
Saraswathi et al. [25]	Discarding frames	This model uses the first 150 sequential frames, which could discard detection of a deepfake in the later portion of the clip.
Shende, Paliwal, and Mahay [62]	Sequence processing	The biggest identified challenge was designing a classifier capable of recursively processing the sequence in a relevant manner.
Singh et al. [63]	Computationally expensive	There is a trade-off between the frame rate input into the model and the computational power used.
Sooda [64]	Training on small datasets	As the model was trained using a small portion of the DFDC dataset, its accuracy decreased when merged with the FF++ dataset.
Stanciu and Ionescu [65]	Pre-processing to handle sequential data	The model requires extra pre-processing to extract feature vectors from Xception for every fifth frame of the video.
Su et al. [66]	Limited to one dataset	The model trains and tests on the same FF++ dataset.
Suratkar and Kazi [67]	Generalisation	Training on specific manipulation techniques hinders the model's ability to generalise against unseen deepfake attacks, limiting its effectiveness when various manipulation methods are employed.
Taviti et al. [68]	Performance plateau despite frame increase	At a 60-frame input, performance plateaus where 80 and 100 frames retrieve the same accuracy.
Wubet [69]	Limited diversity	The model might not be trained on a diverse set of eyes, potentially leading to reduced performance or bias when faced with a broader dataset of eyes.
Yadav et al. [70]	Lack of genuine imagery	Excluding real imagery from the input, opting to only use the deepfake limits the model's ability to detect deepfakes in clips in which real and fake content coexist.
Yesugade et al. [71]	Discarding frames	This model uses the first 150 sequential frames, which could discard detection of a deepfake in the later portion of the clip.

To overcome these challenges, we propose several solutions, which are detailed and discussed in Table 9.

Table 9. Summary of the challenges and their suggested solutions.

Challenge	Suggested Solution
Performance trade-off	To mitigate computational requirements, the LSTM architecture can be leveraged to store essential features and temporal patterns, rather than processing and saving entire video frames on a physical server. This approach significantly reduces the memory overhead, as only relevant and condensed information is retained, instead of storing full-resolution video data for every frame. By focusing on capturing temporal dynamics, LSTM layers can efficiently encode the relationships between video frames. This allows the model to understand changes over time without the need for extensive storage or processing resources. For instance, instead of holding redundant pixel data, the model retains the most meaningful information.
Frame Rate Limitation	Analysing the entire video context in a computationally efficient manner, rather than sampling individual frames, offers a more holistic approach to deepfake detection. By considering the full video sequence, the model can better capture temporal inconsistencies and subtle manipulations that may be missed when only isolated frames are examined. This approach ensures that the detection system leverages both spatial and temporal information, providing a more robust understanding of how visual elements evolve over time, which is critical for detecting sophisticated deepfakes.

