TIJFS

FRONTIER SCIENCE ASSOCIATES

Article

Toward Efficient Fake Frame Detection in Video Using Deep Learning

Iftikhar Alam [1,*], Malik Ahsan Kamran [1] and Ehaab Ullah [1]

[1] Department of Computer Science, City University of Science and Information Technology, Peshawar 25000, Pakistan

* Correspondence: iftikharalam@cusit@edu.pk

Received: 6 February 2026; Accepted: 10 August 2026; Published: 25 August 2026

How to cite: Alam, I.; Ahsan Kamran, M.; Ullah, E. Toward Efficient Fake Frame Detection in Video Using Deep Learning. *TIJFS* **2026**, *8* (1), 4. DOI: 10.37978/tijfs.v08i01.004

Academic Editors: Mian Sahib Zar and Muhammad Shoaib Akhtar

Abstract: **Background:** Deepfake technology is a major social concern due to the rapid development of Artificial Intelligence (AI), especially in machine learning (ML) and deep learning (DL). Deepfakes are artificially modified videos and images that effectively change a person's facial features or expressions to misrepresent reality. These videos and images, often unnoticed by casual observers, present significant ethical, political, and social implications, as they can spread misinformation, damage reputations, and influence public perception. This study is an attempt to detect artificially modified videos by analyzing each frame using DL. We use ResNet-50 architecture, a well-known Convolutional Neural Network (CNN) model, to identify tampered videos. **Methods:** The system is trained using the Celebrity Deep Fake dataset, which includes numerous original and fake video samples. The model assesses whether each frame is original or tampered with after the videos have been split into frames. The system is tested and evaluated using standard metrics, including accuracy, precision, recall, and F1-score. **Results:** The model achieved 82.33% accuracy, 76.70% precision, 89.15% recall, and an F1-score of 82.46%. These results indicate that deepfake videos were correctly detected and that the model was efficient at identifying most real deepfake instances. In addition, the F1-score of 82.46% is further evidence of the model's stability, as it ensures both high accuracy and coherence across numerous cases. **Conclusions:** The findings indicate that the proposed model works effectively compared to existing deepfake methods. In the future, we intend to use a richer dataset with more resources, which may further enhance the accuracy of the model.

Keywords: deepfake videos; deep learning model; ResNet-50; video frames detection; video forensics

1. Introduction

In today's rapidly evolving digital content ecosystem, the internet is flooded with millions of videos and images each day [1], some of which are tampered with using deepfake techniques. Deepfakes are fake/altered videos that are created with the help of AI tools and deep learning (DL) techniques [2,3]. These technologies can realistically recreate a person's face and expression so that it becomes hard to judge whether the content is real or fake. This represents a major problem in every field, including politics, journalism, and social media. [4]. For example, a fake video can be generated using AI tools to make it appear as if a politician said something that they never said, or to show a celebrity performing something that they never did. The video will quickly become viral due to the confusion, spreading false information and damaging public opinion. One of the major concerns regarding fake videos is that they destroy the credibility of social media and public trust [5].

The loss of trust in social media platforms can have a serious effect on society, including political instability, online harassment, financial fraud, and blackmailing. Deepfakes are becoming more advanced day by day, and therefore, detection becomes more difficult. There is a strong need for an automatic system that can detect these fake videos quickly and accurately. This study focuses on such a system, which builds on DL techniques. The purpose is to develop a model for a tool that analyzes videos, helping individuals, along with organizations and media outlets, identify fake content before it spreads. There is a need for effective techniques to identify and deal with these manipulated videos.

With the massive growth of knowledge regarding deepfakes, researchers have created numerous algorithms and tools that are in development [6,7]. These tools are useful for identifying deepfakes, but they have certain gray areas that limit their usefulness. There are some web applications available, such as Deep Fake Detector, Detect Deepfakes, and Alterations. These tools help us to detect whether uploaded videos/movies are real or fake.

While these tools do help in some cases, they nonetheless fail to address important issues like ethical concerns, legal challenges, and public awareness. The above factors are very important when it comes to dealing with the harmful effects of fake videos on society. DeepFakeDetector (https://sightengine.com/detect-deepfakes) has a major limitation, which is that it only deals with analyzing videos of up to 5 megabytes (MB) in size. This is a major issue because in the modern era of digital content, videos are larger in size and of higher quality. As a result, this tool does not fulfill the requirements of the currently uploaded content shared on social platforms. Deepfake videos are becoming more common and more realistic, which makes it difficult to identify whether videos are original or fake [8]. Numerous researchers around the world have proposed various techniques and models to detect fake videos [9].

The following few paragraphs discuss how deepfake detection models work, what kind of datasets they use, how accurate they are, and what limitations they have. This section also provides detail information on the datasets that these researchers have used, such as FaceForensics++, DFDC, and Celeb-DF [10]. Datasets have been used widely in the field and contain both real and fake videos for training and testing models.

Classification plays an important role in deepfake image/video detection [11,12]. After the extraction of features from the video (like facial movements, eye blinking, head motion, or mouth movement), we need a model that can classify them into one of two categories: real or fake. Most researchers prefer to use DL models for this classification task due to their deep learning capabilities. The deep Convolutional Neural Network model, which is used for image classification, is known for its high number of layers and its architecture. Numerous researchers have combined both CNN and Long Short-Term Memory (LSTM) to take advantage of both image-based and time-based features. The CNN extracts features from each of the frames, and the LSTM analyzes the sequence of these features, in order to make the final decision at the end.

In addition to these approaches, some studies use transfer learning with pre-trained models like the Visual Geometric Group-16 (VGG-16), Residual Network 50 (ResNet-50), LSTM, Residual Networks with Aggregated Transformations, and EfficientNet (ResNeXt) [13]. These models are pre-trained on large image datasets, and they can quickly learn new tasks with less training data. Using the technique of transfer learning has sped up the development and improved the accuracy of these models. Although these DL techniques are powerful and have achieved high accuracy, they also present some challenges. They usually require high-performance hardware such as a Graphics Processing Unit or Tensor Processing Unit for the processing of large amounts of data. Also, running them in real-time environments, such as live video detection, can be slow or inefficient [14,15]. In conclusion, classification techniques are always the key part of any deepfake detection system.

