Towards a better classification of deepfake videos

Matheus P. R. Vieira[1], **Arthur Negrão de F. M. C.**[2], **Guilherme A. L. Silva**[2],
Eduardo Luz[1], **Pedro Silva**[1]

[1] Computing Department – Federal University of Ouro Preto (UFOP)
35400-000 – Ouro Preto – MG – Brazil

[2]Postgraduate Program in Computer Science – Federal University of Ouro Preto
35400-000 – Ouro Preto – MG – Brazil

{matheus.peixoto,arthur.negrao,guilherme.lopes}@aluno.ufop.edu.br
{eduluz,silvap}@ufop.edu.br

Abstract. *The increasing accessibility and realism of deepfake technology pose serious threats to information integrity, privacy, and public trust. In this work, we propose a deep learning model for detecting deepfake videos based on short video segments. The architecture combines a TimeDistributed wrapper over an Xception backbone with global average pooling and a dense classification layer. To enhance performance, a Bayesian hyperparameter optimization strategy was employed, tuning both architectural and training parameters. The proposed model was evaluated on the Celeb-DF-V2 dataset, using only 24 frames per video, approximately one second of content. Despite this constraint, the model achieved a competitive accuracy of 97.30% and an AUC of 0.9957, outperforming several existing approaches that rely on longer video sequences. These results demonstrate the feasibility of detecting deepfakes efficiently using shorter clips, suggesting a viable direction for real-time and resource-constrained applications.*

1. Introduction

The term *deepfake* refers to synthetic media created using deep learning techniques, primarily in the form of manipulated videos. These media are often highly realistic and convincing, typically involving alterations to facial features and voices [Mirsky and Lee 2021]. Generative Adversarial Networks (GANs) are among the most commonly used deep learning models for producing such content [Alanazi and Asif 2024].

The manipulation of data to create deepfake videos poses significant societal and security challenges. For instance, presidential candidate Muharrem İnce withdrew from the election after fabricated intimate videos resembling him circulated on social media platforms [Kirby 2023], a situation where a misclassification of the deepfake presence could lead to a worse image crisis. In another high-profile incident, a multinational financial firm in Hong Kong was defrauded of approximately 25 million USD. The fraud occurred after an employee joined a video conference with individuals who appeared and sounded like senior executives but were, in fact, deepfakes [Chen and Magramo 2024].

Beyond these alarming incidents, the proliferation of deepfake content on the internet has grown substantially. According to [Łabuz and Nehring 2024], the number of deepfake videos circulating online has tripled in recent years. This increasing prevalence highlights the urgent need for robust and reliable detection mechanisms, a topic that has attracted significant attention in the recent literature.

Among the various machine learning approaches, deep learning architectures have emerged as the most widely adopted for deepfake detection. Deep neural networks have proven highly effective in identifying manipulated media, often achieving impressive accuracy rates. For instance, [Singh et al. 2020] reported an accuracy of 97.63% using only 30 video frames, equivalent to approximately one second of footage.

The work in [Reis and Ribeiro 2024] reports an accuracy of 97.50% using 10 to 13 videos with 50 frames per prediction. In the same dataset, [Hernandez-Ortega et al. 2022] reported an accuracy of 99.14% using five-second videos, increasing this value to 99.47% when using seven-second videos. Even though these high accuracies are impressive, they can lead to increased computational complexity due to the amount of data required.

These findings raise a critical question: *can current deepfake detection techniques be further optimized to reliably identify manipulations in shorter video segments, where facial alterations are often more subtle, while still maintaining high accuracy?* In response to this question, the present work proposes a model designed to detect deepfakes in short video segments (one second length). The model achieves an accuracy of 97.30% and an AUC of 0.9957 on the Celeb-DF v2 dataset, the same dataset used in several of the aforementioned studies, including [Reis and Ribeiro 2024, Hernandez-Ortega et al. 2022].

This paper is structured as follows. Section 2 reviews related work. Section 3 details the proposed methodology. Section 4 presents the experiments and results. Finally, Section 5 concludes the paper.