Table 9. *Cont.*

Challenge	Suggested Solution
Restricted Frame Rate Input	Utilising motion vectors and information masks from the H.264 video codec significantly enhances the efficiency of detecting temporal inconsistencies in video content. Instead of analysing every individual frame, this approach leverages the motion vectors, which represent the direction and magnitude of pixel movement between frames. By focusing on these vectors, the system can identify areas of significant change without the computational overhead associated with full RGB analysis. Information masks further refine this process by highlighting critical regions within the video, enabling targeted scrutiny of potentially manipulated areas. This dual strategy not only reduces computational costs but also preserves detection effectiveness, allowing for quicker analysis while maintaining accuracy, especially in scenarios involving high frame rates or large datasets.
Video Compression	Pre-processing techniques that combine frequency enhancements with colour domain transformations significantly enhance the model's resilience to compression artefacts. By fusing these enhancements, the model can better capture essential features that may be obscured in highly compressed videos. Additionally, using architecture-specific loss functions optimises training by emphasising relevant characteristics for deepfake detection, allowing the model to focus on crucial elements even when subtle details are compromised. This holistic approach ensures that the model maintains accuracy and effectiveness in identifying deepfakes, even in challenging scenarios where traditional methods may falter due to the loss of critical information.
Facial Occlusions	Adapting pre-processing and feature extraction techniques to focus on key landmarks associated with facial rotations enhances the model's ability to recognise faces under various orientations. This involves training the model to identify critical facial features, such as the eyes, nose, and mouth, despite changes in position. Additionally, designing the CNN-LSTM architecture to effectively track facial dynamics over time allows for the capturing of essential temporal information, ensuring that the model maintains a coherent understanding of the face throughout the video sequence. This dual approach significantly boosts the model's robustness against facial occlusions, improving its accuracy in detecting deepfakes even when facial information is partially obscured.
Lighting Conditions	Training on datasets that encompass a diverse array of facial characteristics including variations in facial hair, skin tones, and distinctive features helps to minimise bias in the model. This diversity enables the model to learn a broader range of patterns and characteristics, thereby enhancing its generalisability across different populations. As a result, the model becomes more adept at recognising authentic faces, regardless of variations in appearance. By being exposed to various features, the model improves its effectiveness in detecting unseen deepfake manipulation techniques. This approach not only strengthens the model's performance on diverse inputs but also enhances its reliability in real-world applications, where facial characteristics can vary significantly.
Evolving Deepfake Techniques	Continuously testing and adapting CNN-LSTM models to keep pace with emerging deepfake techniques is crucial for maintaining their relevance and effectiveness. As new manipulation methods, such as the neural textures subset, evolve, models must be updated to recognise and respond to these advanced threats. This ongoing process involves regular evaluation of the model's performance on new datasets and incorporation of recent findings in deepfake technology. By refining the model through iterative training and testing, it becomes more resilient to previously unseen attacks, ensuring that it can accurately detect deepfakes across a wide range of scenarios. This proactive approach not only bolsters the model's defence mechanisms but also solidifies its role as a reliable tool in combating deepfake proliferation.

6. Open Issues and Future Research Directions

Video deepfake detection has emerged as a critical field within the broader domain of digital technologies, driven by the growing dominance of manipulated video content and its potential misuse. The integration of CNN and LSTM has become essential to build detection tools for identifying deepfakes due to their ability to capture spatial and temporal patterns. However, this area faces significant challenges that need to be addressed, and numerous future research directions must be explored to enhance the effectiveness of detection methods and keep pace with advancing techniques. These challenges and future research directions include the following:

6.1. Real-Time Detection

A key issue with video deepfake detection systems using CNN and LSTM is their computational expense, which hinders real-time analysis. The complexity of these models, coupled with the need to process vast amounts of video data, results in high latency, making them impractical for time-sensitive applications like live video moderation or surveillance [88]. The computational burden is further amplified when dealing with high-resolution or long-duration videos, where current models struggle to balance accuracy and efficiency. This presents a significant bottleneck, as delayed detection can reduce the system's effectiveness in mitigating the spread of harmful deepfake content [57]. Future research should focus on developing lightweight models or optimising existing ones for faster, more efficient processing without compromising detection accuracy. Techniques such as model compression, pruning, and quantisation could help streamline the computational load. Leveraging hardware acceleration, such as GPUs or TPUs, and edge computing to distribute processing closer to the data source could further enhance real-time capabilities. Additionally, exploring hybrid models that combine CNN and LSTM with faster architectures like transformers may offer a solution for achieving both speed and accuracy. These advancements are essential for making real-time deepfake detection feasible in practical, large-scale applications.

6.2. Generalisation

A significant challenge in video deepfake detection using CNN and LSTM is the generalisation of models across diverse datasets. Typically, these models are trained on specific datasets, which can limit their ability to perform well when exposed to new, unseen datasets with different content, quality, or generation methods [84]. This lack of generalisation reduces the practicality of detection systems in real-world scenarios where deepfakes vary widely in their creation techniques and video characteristics [40,56]. One major factor contributing to this challenge is dataset diversity. Most deepfake detection models are trained and validated on a small number of standardised datasets, such as DFDC or FaceForensics++, which represent only a fraction of the possible deepfake types. When tested on entirely different datasets, especially those featuring less common or more sophisticated techniques like neural textures, the performance of these models tends to degrade significantly. Over-reliance on dataset-specific artefacts, such as facial landmarks or lighting patterns, further reduces the model's ability to generalise. To address this, more research is needed to focus on developing large, diverse, and standardised benchmark datasets that can enhance model generalisation. Additionally, exploring transfer learning and domain adaptation techniques may improve the robustness of models, allowing them to perform effectively across different types of deepfakes and video domains.