Aarti Karandikar et al. [16] proposed a deepfake detection method that makes use of a transfer learning-enhanced VGG-16 CNN. This technique finds visual irregularities, especially between facial features, in order to identify deepfakes. This system had trouble with low-resolution deepfakes and was sensitive to compression artifacts. Similarly, Hanqing Zhao et al. [17] presented a Multilayer Feedforward Neural Network that can record artifacts at different levels of granularity. The model performed well, particularly when it came to identifying low-quality deepfakes. However, the model's high energy and computing costs made real-time adoption challenging. TensorFlow and Keras were used to assess the approach on the FaceForensics++ dataset.

Yash Doke et al. [18] created a hybrid model that uses LSTM to capture temporal discrepancies and ResNetCNN to extract features. By concentrating on facial and kinematic abnormalities, this approach produced high detection accuracy and real-time performance. Its significant processing burden during frame-level analysis was its primary flaw. The model was developed using Python frameworks and trained on FaceForensics++ with a 70% training set and 30% test set. Alakananda Mitra et al. [19] created a deepfake detection model using a CNN combined with a classifier. Despite achieving high precision scores, the model was not resistant to different video quality and compression levels. It was trained using the FaceForensics++ dataset and implemented with Keras and TensorFlow.

Nicolo Bonettini et al. [20] suggested a hybrid approach to identify facial changes in deepfake films by fusing DL methods with conventional computer graphics. Although the CNN-based system was quite accurate, its real-time application was limited by modest processing delays. The FaceForensics++ and Deepfake Detection Challenge (DFDC) datasets in PyTorch were also used to assess the model. Ankur Nagulwar et al. [21] used Multi-task Cascaded Convolutional Networks (MTCNNs) to identify irregularities in noise and face features in video frames. The system successfully authenticated videos with single faces and achieved 70% accuracy. But when there were several faces in a single shot, its performance degraded. The FaceForensics++ dataset, which consists of films that have been transformed into frames for analysis, was used for the evaluation. Nimitt Patel et al. [22] used an LSTM and ResNext CNN to identify video frames in a deepfake detection model. However, the model's usefulness was limited due to the limited visual content examined, and it did not analyze audio. The preprocessing with cropping and resizing, FaceForensics++, was used for training.

Neeraj Guhagarkar et al. [23] provided a thorough analysis of several AI and machine learning-based deepfake detection techniques. They examined temporal and spatial approaches and highlighted high-performing designs, such as CNN+LSTM. The study identified issues with low-quality movies and real-time identification despite the extensive use of datasets such as Celeb-DF, University of Albany Deep Fake Video (UADFV), and Reddit Deepfakes. TensorFlow, Xception, and Visual Geometry Group (VGG) were commonly used tools. Ahmed Hatem Soudy et al. [24] suggested a hybrid deepfake detection system that uses a majority voting method and combines CNNs with vision transformers. The eyes, nose, and entire face were the focal points of their model. Although it needed a lot of processing power and might have missed face alterations in other regions, it was able to reach up to 80% accuracy. The FaceForensics++ and DFDC datasets were used to train and assess the system.

Similarly, Arash Heidari et al. [25] carried out a study of the literature on DL-based deepfake detection techniques. CNNs, GANs, RCNNs, and other models for image, video, and audio detection were discussed in the study. Methods outside of DL were not included; they were grouped according to performance and application. They concentrated on models built using platforms like TensorFlow and Keras and compared them using datasets including CelebA, TIMIT, and DFDC. A detailed yet comprehensive literature review is presented in Table 1.

Table 1. Most relevant studies about fake frame detection in terms of their datasets, methodologies, and experimental environments

Studies	Dataset	Methodology	Simulation Environment	Conclusion
Ju, Hu et al. [26]	DFDC, Celeb-DF, DFD, FaceForensics++ (with demographic annotations)	DAG-FDD (agnostic) and DAW-FDD (aware) models using CVaR loss to ensure fairness in deepfake detection	PyTorch; Dlib reprocessing; NVIDIA RTX A6000; frame size 380 × 380; tested on ResNet50, Xception	Fair detection without losing accuracy
Pei, Zhang et al. [27]	FaceForensics++, CelebDF, DFD, DFDC	Categorize detection techniques and propose OOD benchmarks	Literature-based evaluation without model testing	Cross-domain fairness without accuracy loss
Taeb and Chi [28]	Kaggle Deepfake Detection Challenge (subset)	Used CNN-RNN, 2D/3D CNN, Two-Stream with optical flow and F1/BCE loss	Google Colab with OpenCV and TensorFlow	Achieved 79% accuracy without transformers
Passos, Jodas et al. [29]	FaceForensics++, DFDC, Celeb-DF, WildDeepfake, UADFV	Reviewed 100+ studies using CNNs, RNNs, GANs, and Transformers; categorized methods into spatial, temporal, and hybrid approaches	Focused mainly on facial manipulations; lacked cross-modal and audio analysis	Analytical review without implementation; highlights dataset uses and model categorization
Rajagukguk, Kencana et al. [30]	Kaggle (589 real, 700 fake)	Used ResNet50 with transfer learning for binary face classification	Achieved 76% training and 53% test accuracy, indicating overfitting	Trained in TensorFlow for 30 epochs using data Augmentation and a frozen base model
Orlando and Al Rivan [31]	ISIC 2019 dataset with 25,331 images of 8 skin cancer types	Used CNN models (LeNet and VGG-16) for classification	Implemented in Python using Adam optimizer, data augmentation, batch size 32, 30 epochs	VGG-16 achieved 73.22% accuracy (better than LeNet) but required more training time (354 s per epoch)
Notshe, Xiao et al. [32]	Subset of Kaggle Deepfake Detection Challenge (small and imbalanced)	Tested 2D CNN, 3D CNN, CNN-RNN, and Two-Stream CNNs using both spatial and temporal features	Implemented in Google Colab with TensorFlow and OpenCV; used optical flow	Temporal models (e.g., CNN-RNN) outperformed others (up to 79%); no transformers tested