2. Related Works

Computer-generated human images often lack physiological traits, hindering heart rate estimation models. Leveraging this, [Hernandez-Ortega et al. 2022] proposed a deepfake detector exploiting the absence of such signals. On the Celeb-DF v2 dataset, it achieved 98.70% accuracy at the frame level, rising to 99.47% with a 7-second temporal window, and 100% by incorporating statistical video distributions.

In [Singh et al. 2020], using the DFDC dataset, 30 frames were extracted per video, with the face region extended by 35% beyond the region of interest. The model architecture included EfficientNet-B1, a temporal distribution layer, and an LSTM. This approach yielded an accuracy of 97.63%.

In [de Lima et al. 2020], the Celeb-DF dataset was preprocessed using the RetinaFace library to automate face extraction. Various models were trained, achieving a maximum accuracy of 98.26% using 3D convolutional neural networks with 16 consecutive frames.

To tackle class imbalance in Celeb-DF, [Masud et al. 2023] generated 15-second videos by stitching random 3-second clips, ensuring clear real/fake separation. A VGG16 with time-distributed layers and LSTM achieved 98.24% accuracy on 20 equally spaced frames, though overlap between training and test data was not addressed.

[Heo et al. 2023] proposed a model that extracts local features using EfficientNetB7 and extracts global context with a Vision Transformer, with a knowledge distillation mechanism to enhance learning. Despite its high computational cost, the model achieved an AUC of 0.982 on DFDC and 0.993 on Celeb-DF v2.

Exploring cases where the identity of the person is known, [Reis and Ribeiro 2024] propose a model that compares 50 frames from the questioned video with 9 to 11 real videos, achieving an AUC of 0.994 and 97.5% accuracy on Celeb-DF v2.

[Khormali and Yuan 2021] combines CNNs with attention mechanisms, processing 20% of Celeb-DF v2 frames via RetinaFace. Feature maps are fused with attention maps, which are selectively enhanced or suppressed. Using ResNet, Xception, VGG, and MobileNet, the model achieved AUCs of 0.987, 0.978, 0.980, and 0.948, respectively.

Exploiting the generalization from the CLIP's Vision Transformer, [Yermakov et al. 2025] trained only a binary classifier on the FaceForensics++ dataset, using 32 face-cropped frames with a 30% area increased was used. Cross-dataset validation yielded AUCs of 96.62 on Celeb-DF v2, 87.15 on DFDC, and 98.00 on DFD.

Avoiding neural networks, [Chen et al. 2021] used Successive Subspace Learning (SSL) with just seven frames per video. Faces and patches were processed by PixelHop++, reduced via PCA, and classified with XGBoost. An ensemble yielded AUCs of 100 (UADFV), 94.95 (Celeb-DF), and 90.56 (Celeb-DF v2).

Considering the contributions of the aforementioned studies summarized in Table 1, while [Chen et al. 2021] utilizes an extremely low number of frames, others explore larger amounts, such as [Hernandez-Ortega et al. 2022] and [Reis and Ribeiro 2024]. The present work proposes a neural network to detect deepfakes in one-second videos using only 24 frames, a common frame rate in digital media formats [Wilcox et al. 2015], making it a practical and efficient choice. Moreover, as highlighted by [Heidari et al. 2024], there is growing interest in methods that combine high accuracy with improved temporal efficiency, a goal that can be addressed by using 24 frames during neural network training.

Work	Time	Dataset	Accuracy	AUC
[Hernandez-Ortega et al. 2022]	7 secs	Celeb-DF v2	99.47%	0.9999
[Singh et al. 2020]	30 frames	DFDC	97.63%	–
[de Lima et al. 2020]	16 frames	Celeb-DF	98.26%	–
[Masud et al. 2023]	20 frames	Celeb-DF	98.24%	–
[Heo et al. 2023]	32 frames	DFDC / Celeb-DF v2	– / –	0.982 / 0.993
[Reis and Ribeiro 2024]	50 + 50*9	Celeb-DF v2	97.50%	0.994
[Khormali and Yuan 2021]	Not Informed	Celeb-DF v2	–	0.987
[Yermakov et al. 2025]	32	Celeb-DF v2	–	0.966
[Chen et al. 2021]	7	Celeb-DF v2	–	0.906

Table 1. Summary of works and results

3. Methodology

In this section, we present the dataset used in the experiments, along with the proposed model and the evaluation metrics employed to assess its performance.