6.3. Lack of Standardised Performance Metrics

Another significant issue in video deepfake detection is the lack of standardised performance metrics, which makes it difficult to benchmark and compare different models [53]. Currently, various metrics such as accuracy, precision, recall, and AUC are used to evaluate detection systems, but each metric emphasises different aspects of model performance [65]. For instance, while accuracy measures the overall correctness of a model, precision and recall focus on its ability to handle false positives and false negatives, respectively [44]. This variation in evaluation methods leads to inconsistent and non-comparable results across studies, making it challenging to determine which models are more effective. Researchers should focus on establishing standardised performance metrics that reflect the unique challenges of video deepfake detection. A unified framework for evaluation could incorporate multiple aspects of performance, such as robustness to adversarial attacks, sensitivity to low-quality data, and real-time processing capabilities. Additionally, researchers could explore composite metrics that combine the strengths of multiple evaluation criteria to create a more holistic view of a model's effectiveness in detecting deepfakes across various scenarios.

6.4. Multimodal Deepfake Detection

A key limitation in current video deepfake detection approaches using CNN and LSTM is their heavy reliance on visual cues. These models primarily analyse facial expressions, movements, or pixel-level details in video frames, but deepfakes often manipulate multiple modalities, including audio [37]. As a result, purely visual-based detection systems may overlook critical tampering in the audio stream, such as altered voices or speech patterns that do not match the visual content. This single-modality focus reduces the effectiveness of detection systems in identifying more sophisticated deepfakes that manipulate both video and audio [43]. More research is needed to explore multimodal detection techniques that analyse both audio and video components of deepfakes. Additionally, cross-modal consistency checks could be developed to compare the synchrony and coherence between the audio and visual elements, further enhancing the reliability of detection systems.

6.5. Detecting Low-Quality and Compressed Deepfakes

Detecting low-quality and compressed deepfakes remains a significant challenge for many existing video deepfake detection models. These models often struggle when videos are noisy, compressed, or low-resolution, which is common when deepfakes are circulated on social media platforms [34]. Compression artefacts, pixelation, and noise can obscure the visual cues that models like CNNs rely on, reducing their effectiveness in identifying subtle signs of tampering [2]. As a result, video deepfake detection in these real-world scenarios becomes significantly more difficult, allowing manipulated content to evade detection and spread unchecked [87]. To overcome this challenge, researchers should focus on developing techniques that are specifically designed to handle compressed and low-resolution video data. Multi-scale learning methods could be employed, where models analyse video data at different resolutions to detect patterns that may be invisible at a single scale.

6.6. Ethical and Privacy Concerns

A critical issue surrounding deepfake detection models is the ethical and privacy concerns they raise, particularly when analysing large volumes of video data [44]. These systems often require access to personal or sensitive content, raising the risk of privacy violations, especially if the data are collected or processed without consent [7]. Additionally, there is the potential for these detection models to be misused in non-consensual settings, such as mass surveillance or monitoring, further complicating their ethical deployment. The balance between effective deepfake detection and the protection of individual privacy is a challenging ethical dilemma that needs careful consideration [89]. Future research should focus on developing privacy-preserving video deepfake detection techniques that can analyse video content while adhering to privacy laws and regulations. This could involve techniques like federated learning, where models are trained on decentralised data without the need for raw data collection, or differential privacy, which ensures that individual data cannot be easily extracted from the model's output.

6.7. Adversarial Training and Attacks

Adversarial training and attacks present significant challenges for deepfake detection, particularly as adversarial examples can easily deceive Deep Neural Networks (DNNs). Adversarial attacks cause concern due to their transferability across models, allowing inconsistencies designed for one deepfake detector to potentially bypass others, posing a critical threat, as attackers could exploit open-source models to create adversarial input that can evade. To address this, advanced defences like denoising-based approaches, such as D-VAEGAN [66], have shown promise by projecting adversarial examples into a low-dimensional latent space and reconstructing clean images without retraining the detection model. However, deepfake detectors must also contend with the variability in data preprocessing and generation techniques. Future research needs to focus on developing more adaptable defences that can effectively mitigate the cross-model transferability of adversarial attacks while maintaining robust real-time performance.