Table 1. *Cont.*

Studies	Dataset	Methodology	Simulation Environment	Conclusion
Samuel [33]	ImageCLEF med, OASISMRI, TCIA-CT, and NEMA-CT datasets	Fused CNN using shallow and deep layers; applied transfer learning on selected layers	Used Caffe framework, Intel i7 (16 GB RAM), LR 1e-6, momentum 0.9	Achieved better classification; limited by model complexity and small datasets
Alzahrani and Rawat [34]	FaceForensics++ (Face2Face subset)	Used CNNs with optical flow input to detect motion irregularities in deepfake videos	Implemented with PWC-Net and TensorFlow. RGB optical flow processed via VGG16/Res Net50	High accuracy on Face2Face forgeries; limited by dependence on clean optical flow data
Bankar, Pawar et al. [35]	FaceForensics++ (NeuralTextures, FaceSwap, DeepFakes, Face2Face)	Benchmarked deepfake detector using over 1.8Mmanipulatedimages; used XceptionNet for classification	Tested across compression levels; used Face2Face for tracking and frame-level classification	Detection accuracy drops with compression; emphasizes the need for diverse datasets and standard benchmarks

When a fake video, which is generated very carefully and with advanced tools, looks very realistic, these tools are unable to detect the changes made in the video properly. This also leads to inaccurate results, especially when it comes to detecting fake videos that are professionally made/edited. The limitations clearly show the need for a more advanced and more reliable fake video detection system, which can not only process large and high-quality videos but can also provide accurate and trustworthy results. The system should be user-friendly and should be capable of helping media agencies and legal authorities to identify fake content effectively. This study aims to fill the gaps in the research by developing a more efficient and powerful fake video detection tool to overcome these challenges.

The accuracy and dependability of current detection techniques are sometimes insufficient to handle the increasing difficulty of identifying manipulated material, especially deepfakes. Due to this restriction, it is challenging to identify small alterations in video content, which enables bogus videos to aid in the dissemination of false information. The social, political, and media landscapes are at serious risk, and public confidence in digital media is eroded as a result.

The following are the aims and objectives of the study:

- To review the state of the art of the literature in detecting deepfake videos.
- To propose a DL approach for detecting fake videos.
- To build a web application as an application prototype.

This study addresses the issues and challenges that offer reliable and user-friendly problem solutions for verifying whether a video is authentic or not. The system is built to adapt to all these limitations and provide accurate and high-quality results. By using the advanced DL model and user-friendly web interface, the tool can be used for detecting deepfake videos. In addition, we also compared the accuracy of each method to understand how well it performs. Some of the models must achieve high accuracy but may be limited to a specific type of fake video, or they might not perform as well as they are expected to in real-world scenarios where lighting and quality vary. Through this research, by reviewing all these models, datasets, and results, we gain a better understanding of what works perfectly and what does not.

2. Materials and Methods

This study focuses on detecting fake videos, also known as deepfakes, using DL techniques. The methodology includes several key steps: dataset selection, preprocessing, model training, and evaluation, as shown in Figure 1. Each of these steps plays a vital role in ensuring that the model accurately distinguishes real from fake videos. The proposed system is designed for the detection of fake videos using a DL-based approach.

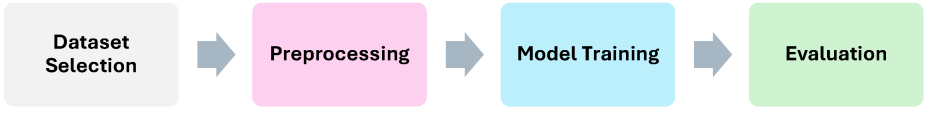

Figure 1. Proposed steps of overall workflow, including dataset selection, preprocessing, model training, and evaluation. These stages are performed sequentially to develop and assess the proposed model.

https://doi.org/10.37978/tijfs.v08i01.004

The architecture shown in Figure 2 includes several components, such as the selection of the dataset, preprocessing of the dataset, selection of the model, model training, video analysis, and the display of the results on the UI. The DL model is trained and evaluated for fake video detection.

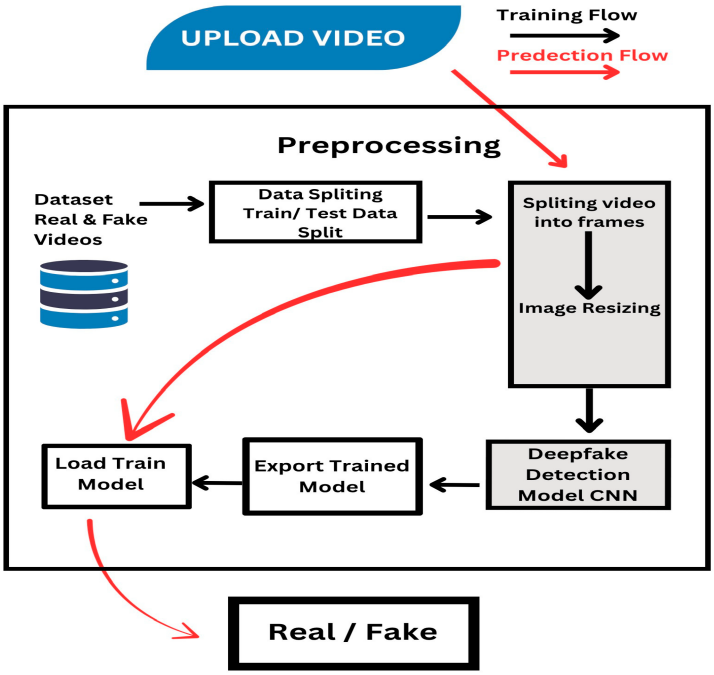

Figure 2. Proposed architecture. This diagram outlines the training and prediction pipeline of a CNN-based deepfake detection system. During training, real and fake video datasets are split, preprocessed into resized frames, and used to train and export a detection model. During prediction, an uploaded video undergoes the same preprocessing steps before being classified as real or fake by the trained model.

2.1. Dataset Selection and Preprocessing

We use the Celeb-DF dataset, which is one of the most well-known and challenging deepfake datasets available for academic and research purposes in deepfakes. The dataset was first introduced in ref. [36]. The main goal of creating this dataset was to address the limitations of earlier deepfake datasets such as University of Albany Deep Fake Video (UADFV) and Faceforensics++, which have noticeable visual artifacts that make the fake content easier to detect.

This dataset contains 408 real videos that were collected from interviews of different celebrities on YouTube, covering both males and females of different ethnicities, ages, and speaking styles. Moreover, this dataset also contains 795 fake videos that were created using advanced deepfake generation techniques, which focus on making the fake videos look more realistic by preserving head poses, lip sync, and facial expression.