3.1. Dataset

The dataset employed in this study is Celeb-DF-V2 [Le et al. 2021], a high-quality video dataset comprising recordings of various celebrities across a wide range of scenarios.

This dataset comprises a total of 890 authentic videos and 5,639 videos synthesized using deepfake techniques. For evaluation purposes, the original authors defined a test subset consisting of 178 real videos (20% of the authentic set) and 340 deepfake videos (6% of the manipulated set). The remaining 712 authentic videos and 5,299 synthesized videos were used for training and validation. Following this protocol, we adopt the same data split to report the results of our experimental analysis.

A detailed examination of the dataset reveals that the average frame rate is not consistent across all videos, ranging from 28.75 to 30.0 frames per second. In terms of video duration, there is substantial variability: the shortest video consists of only a single frame, whereas the longest contains up to 740 frames. On average, videos span 379 frames, which corresponds to approximately 12.6 seconds of playback.

Additionally, the spatial resolution of the videos also varies. On average, the frame dimensions are 813 pixels in width and 480 pixels in height.

3.1.1. Dataset Pre-Processing

Given our focus on evaluating the feasibility of using shorter video segments, we adopted a strategy based on 24-frame clips, which correspond to approximately one second of footage. This frame count was selected due to its widespread adoption across various media formats, particularly in film and video production, where 24 frames per second is a standard frame rate [Wilcox et al. 2015]. Furthermore, this choice helps reduce model complexity, thereby enhancing computational efficiency without sacrificing performance.

Due to the natural imbalance in the dataset, where authentic videos are significantly underrepresented, we applied a data augmentation strategy to mitigate the skew. In the training and validation sets, the ratio of real to fake videos is 712 to 5,299, corresponding to approximately 14%. In other words, for every real video, there are about 7.44 fake ones. To address this imbalance, we extracted only one 24-frame segment from each fake video, while generating multiple segments from the authentic videos, trying to match the number of real videos to the number of fake ones, which are very unbalanced.

The test dataset comprises 178 real and 340 fake videos, as pre-selected by the dataset authors [Li et al. 2020]. This predefined split was preserved to ensure the reproducibility of experiments and to allow direct comparison with existing results.

To build a balanced dataset for training and validation, it was necessary to address the disparity in the number of real videos. A practical solution was to generate multiple sub-clips from each original real video. Initially, the validation set was configured to contain the same number of fake videos as the test set (i.e., 340). However, this setup led to a significant imbalance in the training data. Through manual inspection, it was determined that transferring 14 fake videos from the validation set to the training set would improve balance, resulting in a final validation dataset with 326 fake videos and 163 real videos.

Aiming to match the number of real and fake videos on the validation dataset, two 24-frame segments were extracted from each real video. For training, we used 4,973 fake video clips and extracted nine 24-frame segments from each of the 549 real videos, yielding a total of 4,932 segments per class, ensuring a balanced training set.

It's important to note that data augmentation is applied only during training and validation, not during testing, so it does not affect the final evaluation metrics. Additionally, all segments are non-overlapping and extracted sequentially; this is, each starts right after the previous one ends.

To be included in the validation dataset, a video was required to have at least 48 frames, enough for two 24-frame segments. For inclusion in the training dataset, a minimum of 256 frames was necessary, corresponding to nine segments. One video containing only one frame was excluded from the dataset. Videos with durations between 48 and 215 frames were assigned exclusively to the validation set, totaling seven instances.

All the remaining videos were randomly assigned to either the training or validation sets prior to segment extraction. This ensured that no single video contributed clips to both sets, maintaining a strict separation between training and validation data. Table 2 presents a summary of the number of videos included in each subset of the dataset.

	Real	Fake	Total
Train	$548 \times 9 = 4932$	4973	9905
Validation	$163 \times 2 = 326$	326	652
Test	178	340	518
Total	5436	5639	11075

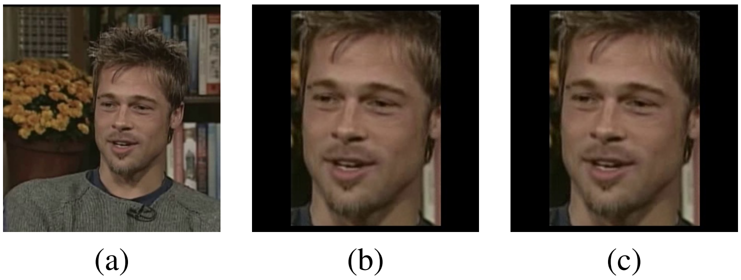
Table 2. Videos split.