7. Conclusions

The integration of CNN with LSTM networks has emerged as a promising strategy for enhancing video deepfake detection, demonstrating remarkable accuracy levels that can approach 100%. This hybrid approach capitalises on the unique strengths of CNNs in extracting spatial features from individual video frames, allowing for detailed analysis of visual elements such as facial textures, lighting inconsistencies, and resolution anomalies. At the same time, LSTMs shine at analysing temporal patterns, providing insights into how these visual features evolve and enabling the detection of irregularities that may indicate manipulation. By effectively combining spatial and temporal analyses, this hybrid model significantly improves the evaluation of videos, making it increasingly difficult for deepfakes to evade detection. While the potential of CNN-LSTM models is evident, the existing research lacks systematic evaluations that comprehensively assess their effectiveness, optimal configurations, and best practices across various contexts. This paper presents a comprehensive systematic review that assesses state-of-the-art video deepfake detection tools, focusing on the key factors influencing detection performance in this rapidly evolving field. This review emphasises the critical importance of understanding both the capabilities and limitations of current detection systems as deepfake technology advances. Following a rigorous search strategy, a total of 45 articles were selected from 674 publications across multiple reputable databases. The selected articles were examined to evaluate recent advancements in video deepfake detection methodologies using CNN-LSTM. This paper identifies the most common feature extraction techniques employed in these tools and examines the datasets frequently used for training and testing. This paper also evaluates the performance of these models on different datasets to highlight their strengths and limitations. Additionally, this paper compares CNN-LSTM models with alternative approaches that do not employ LSTM to assess their effectiveness in deepfake detection. This paper also explores the challenges associated with implementing these tools and proposes potential solutions to overcome these barriers. Lastly, this paper presents open issues and research directions that are critical including the need for real-time detection methods, strategies for improving model generalisation across diverse datasets, the establishment of standardised performance metrics, and techniques for effectively detecting low-quality and compressed deepfakes. Addressing these issues is essential to enhance the robustness and applicability of deepfake detection systems in real-world scenarios.

Author Contributions: Conceptualisation, H.F.A. and S.T.; methodology, H.F.A.; software, S.T.; validation, S.T., H.F.A. and H.S.L.; formal analysis, S.T and H.F.A.; investigation, S.T.; resources, S.T. and H.S.L.; data curation, S.T.; writing—original draft preparation, S.T and H.F.A.; writing—review and editing, H.S.L. and H.F.A.; visualisation, S.T.; supervision, H.F.A. and H.S.L.; project administration, H.F.A. and H.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Agre, P.; Rotenberg, M. *Technology and Privacy: The New Landscape*; MIT Press: Cambridge, MA, USA, 1998.
2. Chintha, A.; Thai, B.; Sohrawardi, S.J.; Bhatt, K.; Hickerson, A.; Wright, M.; Ptucha, R. Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 1024–1037. [CrossRef]
3. Lyu, S. DeepFake Detection: Current Challenges and Next Steps. *arXiv* **2020**, arXiv:2003.09234.