The DL model works better with images rather than videos. For this purpose, each video, whether it is real or fake, is first split into individual frames using OpenCV, which is a Python library. Each video is processed frame by frame. Moreover, the proposed frames must be in a proper sequence. The frame rate is kept consistent by extracting eight frames per second from 30 frames per second to avoid redundancy and to reduce processing time. These extracted frames serve as input data samples for model detection.

The images are cropped and resized to a size of 224 × 224. This is necessary because the ResNet50 model expects the input of images to be of a fixed dimension. OpenCV is utilized to convert color formats (such as RGB), resize photos to the necessary input shape (224 × 224 pixels), and carry out simple tasks like cropping or grayscale conversion when necessary. For DL training, OpenCV makes the video-to-frame conversion dependable and effective. The pixel values of the images are also normalized to a scale of between 0 and 1 to reduce the computational load and to help the model train more efficiently. Each image is assigned a label; Label 0 is given to real images, and Label 1 is assigned to fake images.

2.2. Model Selection

To achieve the accuracy and reliability of the model, we have selected the ResNet50 model, which is a powerful and widely used CNN model. This model was first introduced by Microsoft Research in ref. [37].

2.2.1. Comparison with Other CNN Models

The ResNet50 model is a better, more accurate, and more reliable CNN model than other available models. It extracts features and is a lightweight model compared to the VGG16 model. Also, fine-tuning is easily comparable to InceptionV3 and is more accurate compared to MobileNet. A detailed comparison of these models is provided in Table 2.

Table 2. Comparative analysis of commonly used deep learning models for deepfake detection

Model	Depth	Parameters	Accuracy (General)	Suitability for Deepfake Detection
VGG16	16 Layers	138M	Good	Large model, slower training.
VGG19	19 Layers	144M	Good	A very large model, limited to image classification.
Inception V3	48 layers	24M	Excellent	Complex architecture, harder to fine-tune.
MobileNet V2	53 Layers	3.5M	Fair	Lightweight but less accurate.
EfficientNet B0	82 Layers	5.3M	very Good	Efficient, but might underperform on complex tasks.
ResNet50	50 Layers	23M	Excellent	Balanced: good depth, accuracy, and speed.

2.2.2. Training Process

The training set used to train the model contains 70% of the data from the dataset. The testing set is used to evaluate the model after training, and contains 15% of the data from the dataset, and the validation set is used to check how well the model is learning during the training session. It also contains 15% of the data from the dataset.

During training, the model uses the training set, which is already preprocessed with real and fake video frames. The actual goal is to enable the model to properly learn the patterns and features that distinguish the real frames from the fake frames. The Adam optimizer is used for updating the model weights during training. Adam is widely used because it adapts the learning rate during training and combines the benefits of two other optimizers, AdaGrad and RMSProp. It is efficient, fast, and works well for large datasets, as well as for DL models like ResNet50.

TensorFlow manages the training loop, automatically computes gradients, and updates model weights to minimize loss over time. Keras, with its simple syntax, makes the process of defining models, compiling them, and evaluating their performance more intuitive for quick experimentation. These tools are crucial for implementing and testing different DL techniques during the fake video detection pipeline. At a predetermined frame rate (e.g., eight frames per second), each video is divided into individual image frames, which the model then processes separately.

The model is trained for 27 epochs. The early stopping technique monitors the validation loss. When the loss does not improve for a certain number of epochs, the training is automatically stopped. This provides the benefit of avoiding training that might harm the model's ability to generalize. The model checkpoints save the best version of the model, depending on the validation performed automatically. When training is completed, the final model will be saved as Hierarchical Data Format Version 5 (HDF5), which is also popularly known as the .h5 format. This model stores the architecture of the model, weights, and the optimized state, which makes it very easy to reuse the model for prediction and deployment in different web applications.

2.3. Evaluation Metrics

The performance of deepfake detection was evaluated using the following standard classification matrices [38].

2.3.1. Accuracy

Accuracy measures how well the model classifies videos as real or fake. It is calculated as the ratio of correctly classified videos (both real and fake) to the total number of videos. Our system achieved 82.33% accuracy, computed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2.3.2. Precision

Precision measures a model's dependability in identifying a fake video [36]. It determines the percentage of films that are accurately classified as fraudulent (True Positives) compared to all videos that are anticipated to be fraudulent (True Positives + False Positives). Our system achieved 76.70% precision. Precision shows how many of the fake videos detected are in fact fake:

https://doi.org/10.37978/tijfs.v08i01.004

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

2.3.3. Recall

The model's recall measures its capacity to recognize every fake video in the collection [37]. It determines the percentage of fake videos that are accurately identified (True Positives) relative to the total number of fake videos (True Positives + False Negatives). Our system achieved 89.15% recall. Recall shows how many actual fake videos were correctly identified:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

2.3.4. F1-Score

The F1-score is the harmonic mean of recall and precision [38]. It offers a balance between the two metrics, which is particularly helpful in situations when the distribution of classes is not uniform. Our system achieved an F1-score of 82.46%. The F1-score is a balance between precision and recall, calculated as

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The final system allows the user to upload the video through a web interface, which is developed in Streamlit (https://streamlit.io/), Ver 1.59.1. The uploaded file is then processed in real time, extracting frames from the videos by cropping, resizing, and normalizing them. Streamlit, in which the web application is developed, allows users to upload videos and view the results after processing. The backend handles the preprocessing, the loading of the model, and real-time prediction for the user.

3. Results

The Google Colab environment's TensorFlow and Keras frameworks were used for the training procedure. A batch size of 32 samples was used to train the system across 30 epochs. Model weights were updated during training using the Adam optimizer, which was chosen due to its effectiveness while working with sparse gradients and large datasets. During training, a constant learning rate of 0.001 was used. The binary cross-entropy loss function was employed because it works well for two-class classification issues. In general, the training setup was chosen to minimize overfitting and provide strong convergence.

Figure 3 demonstrates the model's ability to discriminate between real and false frames, as shown by the high number of true positives and true negatives. There were a few instances of false positives, in which genuine frames were mistakenly identified as fraudulent.