Following the methodology proposed in [de Lima et al. 2020, Singh et al. 2020], face regions were extracted from the videos with an additional 10% padding. This approach prevents excessively tight cropping and provides additional contextual information. The extracted face frames were then resized to a fixed resolution of 299 × 299 pixels while preserving the original aspect ratio to avoid distortions caused by interpolation. Any extra space resulting from this resizing was padded with zeros (black borders), in contrast to previous works where the face typically occupied the entire image.

Face detection and extraction were performed using the RetinaFace model [Serengil and Ozpinar 2020], selected for its superior accuracy and lower false detection rate compared to other available models, as noted in [de Lima et al. 2020].

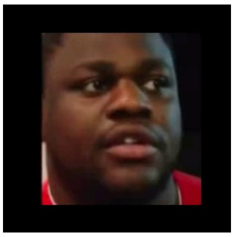
Figure 1 illustrates the face extraction process while maintaining the original aspect ratio. If either the height or width of a face exceeded 299 pixels, it was resized proportionally, capping the largest dimension at 299 pixels. For example, in Figure 1(c), the original height exceeded the frame limit and was adjusted accordingly.

Figure 1. Original image in (a), the cropped in (b) and the cropped image keeping the aspect ratio in (c). Original image from [Li et al. 2020].


(a) (b) (c)

In cases where the region of interest was smaller than 299 × 299, the face was centered within the frame, and the remaining space was filled with black padding, as illustrated in Figure 2.

Figure 2. Area of interest smaller than 299×299. Original image from [Li et al. 2020].



3.2. Proposed Model

The architecture proposed in this study employs the Xception network as a backbone for transfer learning from ImageNet. Although not among the most commonly used architectures, Xception has been investigated in [Khormali and Yuan 2021], which demonstrated its high potential for performance improvement. As the original Xception model is designed for feature extraction from still images rather than sequential video data, it is incorporated within a TimeDistributed wrapper. This configuration enables the model to independently process each of the 24 frames in a video using shared Xception parameters, generating a four-dimensional tensor that encodes spatiotemporal features across the sequence.

This output must be reduced to a lower-dimensional representation suitable for connection to a fully connected layer, allowing classification. While a flattening operation is a straightforward solution, it may result in a high number of trainable parameters, which can lead to increased computational cost and potential overfitting. To mitigate this, a three-dimensional global average pooling layer (GlobalAveragePooling3D) is employed as a more efficient alternative, reducing the dimensionality while preserving salient information.

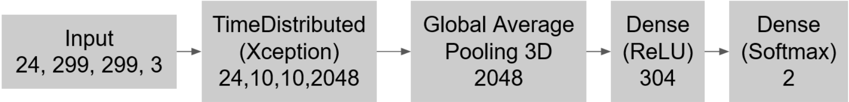
Subsequently, a fully connected output layer is applied to perform the final classification. As highlighted by [Zhang et al. 2023], identifying the optimal combination of hyper-parameters for such a model is a non-trivial task, as even minor variations can significantly influence training dynamics and convergence. To automate this process and enhance performance, hyperparameter tuning techniques were utilized for the baseline architecture. Specifically, we explored the following hyper-parameters: (i) the number of neurons in the dense layer, ranging from 216 to 400 with increments of 16, using ReLU activation; and (ii) the output layer configuration, employing either one-hot encoding (with softmax activation) or binary classification (also using softmax) to determine whether a video is authentic or manipulated.

In addition to these architectural parameters, the optimization of training hyper-parameters was also considered essential. To this end, we tuned: (i) the learning rate, logarithmically sampled between 1×10^{-4} and 1×10^{-2}; (ii) the momentum parameter, sampled logarithmically between 0.87 and 0.92; and (iii) the use of the Nesterov momentum variant in the stochastic gradient descent (SGD) optimizer, as a Boolean option. This process was made using the KerasTuner library using the Bayesian optimization [O'Malley et al. 2019]. The optimization process was constrained to a maximum of 20 training rounds, with early stopping employed to terminate any round if the validation loss failed to improve for seven consecutive epochs.