4. Santha, A. Deepfakes Generation Using LSTM Based Generative Adversarial Networks Networks. 2020. Available online: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (accessed on 20 November 2023).
5. Rocca, J. Understanding Variational Autoencoders (VAEs). Medium. 2019. Available online: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73 (accessed on 20 November 2023).
6. Liu, Y.; Li, Q.; Deng, Q.; Sun, Z.; Yang, M.H. GAN-Based Facial Attribute Manipulation. *arXiv* **2022**, arXiv:2210.12683. [CrossRef] [PubMed]
7. Zobaed, S.; Rabby, F.; Hossain, I.; Hossain, E.; Hasan, S.; Karim, A.; Md. Hasib, K. DeepFakes: Detecting Forged and Synthetic Media Content Using Machine Learning. In *Artificial Intelligence in Cyber Security: Impact and Implications: Security Challenges, Technical and Ethical Issues, Forensic Investigative Challenges*; Montasari, R., Jahankhani, H., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 177–201. [CrossRef]
8. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 2 July 2011; Volume 27.
9. OValery. Swap-Face. 2017. Available online: https://github.com/OValery16/swap-face (accessed on 4 October 2024).
10. Xu, F.J.; Wang, R.; Huang, Y.; Guo, Q.; Ma, L.; Liu, Y. Countering Malicious DeepFakes: Survey, Battleground, and Horizon. *arXiv* **2021**, arXiv:2103.00218.
11. Verdoliva, L. Media Forensics and DeepFakes: An Overview. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 910–932. [CrossRef]
12. Agarwal, S.; Farid, H.; Fried, O.; Agrawala, M. Detecting Deep-Fake Videos from Phoneme-Viseme Mismatches. 2020. Available online: www.instagram.com/bill_posters_uk (accessed on 3 October 2024).
13. Chugh, K.; Gupta, P.; Dhall, A.; Subramanian, R. *Not Made for Each Other-Audio-Visual Dissonance-Based Deepfake Detection and Localization*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 439–447.
14. Masi, I.; Killekar, A.; Mascarenhas, R.M.; Gurudatt, S.P.; AbdAlmageed, W. Two-branch Recurrent Network for Isolating Deepfakes in Videos. *arXiv* **2020**, arXiv:2008.03412.
15. de Lima, O.; Franklin, S.; Basu, S.; Karwoski, B.; George, A. Deepfake Detection using Spatiotemporal Convolutional Networks. *arXiv* **2020**, arXiv:2006.14749.
16. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. *Convolutional Neural Networks: An Overview and Application in Radiology*; Insights into Imaging; Springer: Berlin/Heidelberg, Germany, 2018; Volume 9, pp. 611–629.
17. Nunnari, G.; Calvari, S. Exploring Convolutional Neural Networks for the Thermal Image Classification of Volcanic Activity. *Geomatics* **2024**, *4*, 124–137. [CrossRef]
18. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef]
19. Keita, Z. An Introduction to Convolutional Neural Networks (CNNs). 2023. Available online: https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns (accessed on 25 April 2024).
20. Wu, H.; Gu, X. Towards dropout training for convolutional neural networks. *Neural Netw.* **2015**, *71*, 1–10. [CrossRef]
21. More, Y.; Dumbre, K.; Shiragapur, B. Horizontal Max Pooling a Novel Approach for Noise Reduction in Max Pooling for Better Feature Detect. In Proceedings of the 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 1–3 March 2023; pp. 1–5.
22. Mishra, M. Convolutional Neural Networks, Explained. 2020. Available online: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 (accessed on 25 April 2024).
23. Jain, A.; Korshunov, P.; Marcel, S. Improving Generalization of Deepfake Detection by Training for Attribution. In Proceedings of the 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 6–8 October 2021; pp. 1–6.
24. Graves, A. Long Short-Term Memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Graves, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45. [CrossRef]
25. Saraswathi, R.V.; Gadwalkar, M.; Midhun, S.S.; Goud, G.N.; Vidavaluri, A. Detection of Synthesized Videos using CNN. In Proceedings of the International Conference on Augmented Intelligence and Sustainable Systems, ICAISS 2022, Trichy, India, 24–26 November 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022.
26. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
27. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. Available online: https://www.sciencedirect.com/science/article/pii/S0893608005001206 (accessed on 4 October 2024). [CrossRef] [PubMed]
28. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. Available online: https://www.bmj.com/content/372/bmj.n71 (accessed on 3 October 2024). [CrossRef] [PubMed]
29. Nightingale, A. A guide to systematic literature reviews. *Surgery* **2009**, *27*, 381–384. Available online: https://www.sciencedirect.com/science/article/pii/S0263931909001707 (accessed on 4 October 2024). [CrossRef]
30. Easterbrook, P.J.; Gopalan, R.; Berlin, J.A.; Matthews, D.R. Publication bias in clinical research. *Lancet* **1991**, *337*, 867–872. [CrossRef]