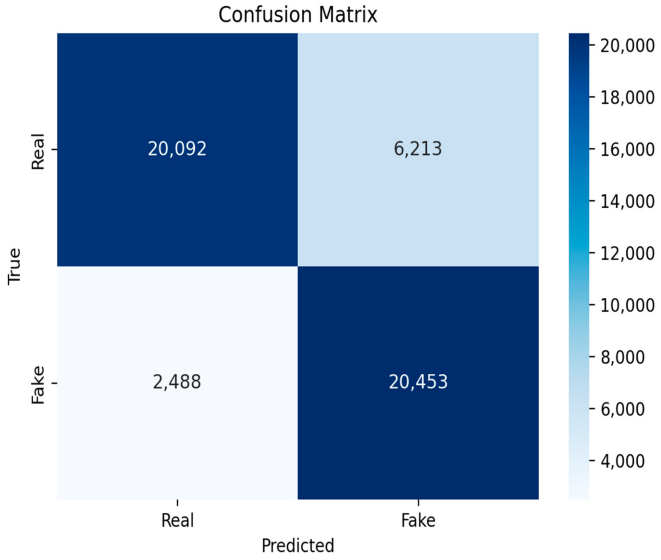

Figure 3. The confusion matrix shows a real vs. fake classifier with 20,092 true negatives, 20,453 true positives, 6213 false positives, and 2488 false negatives out of 49,246 total samples. It further illustrates the classification performance of the proposed binary classification model in distinguishing between real and fake frames.

https://doi.org/10.37978/tijfs.v08i01.004

The confusion matrix presented in Figure 3 illustrates the classification performance of the proposed binary classification model in distinguishing between real and fake frames. The matrix indicates that the model correctly classified 20,092 real samples as real and 20,453 fake samples as fake, resulting in a total of 40,545 correct predictions out of 49,246 test samples. Conversely, 6213 real samples were incorrectly classified as fake, while 2488 fake samples were misclassified as real. These results correspond to an overall classification accuracy of approximately 82.33%, demonstrating that the proposed model effectively differentiates between the two classes while maintaining a relatively low misclassification rate.

These results suggest that the model is more conservative in predicting the real class, thereby reducing the chances of fake samples being accepted as real. Although some genuine samples are falsely identified as fake, the overall distribution of correct predictions demonstrates that the classifier is well-balanced and reliable for practical applications. Figure 3 confirms the robustness of the proposed model and highlights its effectiveness in accurately recognizing both classes while providing opportunities for further optimization to reduce false classifications, particularly for the real class.

The following Figure 4 shows the evaluation metrics of the deepfake detector. These metrics show in detail the classification ability of the model on the test set and allow us to determine how well the system can distinguish between real and fake videos. A high precision value indicates that most of the videos classified as fake are fake, reducing the false positive cases, while a high recall value indicates that many of the fake videos are properly identified by the model, therefore reducing the cases of false negatives. F1-score, as the harmonic mean of precision and recall, reveals the similarity in the results of those two measures, as well as the reliability of the grouping model in terms of the two classes. Lastly, there is the overall accuracy of the model with reference to the accurate classification of samples on the test set.

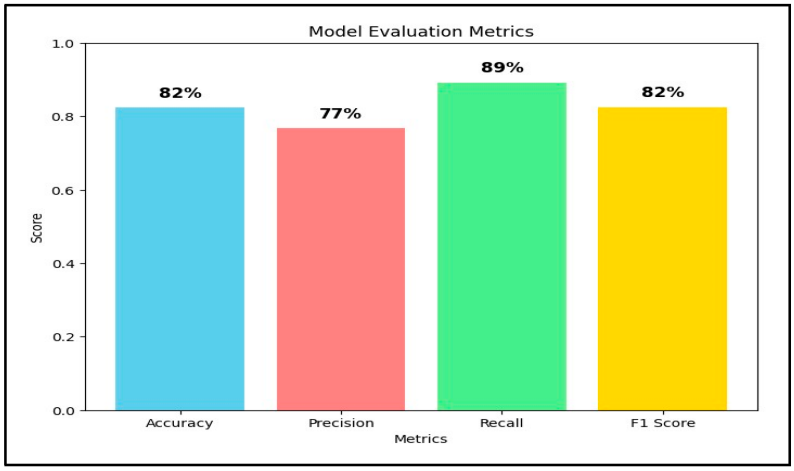

Figure 4. The model achieves an accuracy of 82%, precision of 77%, recall of 89%, and an F1 score of 82%, indicating strong overall performance with a particular strength in correctly identifying positive cases (high recall) relative to precision.

The following Figure 5 displays the model's training and validation losses over 30 epochs. As can be observed, the training loss gradually drops, suggesting that the model is successfully picking up on trends in the data. The model appears to be generalizing well with new data, as evidenced by the validation loss's comparable trend and its close alignment with the training curve. One indication that there is little overfitting is the lack of significant gaps or oscillations between the two curves.

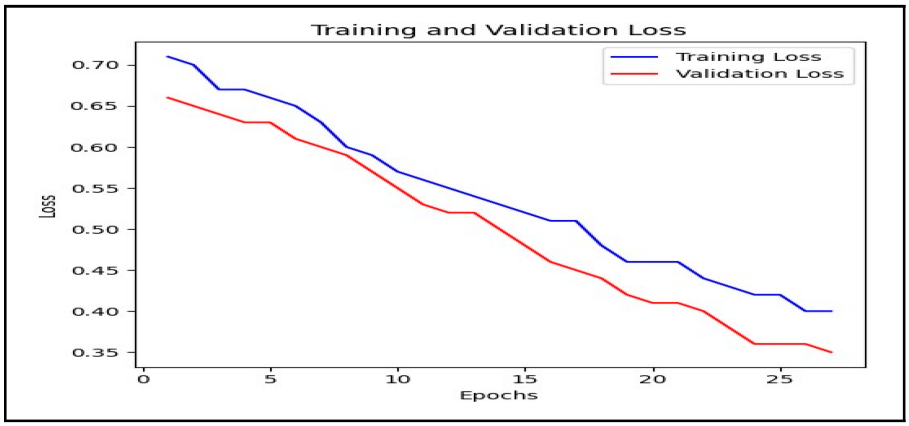

Figure 5. Training and validation loss over epochs.

https://doi.org/10.37978/tijfs.v08i01.004

Similarly, Figure 6 below further illustrates the training and validation accuracy of the proposed ResNet-50 model over 27 training epochs. The graph demonstrates a steady improvement in classification performance throughout the training process, indicating that the model effectively learns meaningful feature representations from the training data. The training accuracy increases consistently from approximately 53% in the first epoch to 80% in the final epoch, reflecting gradual convergence and stable optimization. Similarly, the validation accuracy improves from approximately 56% to 83%, showing that the model generalizes well to previously unseen data.