Bayesian Optimization consists of two components working together: (i) a Gaussian Process model, which approximates the objective function and provides both a predicted mean and an uncertainty estimate for each point in the domain; and (ii) an acquisition function, which decides where to evaluate next by balancing exploration and exploitation. The algorithm starts by evaluating the objective function at an initial set of points and fits the Gaussian Process model to this data. Then, it uses the acquisition function to select the next point to evaluate. This process of evaluation, model update, and point selection is repeated iteratively until a stopping criterion is met [Frazier 2018]. This method effectively avoids non-promising regions of the search space, unlike random search, and is more computationally efficient than exhaustive grid search [Yang et al. 2024].

The final architecture resulting from the hyperparameter tuning process is presented in Figure 3. The model receives input tensors with dimensions $24 \times 299 \times 299 \times 3$, corresponding to sequences of 24 RGB frames, each with a spatial resolution of 299×299 pixels. The temporal dimension (24 frames) is handled by a TimeDistributed layer, which allows the Xception network to process each frame independently using shared weights. This operation yields an intermediate output of shape $24 \times 10 \times 10 \times 2048$.

Figure 3. Final Model



To aggregate these spatiotemporal features, a GlobalAveragePooling3D layer is applied, reducing the representation to a single vector of 2048 features. This vector is then passed through a fully connected dense layer with 304 neurons and ReLU activation. The final classification is performed by an output layer with two neurons and a softmax activation function, enabling the model to assign class probabilities to the categories *authentic* and *manipulated.*

During the training phase, the model checkpoint corresponding to the lowest validation loss is selected for the testing stage. This approach is adopted to mitigate overfitting and ensure that the model used for evaluation demonstrates the best generalization performance on unseen data.

3.3. Metrics

To evaluate the effectiveness of the proposed strategy, two performance metrics were employed: **Area Under the Curve (AUC)** and **Accuracy (ACC)**. The AUC metric provides insight into the model's ability to distinguish between classes by analyzing the trade-off between the true positive rate and the false positive rate across various classification thresholds.

Accuracy, on the other hand, quantifies the proportion of correct predictions made by the model, defined by:

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

where its terms are defined by: (i) **True Positives (TP):** authentic videos correctly classified as authentic; (ii) **True Negatives (TN):** manipulated videos correctly classified as manipulated; (iii) **False Positives (FP):** manipulated videos incorrectly classified as authentic; and (iv) **False Negatives (FN):** authentic videos incorrectly classified as manipulated.

4. Experiments and Results

This section presents the implementation details, experimental procedures, and the results obtained, along with a comparison to state-of-the-art approaches.

4.1. Experimental Setup

The experiments were conducted on a machine equipped with 128 GB of DDR4 RAM, an NVIDIA RTX 3090 GPU, and an Intel Core i9-10900 processor. The implementation was carried out using the Python programming language. The TensorFlow framework (version 2.15) was employed for the *Xception* model and the KerasTuner (version 1.4.7) was utilized for hyper-parameter tuning.

Model training was performed over 15 epochs. The hyper-parameters tuned for training were:

- Number of neurons in the dense layer: 304.
- Learning rate: 7.1030×10^{-4}.
- Momentum: 0.9144.
- Nesterov momentum: `False`.

For transparency and reproducibility purposes, the code used in the experiments is publicly available.[1]

4.2. Results

During the validation process, the model achieved a maximum accuracy of 97.39% and an AUC of 0.9876, as shown in Figure 4. It is worth noting that the lowest validation loss occurred at the sixth epoch, which also corresponded to the highest training accuracy. However, the highest validation accuracy was observed at epoch 13. Therefore, the model weights obtained at epoch 13 were selected for the final evaluation. Using these weights, the model achieved a final accuracy of 97.30% and an AUC of 0.9957.