31. Al-Adwan, A.; Alazzam, H.; Al-Anbaki, N.; Alduweib, E. Detection of Deepfake Media Using a Hybrid CNN–RNN Model and Particle Swarm Optimization (PSO) Algorithm. *Computers* **2024**, *13*, 99. [CrossRef]
32. Al-Dhabi, Y.; Zhang, S. Deepfake Video Detection by Combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). In Proceedings of the 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering, CSAIEE 2021, Virtual, 20–22 August 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021; pp. 236–241.
33. Al-Dulaimi, O.A.H.H.; Kurnaz, S. A Hybrid CNN-LSTM Approach for Precision Deepfake Image Detection Based on Transfer Learning. *Electronics* **2024**, *13*, 1662. [CrossRef]
34. Amerini, I.; Caldelli, R. Exploiting Prediction Error Inconsistencies through LSTM-based Classifiers to Detect Deepfake Videos. In Proceedings of the IH and MMSec 2020, 2020 ACM Workshop on Information Hiding and Multimedia Security, New York, NY, USA, 22–24 June 2020; Association for Computing Machinery, Inc.: New York, NY, USA, 2020; pp. 97–102.
35. Chan, K.; Chun, C.; Kumar, V.; Delaney, S.; Gochoo, M. Combating Deepfakes: Multi-LSTM and Blockchain as Proof of Authenticity for Digital Media. In Proceedings of the 2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G), Geneva, Switzerland, 21–25 September 2020.
36. Chen, B.; Li, T.; Ding, W. Detecting deepfake videos based on spatiotemporal attention and convolutional LSTM. *Inf. Sci.* **2022**, *601*, 58–70. [CrossRef]
37. Chinchalkar, R.; Sinha, R.; Kumar, M.; Chauhan, N.; Deokar, S.; Gonge, S. Detecting Deepfakes Using CNN and LSTM. In Proceedings of the 2023 2nd International Conference on Informatics, ICI 2023, Noida, India, 23–25 November 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023.
38. Hasan Fuad, M.T.; Bin Amin, F.; Masudul Ahsan, S.M. Deepfake Detection from Face-swapped Videos Using Transfer Learning Approach. In Proceedings of the 2023 26th International Conference on Computer and Information Technology, ICCIT 2023, Cox's Bazar, Bangladesh, 13–15 December 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023.
39. Gravina, M.; Galli, A.; De Micco, G.; Marrone, S.; Fiameni, G.; Sansone, C. FEAD-D: Facial Expression Analysis in Deepfake Detection. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer Science and Business Media Deutschland GmbH: Berlin, Germany, 2023; pp. 283–294.
40. Güera, D.; Delp, E.J. Deepfake Video Detection Using Recurrent Neural Networks. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–6.
41. Hashmi, M.F.; Ashish, B.K.K.; Keskar, A.G.; Bokde, N.D.; Yoon, J.H.; Geem, Z.W. An Exploratory Analysis on Visual Counterfeits Using Conv-LSTM Hybrid Architecture. *IEEE Access* **2020**, *8*, 101293–101308. [CrossRef]
42. Jaiswal, G. Hybrid Recurrent Deep Learning Model for DeepFake Video Detection. In Proceedings of the 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2021, Dehradun, India, 11–13 November 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021.
43. Jalui, K.; Jagtap, A.; Sharma, S.; Mary, G.; Fernandes, R.; Kolhekar, M. Synthetic Content Detection in Deepfake Video Using Deep Learning. In Proceedings of the 2022 IEEE 3rd Global Conference for Advancement in Technology, GCAT 2022, Bangalore, India, 7–9 October 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022.
44. Jindal, A. Deepfake Video Forgery Detection. 2023. Available online: https://link.springer.com/chapter/10.1007/978-3-031-12413-6_53 (accessed on 3 October 2024).
45. John, J.; Sherif, B.V. Multi-model DeepFake Detection Using Deep and Temporal Features. In *Lecture Notes in Networks and Systems*; Springer Science and Business Media Deutschland GmbH: Berlin, Germany, 2022; pp. 672–684.
46. Jolly, V.; Telrandhe, M.; Kasat, A.; Shitole, A.; Gawande, K. CNN based Deep Learning model for Deepfake Detection. In Proceedings of the 2022 2nd Asian Conference on Innovation in Technology, ASIANCON 2022, Ravet, India, 26–28 August 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022.
47. Jungare, M.; Ganganmale, P.; Khandagale, R.; Dhamane, S.; Susar, A. DeepFake Detection Model Using LSTM-CNN with Image and Temporal Video Analysis. *Int. J. All Res. Educ. Sci. Methods* **2024**, *12*, 94–99. Available online: www.ijaresm.com (accessed on 3 October 2024).
48. Kaur, S.; Kumar, P.; Kumaraguru, P. Deepfakes: Temporal sequential analysis to detect face-swapped video clips using convolutional long short-term memory. *J. Electron. Imaging* **2020**, *29*, 033013. [CrossRef]
49. Koshy, R.; Mahmood, A. Enhanced deep learning architectures for face liveness detection for static and video sequences. *Entropy* **2020**, *22*, 1186. [CrossRef] [PubMed]
50. Kukanov, I.; Karttunen, J.; Sillanpää, H.; Hautamäki, V. Cost Sensitive Optimization of Deepfake Detector. In Proceedings of the APSIPA Annual Summit and Conference, Auckland, New Zealand, 7–10 December 2020; Available online: https://ieeexplore.ieee.org/abstract/document/9306476/ (accessed on 4 October 2024).
51. Lai, Z.; Wang, Y.; Feng, R.; Hu, X.; Xu, H. Multi-Feature Fusion Based Deepfake Face Forgery Video Detection. *Systems* **2022**, *10*, 31. [CrossRef]
52. Li, Y.; Chang, M.C.; Lyu, S. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 11–13 December 2018; Available online: https://ieeexplore.ieee.org/abstract/document/8630787 (accessed on 4 October 2024).