It should be noted that the validation accuracy remains slightly higher than the training accuracy during most training epochs. This behavior suggests that the model benefits from effective regularization techniques, such as data augmentation, dropout, or batch normalization, which help prevent overfitting and improve generalization performance. Moreover, the relatively small gap between the two curves indicates that the model has achieved a good balance between learning the training data and maintaining strong predictive performance on the validation set.

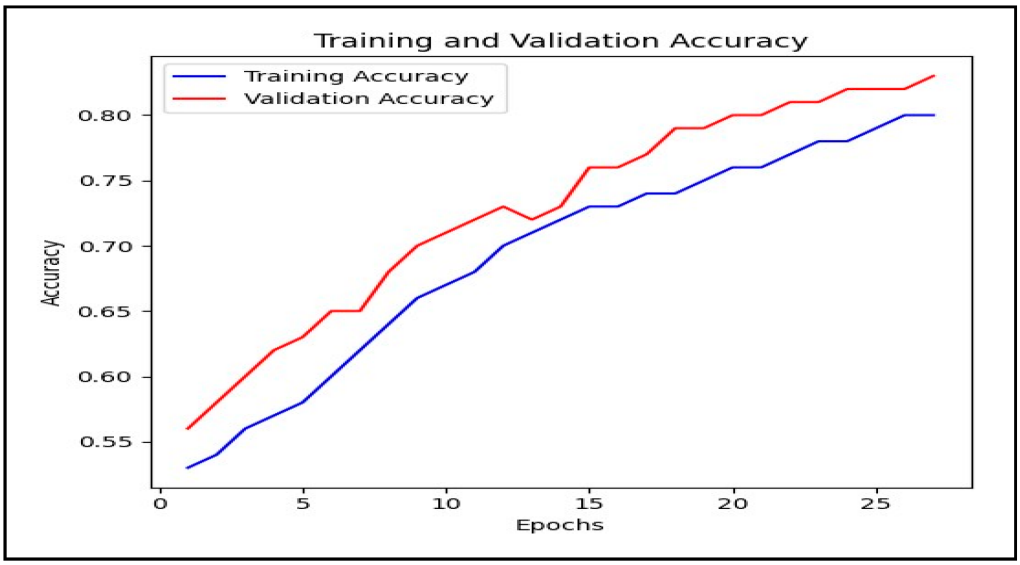

Figure 6. Training and validation accuracy over epochs. Both training and validation loss decreased steadily over 27 epochs, from around 0.71 and 0.66 to approximately 0.40 and 0.35, respectively, indicating consistent learning without signs of overfitting, as the validation loss remains close to and even below the training loss throughout.

A steady rising trend is seen in both curves, suggesting that performance has improved over time. Training that is balanced and successful is indicated by both curves increasing steadily and without divergence.

3.1. Model Performance Across Selected Epochs

The model training history at certain epochs, the first, fifth, tenth, fifteenth, twenty-fifth, and twenty-seventh epochs, is displayed in Table 3.

Table 3. Model training history at selected epochs.

Epoch	Accuracy	Val Accuracy	Loss	Val Loss
1	0.53	0.56	0.71	0.66
5	0.58	0.63	0.66	0.63
10	0.67	0.71	0.57	0.55
15	0.73	0.76	0.52	0.48
20	0.76	0.80	0.46	0.41
25	0.79	0.82	0.42	0.36
27	0.82	0.83	0.40	0.35

Table 3 provides the results for training loss, validation loss, training accuracy, and validation accuracy. On both the training and validation datasets, the model's accuracy steadily increased as training went on, while the loss values continuously dropped. These findings show that the model was learning efficiently and performing well in generalization on data that had not yet been seen.

https://doi.org/10.37978/tijfs.v08i01.004

3.1.1. Model Performance Comparison

The prior study shown in Table 4 relies on the ResNet-50 model, but uses it with different data and training strategies. In that study, the focus lies on image-level detection, followed by using preprocessing methods, i.e., Local Binary Pattern (LBP) and Gaussian filtering, to enhance feature extraction and the corresponding performance. The second study focuses on video-level classification, where a second layer of binary classification is added, and a binary cross-entropy loss is initialized to learn from a larger amount of classification data. The comparison generally reveals that the size of the dataset and preprocessing and training approaches are also important in identifying the authenticity of deepfake detection models. The model comparison is given in Table 4.

Table 4. Model comparison.

Author	Dataset	Methodology	Model Used	Simulation Environment	Conclusion
Arini et al. [39]	Celeb-DF (2000 images: 1000 real/1000 deepfake), JPG format, 224 × 224 Pixels	Gaussian Filter + LBP	ResNet-50	20 epochs, batch size 100, Adam optimizer, cross-entropy loss	Performs well on a small dataset. LBP and filtering enhance results
Proposed	Celeb-DF which has 408 real videos and 795 fake videos	A binary classification layer is added on top + binary cross-entropy	ResNet-50	30 epochs, batch size is 32, Adam optimizer	perform Well, on the large dataset, and enhance the accuracy.

3.1.2. Performance Metrics Comparison

As can be seen in Table 1, the proposed model based on ResNet-50 performed well and showed an accuracy of 0.82, which indicates the overall correctness of the prediction. These metrics of 0.76 and 0.89 indicate that the majority of detected deepfakes were reported correctly, and the model was very efficient in identifying most of the real deepfake instances. In addition, the F1-score of 0.82, which is balanced with precision and recall, is a further indication of the stability of the model because it ensures both high accuracy and coherence in various cases.

These findings indicate that the ResNet-50 network shows an appropriate balance and offers an effective deepfake-detection option overall. The following is the performance metrics comparison, which is given in Table 5.

Table 5. Performance metrics of the ResNet-50 model.

	Model	Accuracy	Precision	Recall	F1-Score
Existing	ResNet-50	0.75	0.80	0.68	0.77
Proposed	ResNet-50	0.82.33	0.76.70	0.89.15	0.82.46

4. Discussion

The proposed model achieved 82.33% accuracy, 76.70% precision, 89.15% recall, and an 82.46% F1-score. These evaluation results show that the model performs well in distinguishing between real and fake videos. The training and validation curves represent stable learning behavior without signs of overfitting, while the confusion matrix confirms the high number of correct predictions. We propose the use of this system for real-world deployments in media forensics and content moderation platforms. The outcomes and implementation specifics of the deepfake detection system have been shown, and we started by explaining the experiment configuration, namely the instruments, libraries, and hardware used for performing the DL model training and testing process.