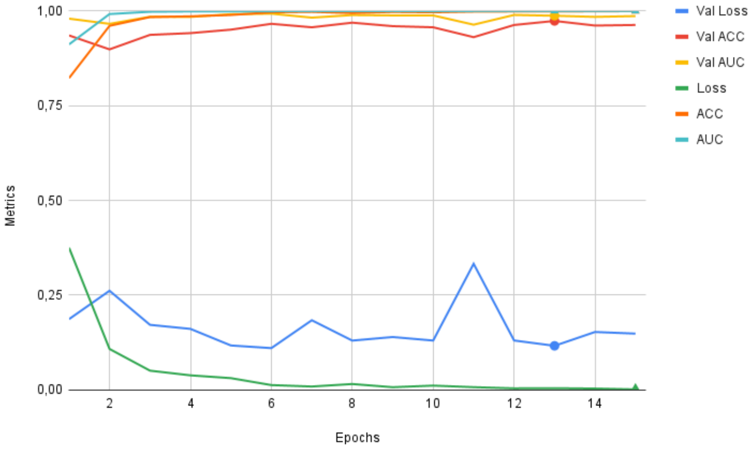


Figure 4. Obtained results over the fifteen epochs.

Finally, when comparing these results with the ones extracted from the related works that also utilize the Celeb-DF v2 dataset, as presented in Table 3, it is evident that the proposed model achieves competitive performance in terms of accuracy and AUC.

[1] Code available at `https://github.com/matheusprv/deepfake-detection-CelebDF`.

Work	Frames utilized	Accuracy	AUC
[Chen et al. 2021]	7	Not informed	0.906
[Yermakov et al. 2025]	32	Not informed	0.966
[Khormali and Yuan 2021]	Not informed	Not informed	0.987
[Heo et al. 2023]	32	Not informed	0.9930
[Reis and Ribeiro 2024]	50 + 50*9	97,50%	0.994
Proposed model	24	97,30%	0.9957
[Hernandez-Ortega et al. 2022]	210	99,47%	0.9999

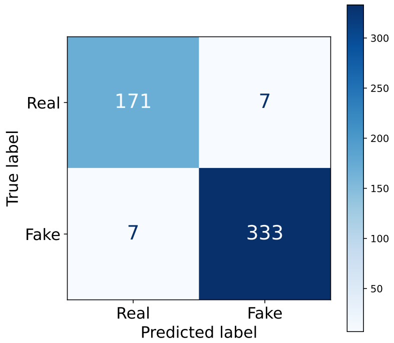
Table 3. Comparison with the literature method tested on Celeb-DF v2 ordered by AUC.

Although the proposed strategy does not achieve the highest accuracy among all related methods in Celeb-DF v2, it operates using the second smallest number of frames, that is, with less data, resulting in potentially lower computational cost while maintaining competitive accuracy and AUC.

4.3. Results Discussion

Figure 5 presents the confusion matrix resulting from the evaluation on the test dataset. The model achieved a misclassification rate of 2.70%, corresponding to 14 videos. Among these, seven authentic videos were incorrectly classified as fake, while seven fake videos were misclassified as real. Despite the significant class imbalance in the dataset, the confusion matrix demonstrates that the model was able to effectively learn the distinguishing patterns and achieve high classification performance.

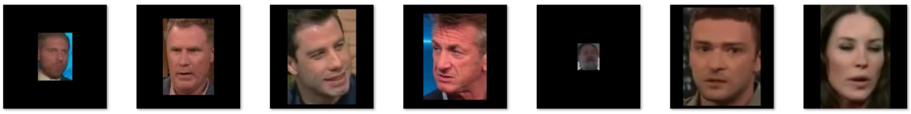
Figure 5. Test dataset confusion matrix



It is noticeable on Figure 6 (one frame of each real video classified as fake) that some videos were either resized to fit within a 299-pixel boundary for the facial region or were originally of low resolution. In the first case, a possible explanation for the misclassification is that the resizing process may have resulted in pixel aggregation, obscuring relevant features and hindering the model's ability to correctly identify the video as genuine. In the second case, where the original resolution was already low, the model may have lacked sufficient visual information to detect the necessary patterns for accurate classification, thereby leading to incorrect predictions.