53. Liang, P.; Liu, G.; Xiong, Z.; Fan, H.; Zhu, H.; Zhang, X. A facial geometry based detection model for face manipulation using CNN-LSTM architecture. *Inf. Sci.* **2023**, *633*, 370–383. [CrossRef]
54. Malik, M.H.; Ghous, H.; Qadri, S.; Ali Nawaz, S.; Anwar, A.; Author, C. Frequency-based Deep-Fake Video Detection using Deep Learning Methods. *J. Comput. Biomed. Inform.* **2023**, *4*, 41–48. [CrossRef]
55. Nawaz, M.; Javed, A.; Irtaza, A. Convolutional long short-term memory-based approach for deepfakes detection from videos. *Multimed. Tools Appl.* **2023**, *83*, 16977–17000. [CrossRef]
56. Parayil, A.M.; Masood, A.; Ajas, M.; Tharun, R.; Usha, K. Deepfake Detection Using Xception and LSTM. *Int. Res. J. Mod. Eng. Technol. Sci.* **2023**, *5*, 9191–9196. [CrossRef]
57. Patel, S.; Chandra, S.K.; Jain, A. DeepFake Videos Detection and Classification Using Resnext and LSTM Neural Network. In Proceedings of the 2023 3rd International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2023, Bangalore, India, 29–31 December 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023.
58. Ritter, P.; Lucian, D.; Anderies; Chowanda, A. Comparative Analysis and Evaluation of CNN Models for Deepfake Detection. In Proceedings of the 2023 4th International Conference on Artificial Intelligence and Data Sciences: Discovering Technological Advancement in Artificial Intelligence and Data Science, AiDAS 2023, Ipoh, Malaysia, 6–7 September 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023; pp. 250–255.
59. Saealal, M.S.; Ibrahim, M.Z.; Mulvaney, D.J.; Shapiai, M.I.; Fadilah, N. Using cascade CNN-LSTM-FCNs to identify AIaltered video based on eye state sequence. *PLoS ONE* **2022**, *17*, e0278989. [CrossRef]
60. Saif, S.; Tehseen, S.; Ali, S.S.; Kausar, S.; Jameel, A. Generalized Deepfake Video Detection Through Time-Distribution and Metric Learning. *IT Prof.* **2022**, *24*, 38–44. [CrossRef]
61. Saikia, P.; Dholaria, D.; Yadav, P.; Patel, V.; Roy, M. A Hybrid CNN-LSTM model for Video Deepfake Detection by Leveraging Optical Flow Features. In Proceedings of the International Joint Conference on Neural Networks, Padua, Italy, 18–23 July 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022.
62. Shende, A.; Paliwal, S.; Mahay, T.K. Using deep learning to detect deepfake videos. *Turk. J. Comput. Math. Educ.* **2021**, *12*, 5012–5017.
63. Singh, A.; Saimbhi, A.S.; Singh, N.; Mittal, M. DeepFake Video Detection: A Time-Distributed Approach. *SN Comput. Sci.* **2020**, *1*, 212. [CrossRef]
64. Sooda, K. DeepFake Detection Through Key Video Frame Extraction using GAN. In Proceedings of the International Conference on Automation, Computing and Renewable Systems, ICACRS 2022, Pudukkottai, India, 13–15 December 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022; pp. 859–863.
65. Stanciu, D.C.; Ionescu, B. Deepfake Video Detection with Facial Features and Long-Short Term Memory Deep Networks. In Proceedings of the ISSCS 2021, International Symposium on Signals, Circuits and Systems, Iasi, Romania, 15–16 July 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021.
66. Montserrat, D.M.; Hao, H.; Yarlagadda, S.K.; Baireddy, S.; Shao, R.; Horváth, J.; Bartusiak, E.; Yang, J.; Guera, D.; Zhu, F.; et al. Deepfakes detection with automatic face weighting. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 2851–2859.
67. Suratkar, S.; Kazi, F. Deep Fake Video Detection Using Transfer Learning Approach. *Arab. J. Sci. Eng.* **2022**, *48*, 9727–9737. [CrossRef] [PubMed]
68. Taviti, R.; Taviti, S.; Reddy, P.A.; Sankar, N.R.; Veneela, T.; Goud, P.B. Detecting Deepfakes with ResNext and LSTM: An Enhanced Feature Extraction and Classification Framework. In Proceedings of the 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication, IConSCEPT 2023, Karaikal, India, 25–26 May 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023.
69. Wubet, W.M. The Deepfake Challenges and Deepfake Video Detection. *Int. J. Innov. Technol. Explor. Eng.* **2020**, *9*, 789–796. Available online: https://www.ijitee.org/portfolio-item/E2779039520/ (accessed on 4 October 2024). [CrossRef]
70. Yadav, P.; Jaswal, I.; Maravi, J.; Choudhary, V.; Khanna, G. DeepFake Detection Using InceptionResNetV2 and LSTM. In Proceedings of the International Conference on Emerging Technologies: AI, IoT, and CPS for Science Technology Applications, Chandigarh, India, 6–7 September 2021.
71. Yesugade, T.; Kokate, S.; Patil, S.; Varma, R.; Pawar, S. Deepfake detection using LSTM-based neural network. In *Object Detection by Stereo Vision Images*; Wiley: Hoboken, NJ, USA, 2022; pp. 111–120.
72. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2016**, arXiv:1610.02357.
73. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
74. Bansal, A.; Ma, S.; Ramanan, D.; Sheikh, Y. Recycle-gan: Unsupervised video retargeting. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 119–135.
75. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
76. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multi-Task Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]