Table 5 presents a comparative performance analysis of the existing and proposed ResNet-50 models using four standard evaluation metrics: accuracy, precision, recall, and F1-score. The results demonstrate that the proposed model consistently outperforms the baseline ResNet-50 across all major performance indicators. Specifically, the existing ResNet-50 achieved an accuracy of 75.00%, whereas the proposed approach improved the classification accuracy to 82.33%, representing a significant enhancement in the model's overall predictive capability.

Furthermore, the proposed model achieved a precision of 76.70%, a recall of 89.15%, and an F1-score of 82.46%, compared with the existing ResNet-50, which obtained 80.00%, 68.00%, and 77.00%, respectively. Although the proposed model exhibits a slightly lower precision than the baseline, it demonstrates a substantial improvement in recall, indicating a much stronger ability to correctly identify positive instances while significantly reducing false negatives. The higher

F1-score confirms that the proposed approach provides a better balance between precision and recall, leading to superior overall classification performance. These results validate the effectiveness of the proposed enhancements to the ResNet-50 architecture and demonstrate its suitability for accurate and reliable classification in the target application.

The surroundings were properly organized to be acceptable, so that the system could be efficiently trained and tested precisely. The evaluation metrics were accuracy, precision, recall, and F1-score. They assisted in assessing the effectiveness of the model in identifying deepfake videos versus real videos. A confusion matrix was also used to provide a clear insight into the correct or wrong number of videos that were identified as real or fake. It assisted in determining the strengths and weaknesses of the model, in particular, where the model went wrong. The training and validation accuracy were also plotted, demonstrating the level of model performance during training and on validation data. These outcomes proved that the model was not overfitting.

5. Conclusions

This study introduces a deepfake video detection system based on DL, namely a CNN model based on ResNet50. The utilization of deepfakes represents a significant problem because it leads to false information, reputation destruction, a potential impact on national security. Our system includes preprocessing of videos by the extraction of certain important frames, identifying and scaling faces, and converting them to a suitable format. The ResNet50 model used has a particularly strong architecture in terms of its image recognition abilities, and the model's performance, indicated by the accuracy, precision, recall, and F1-score, was reasonably good. Although there are limitations to this study, such as the large amounts of data and high-performance equipment required, this study shows that DL is potentially useful when combating fake media and misinformation.

However, within the scope of this work on detecting deepfake videos through the classification and visual analysis of images, there are many domains in which improvements can be achieved. Another potential direction of study is related to introducing audio analysis, because many deepfakes also exploit speech editing by using AI-generated voice models. Future research may focus on speech patterns, tone, frequency, and lip sync to analyze videos using DL and signal processing, aiming for higher accuracy in detecting elements such as inconsistencies between audio and audiovisual data. Another critical area is the improvement of the user interface.

Author Contributions: Conceptualization, I.A.; methodology, M.A.K. and E.U.; software, M.A.K. and E.U.; validation, I.A., M.A.K. and E.U.; formal analysis, I.A.; investigation, M.A.K. and E.U.; resources, I.A.; data curation, M.A.K. and E.U.; writing—original draft preparation, M.A.K. and E.U.; writing—review and editing, I.A.; visualization, M.A.K. and E.U.; supervision, I.A.; project administration, I.A.; funding acquisition, N/A. All authors have read and agreed to the published version of the manuscript.

Funding: This study received no external funding from any source.

Data Availability Statement: The source code, implementation, and user interface (UI) developed for this study are publicly available to facilitate reproducibility, validation, and further research. Interested researchers can access the frontend and backend repositories through the following GitHub links: Frontend (User Interface): https://github.com/ehaab212/fake-video-detection-frontend (accessed on 2 August 2026); Backend (Model and API): https://github.com/ehaab212/fake-video-detection-backend (accessed on 2 August 2026).

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

References

1. Chadha, A.; Kumar, V.; Kashyap, S.; Gupta, M. Deepfake: An Overview. In *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security. Lecture Notes in Networks and Systems*; Singh, P.K., Wierzchoń, S.T., Tanwar, S., Ganzha, M., Rodrigues, J.J.P.C., Eds.; Springer: Singapore, 2021; Volume 203. [CrossRef]
2. Gaur, L.; Arora, G.K.; Jhanjhi, N.Z. Deep learning techniques for creation of deepfakes. In *DeepFakes*; CRC Press: Boca Raton, FL, USA, 2022; pp. 23–34. [CrossRef]
3. Asim, M.; Waqar, M.; Alam, I. A Comparative Analysis of Deep Learning Methods for Slang Detection in Twitter Data. *Spectr. Eng. Sci.* **2025**, *3*(12), 254–270. [CrossRef]
4. Yessimova, M.; Shevyakova, T. Deep Fakes in the Digital Media Age: Opportunities and Threats. *Her. J. Žurnalistika Seriâsy* **2024**, *73*(3), 44–54. [CrossRef]
5. Majerczak, P.; Strzelecki, A. Trust, media credibility, social ties, and the intention to share towards information verification in an age of fake news. *Behav. Sci.* **2022**, *12*(2), 51. [CrossRef]
6. Mirsky, Y.; Lee, W. The creation and detection of deepfakes: A survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*(1), 1–41. [CrossRef]
7. Fatima, S.; Ali, Z.; Alam, I.; Muhammad, S.; Jan, G. Unmasking Hate: Deep Learning-Based Detection of Violent Incitement in Social Media. *Annu. Methodol. Arch. Res. Rev.* **2025**, *3*(12), 138–160. [CrossRef]
8. Goh, D.H.L. "He looks very real": Media, knowledge, and search-based strategies for deepfake identification. *J. Assoc. Inf. Sci. Technol.* **2024**, *75*(6), 643–654. [CrossRef]