Figure 7 presents a frame from each fake video that was incorrectly classified as real. In these examples, it can be observed that the images were already smaller than

Figure 6. Real classified as fake



the 299-pixel threshold, as evidenced by the presence of black margins surrounding the facial regions. This reduced resolution likely impaired the model's capacity to detect deepfake-specific artifacts, contributing to the misclassification.

Figure 7. Fake classified as real

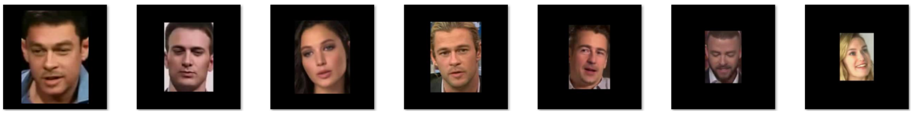
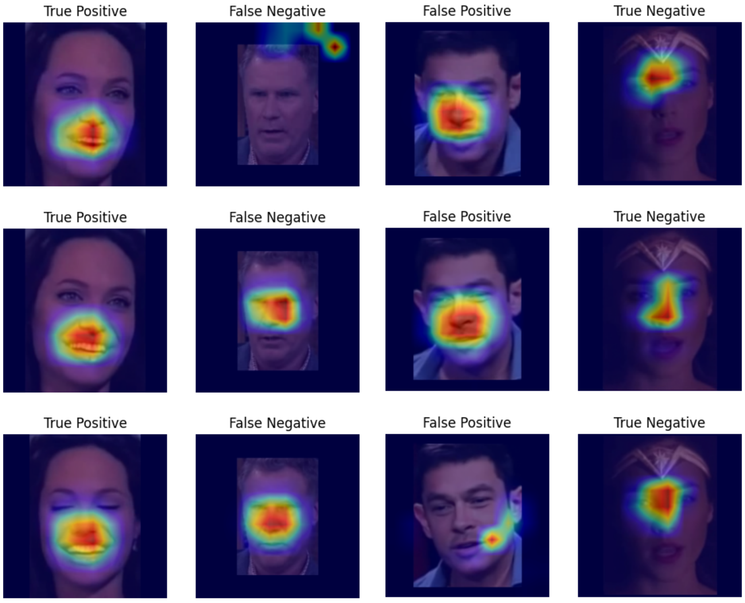


Figure 8 displays the Grad-CAM activation maps for the first frame, a middle frame (the twelfth), and the final frame of a video. These frames were selected to illustrate how the proposed model interprets and classifies different segments throughout the video. The visualizations indicate that the model primarily focuses on regions associated with noise, the mouth, and the eyes, areas that align with visual artifacts commonly perceived by humans. Moreover, the model demonstrates a dynamic attention mechanism, shifting its focus across different regions of the frames over time. This behavior suggests a frame-specific analysis strategy rather than a global one, which is consistent with the fact that manipulated regions often vary as the video progresses.

Figure 8. Activation maps of the first frame (first line), the middle frame (second line, twelfth frame) and the final frame of four videos and their classification. Regions with higher relevance are highlighted in red, while less significant areas are depicted in blue.



These results demonstrate the model's strong generalization capability in distinguishing between real and fake videos, highlighting the critical role of effective hyperparameter tuning. Although the proposed model does not surpass all state-of-the-art approaches in terms of raw performance, it was developed using a highly reproducible and transparent methodology. This reinforces the key finding that accurate deepfake detection is indeed achievable using short video segments and lighter architectures, without relying on more computationally intensive techniques like Vision Transformers. The findings validate that current, well-optimized methods can offer a compelling balance between performance, efficiency, and practicality.

5. Conclusion and further works

The proliferation of deepfake videos presents a growing challenge in digital media security, with significant implications for misinformation and fraud. Detecting such manipulated content is a complex task, particularly with limited computational resources. Traditional approaches often rely on long video sequences, which may not be feasible in real-time or resource-constrained environments. In this work, we proposed an architecture for deepfake video classification that operates using a reduced number of frames (24 per video). Despite this limitation, the proposed model achieved competitive results on the Celeb-DF-V2 dataset, reaching an accuracy of 97.30% and an AUC of 0.9957, which is comparable to results reported in related studies. These results demonstrated the potential benefits of the adopted strategy and the role of hyperparameter tuning in improving performance.