77. Horn, B.K.P.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. Available online: https://www.sciencedirect.com/science/article/pii/0004370281900242 (accessed on 4 October 2024). [CrossRef]
78. Dolhansky, B.; Howes, R.; Pflaum, B.; Baram, N.; Ferrer, C.C. The Deepfake Detection Challenge (DFDC) Preview Dataset. *arXiv* **2019**, arXiv:1910.08854.
79. Rössler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; Nießner, M. FaceForensics++: Learning to Detect Manipulated Facial Images. *arXiv* **2019**, arXiv:1901.08971.
80. Thies, J.; Zollhöfer, M.; Nießner, M. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph. (TOG)* **2019**, *38*, 1–12. [CrossRef]
81. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S. Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. *arXiv* **2019**, arXiv:1909.12962.
82. Korshunov, P.; Marcel, S. DeepFakes: A New Threat to Face Recognition? Assessment and Detection. *arXiv* **2018**, arXiv:1812.08685.
83. Jiang, L.; Li, R.; Wu, W.; Qian, C.; Loy, C.C. DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection. *arXiv* **2020**, arXiv:2001.03024.
84. Guilloux, L. FakeApp. 2019. Available online: https://www.malavida.com/en/soft/fakeapp/ (accessed on 29 January 2024).
85. Brownlee, J. What Is the Difference Between a Batch and an Epoch in a Neural Network. *Mach. Learn. Mastery* **2018**, *20*, 1–5.
86. Su, Y.; Xia, H.; Liang, Q.; Nie, W. Exposing DeepFake Videos Using Attention Based Convolutional LSTM Network. *Neural Process Lett.* **2021**, *53*, 4159–4175. [CrossRef]
87. Xia, Z.; Qiao, T.; Xu, M.; Wu, X.; Han, L.; Chen, Y. Deepfake video detection based on MesoNet with preprocessing module. *Symmetry* **2022**, *14*, 939. [CrossRef]
88. Selim, S. A Prize Winning Solution for DFDC Challenge. 2020. Available online: https://github.com/selimsef/dfdc_deepfake_challenge (accessed on 4 October 2024).
89. Grönquist, P.; Ren, Y.; He, Q.; Verardo, A.; Süsstrunk, S. Efficient Temporally-Aware DeepFake Detection Using H. 264 Motion Vectors. *arXiv* **2023**, arXiv:231110788. [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.