9. Jin, X.; Yi, K.; Xu, J. MoADNet: Mobile asymmetric dual-stream networks for real-time and lightweight RGB-D salient object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*(11), 7632–7645. [CrossRef]
10. Sharma, V.K.; Rawat, S. Enhancing Deepfake Detection Through Dynamics of Facial Expressions. In Proceedings of the 2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 17–19 June 2025; pp. 70–79. [CrossRef]
11. Mansoor, A.; Zaheen, A.; Ali, Z.; Idrees, F.; Rahim, M.; Jan, G.; Alam, I. Enhancing thyroid ultrasound diagnosis with a hybrid CNN and graph attention network. *Spectr. Eng. Sci.* **2025**, *3*, 95–105. [CrossRef]
12. Jin, X.; Jing, P.; Wu, J.; Xu, J.; Su, Y. Visual sentiment classification via low-rank regularization and label relaxation. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *14*(4), 1678–1690. [CrossRef]
13. Jin, X.; Yu, W.; Chen, D.-W.; Shi, W. DFD-NAS: General deepfake detection via efficient neural architecture search. *Neurocomputing* **2025**, *619*, 129129. [CrossRef]
14. Alam, I.; Basit, A.; Ziar, R.A. Utilizing Age-Adaptive Deep Learning Approaches for Detecting Inappropriate Video Content. *Hum. Behav. Emerg. Technol.* **2024**, *2024*(1), 7004031. [CrossRef]
15. Jin, X.; Guo, C.; He, Z.; Xu, J.; Wang, Y.; Su, Y. FCMNet: Frequency-aware cross-modality attention networks for RGB-D salient object detection. *Neurocomputing* **2022**, *491*, 414–425. [CrossRef]
16. Karandikar, A.; Deshpande, V.; Singh, S.; Nagbhidkar, S.; Agrawal, S. Deepfake video detection using convolutional neural network. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 1311–1315. [CrossRef]
17. Zhao, H.; Zhou, W.; Chen, D.; Wei, T.; Zhang, W.; Yu, N. Multi-attentional deepfake detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2185–2194. [CrossRef]
18. Doke, Y.; Dongare, P.; Marathe, V.; Gaikwad, M.; Gaikwad, M. Deepfake video detection using deep learning. *Int. J. Res. Publ. Rev.* **2022**, *2582*, 7421.
19. Mitra, A.; Mohanty, S.P.; Corcoran, P.; Kougianos, E. A novel machine learning based method for deepfake video detection in social media. In Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly INiS), Chennai, India, 14–16 December 2020; pp. 91–96. [CrossRef]
20. Bonettini, N.; Cannas, E.D.; Mandelli, S.; Bondi, L.; Bestagini, P.; Tubaro, S. Video face manipulation detection through ensemble of cnns. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5012–5019. [CrossRef]
21. Suratkar, S.; Kazi, F. Deep fake video detection using transfer learning approach. *Arab. J. Sci. Eng.* **2023**, *48*(8), 9727–9737. [CrossRef]
22. Patel, N.; Jethwa, N.; Mali, C.; Deone, J. Deepfake video detection using neural networks. *ITM Web Conf.* **2022**, *44*, 03024. [CrossRef]
23. Guhagarkar, N.; Desai, S.; Vaishyampayan, S.; Save, A. Deepfake Detection Techniques: A Review. *VIVA-Tech Int. J. Res. Innov. IJRI* **2021**, *1*(4), 1–10. [CrossRef]
24. Soudy, A.H.; Sayed, O.; Tag-Elser, H.; Ragab, R.; Mohsen, S.; Mostafa, T.; Abohany, A.A.; Slim, S.O. Deepfake detection using convolutional vision transformers and convolutional neural networks. *Neural Comput. Appl.* **2024**, *36*(31), 19759–19775. [CrossRef]
25. Heidari, A.; Jafari Navimipour, N.; Dag, H.; Unal, M. Deepfake detection using deep learning methods: A systematic and comprehensive review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2024**, *14*(2), e1520. [CrossRef]
26. Ju, Y.; Hu, S.; Jia, S.; Chen, G.H.; Lyu, S. Improving fairness in deepfake detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 4655–4665. [CrossRef]
27. Pei, G.; Zhang, J.; Hu, M.; Zhang, Z.; Wang, C.; Wu, Y.; Zhai, G.; Yang, J.; Tao, D. Deepfake generation and detection: A benchmark and survey. *arXiv* **2024**, arXiv:2403.17881.
28. Taeb, M.; Chi, H. Comparison of deepfake detection techniques through deep learning. *J. Cybersecur. Priv.* **2022**, *2*(1), 89–106. [CrossRef]
29. Passos, L.A.; Jodas, D.; Costa, K.A.; Souza Júnior, L.A.; Rodrigues, D.; Del Ser, J.; Camacho, D.; Papa, J.P. A review of deep learning-based approaches for deepfake content detection. *Expert Syst.* **2024**, *41*(8), e13570. [CrossRef]
30. Rajagukguk, N.; Kencana, I.P.E.N.; Kusuma, I.G.L.W. Classification of Original and Fake Images Using Deep Learning-Resnet50. In *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*; Springer Nature: Berlin, Germany, 2024; p. 51. [CrossRef]
31. Orlando, O.; Al Rivan, M.E. Klasifikasi jenis kanker kulit manusia menggunakan convolution neural network. *MDP Stud. Conf.* **2023**, *2*(1), 144–150. [CrossRef]
32. Notshe, H.; Xiao, S.; Svoboda, T. Deep Learning Deepfake Detection. 2024. Available online: https://cs231n.stanford.edu/2024/papers/deep-learning-deepfake-detection.pdf (accessed on 18 July 2026).
33. Samuel, F. Deep Learning vs. Shallow Learning in Detecting Anomalies in Biomedical Images. *researchgate. Net*. 2024. Available online: https://www.researchgate.net/publication/385084822_Deep_Learning_vs_Shallow_Learning_in_Detecting_Anomalies_in_Biomedical_Images (accessed on 18 July 2026).
34. Alzahrani, A.; Rawat, D.B. Enhance Deepfake Video Detection Through Optical Flow Algorithms-Based CNN. In *International Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 14–22. [CrossRef]
35. Bankar, V.; Pawar, D.R.; Yannawar, P.L. Hybrid ResNeXt and LSTM Model for Enhanced Deepfake Detection on the FaceForensics++ Dataset. *Technology* **2025**, *13*, 14. [CrossRef]
36. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S. Celeb-df: A large-scale challenging dataset for deepfake forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3207–3216. [CrossRef]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]

38. Yacouby, R.; Axman, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Online, 20 November 2020; pp. 79–91. [CrossRef]
39. Arini, A.; Bahaweres, R.B.; Al Haq, J. Quick classification of xception and resnet-50 models on deepfake video using local binary pattern. In Proceedings of the 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), Jakarta, Indonesia, 29–30 January 2022; pp. 254–259. [CrossRef]