A key feature of the architecture is its ability to extract features from all frames simultaneously and integrate them through a dense layer. This approach appears to leverage the temporal consistency present in the sequence of frames, contributing to its overall performance. As directions for future work, it may be useful to explore the use of time-distributed layers applied to uncropped video frames, rather than focusing exclusively on facial regions. Additionally, the incorporation of Vision Transformers could be investigated as a means to further enhance classification performance.

Acknowledgments

The authors would like to thank the *Fundação de Amparo à Pesquisa do Estado de Minas Gerais* (FAPEMIG, grants APQ-01768-24), the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq) and the *Universidade Federal de Ouro Preto* (UFOP/PROPPI) for supporting the development of this study.

References

Alanazi, S. and Asif, S. (2024). Exploring deepfake technology: creation, consequences and countermeasures. Human-Intelligent Systems Integration, 6(1):49–60.

Chen, H. and Magramo, K. (2024). Finance worker pays out $25 million after video call with deepfake chief financial officer.

Chen, H.-S., Rouhsedaghat, M., Ghani, H., Hu, S., You, S., and Kuo, C. C. J. (2021). Defakehop: A light-weight high-performance deepfake detector.

de Lima, O., Franklin, S., Basu, S., Karwoski, B., and George, A. (2020). Deepfake detection using spatiotemporal convolutional networks.

Frazier, P. I. (2018). A tutorial on bayesian optimization.

Heidari, A., Jafari Navimipour, N., Dag, H., and Unal, M. (2024). Deepfake detection using deep learning methods: A systematic and comprehensive review. WIREs Data Mining and Knowledge Discovery, 14(2):e1520.

Heo, Y.-J., Yeo, W.-H., and Kim, B.-G. (2023). Deepfake detection algorithm based on improved vision transformer. Applied Intelligence, 53(7):7512–7527.

Hernandez-Ortega, J., Tolosana, R., Fierrez, J., and Morales, A. (2022). DeepFakes Detection Based on Heart Rate Estimation: Single- and Multi-frame, pages 255–273. Springer International Publishing, Cham.

Khormali, A. and Yuan, J.-S. (2021). Add: Attention-based deepfake detection approach. Big Data and Cognitive Computing, 5(4).

Kirby, P. (2023). Muharrem ince: Turkish candidate dramatically pulls out before election.

Le, T.-N., Nguyen, H. H., Yamagishi, J., and Echizen, I. (2021). Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild.

Li, Y., Yang, X., Sun, P., Qi, H., and Lyu, S. (2020). Celeb-df: A large-scale challenging dataset for deepfake forensics.

Masud, U., Sadiq, M., Masood, S., Ahmad, M., and El-Latif, A. A. A. (2023). Lw-deepfakenet: a lightweight time distributed cnn-lstm network for real-time deepfake video detection. Signal, Image and Video Processing, 17(8):4029–4037.

Mirsky, Y. and Lee, W. (2021). The creation and detection of deepfakes: A survey. ACM Computing Surveys, 54(1):1–41.

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). Kerastuner. `https://github.com/keras-team/keras-tuner`.

Reis, P. M. G. I. and Ribeiro, R. O. (2024). A forensic evaluation method for deepfake detection using dcnn-based facial similarity scores. Forensic Science International, 358:111747.

Serengil, S. I. and Ozpinar, A. (2020). Lightface: A hybrid deep face recognition framework. In 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), pages 23–27. IEEE.

Singh, A., Saimbhi, A. S., Singh, N., and Mittal, M. (2020). Deepfake video detection: A time-distributed approach. SN Computer Science, 1(4):212.

Wilcox, L. M., Allison, R. S., Helliker, J., Dunk, B., and Anthony, R. C. (2015). Evidence that viewers prefer higher frame-rate film. 12(4).

Yang, K., Liu, L., and Wen, Y. (2024). The impact of bayesian optimization on feature selection. Scientific Reports, 14(1):3948.

Yermakov, A., Cech, J., and Matas, J. (2025). Unlocking the hidden potential of clip in generalizable deepfake detection.

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2023). Dive into Deep Learning. Cambridge University Press. `https://D2L.ai`.

Łabuz, M. and Nehring, C. (2024). On the way to deep fake democracy? deep fakes in election campaigns in 2023. European Political Science.