Combining Computer Vision Techniques and Intraframe Noise Methods to Detect a Deepfake

Maya P Shelke, Nihar Ranjan, Ajinkya Kharade, Pranav Gaikwad, Shubham Arakh, Analp Kalore

Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Corresponding author: Maya P Shelke, Email: mayabembde04@gmail.com

Deep fakes are synthetic videos created by deep learning algorithms that can convincingly depict individuals saying or doing things they never did. With the proliferation of deepfake videos in social media and the potential for them to cause significant harm, deep fake detection has become a pressing issue. In this study, we offer a unique method for detecting deepfakes by combining computer vision methods with intraframe noise. The proposed approach involves extracting features from the video frames, including texture, color, and edges, and then adding a layer of intraframe noise to the video frames. We tested the proposed method on a variety of benchmark datasets and found that it achieved high accuracy in detecting deepfake videos.

Keywords: Deep Fake, intraframe noise, extracting features, high accuracy

2023. In Satyasai Jagannath Nanda & Rajendra Prasad Yadav (eds.), *Data Science and Intelligent Computing Techniques*, 471–480. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-43

1. Introduction

Deep Fake is a technology that can superimpose the face of a person with another person's face, alter facial expressions or gestures, appearing as in the original video using artificial intelligence concepts. It manipulates the whole activity of the targeted person which synthesizes audio and visual aspects. Deepfakes are often created with the intention to deceive or mislead viewers into believing that the manipulated content is genuine. Some social media users posted a doctored video to promote the myth that After meeting with President Donald Trump, House Speaker Nancy Pelosi fumbled over her speech [1]. The deep fake technology has created and developed videos that cannot be identified by the normal human being eyes [2]. It can detect the real and fake images by using advanced network architecture.To identify the forged videos and images created by using deep learning methods can be achieved [3]. The fake images can be present in three forms [4]. Any AI-generated mimicking videos are now referred to as "Deep Fake," which is a more general term. Basically, there are three major types of Deep Fake videos. Making a movie of a person's complete head and upper shoulders is what head puppetry entails. Face exchanging entails exchanging the face of the targeted person in a video with the source footage while maintaining the same facial expressions. Lip Syncing isolates the lip region and syncs it with audio that looks to pronounce something that the individual does not actually say. [5]. In order to create Deep Fake the Deep neural networks are used. Various recent technological advancements with deep learning techniques including auto encoders and GAN [6][7](Generative Adversarial Networks) are used to create fake faces which are applied mainly in the computer vision. Deep Fake detection method using the Haar wavelet transforms. The method aims to distinguish between original videos and Deep Fake videos by exploiting the limitations and artifacts introduced during the Deep Fake generation process.

The suggested method takes advantage of the fact that Deep false algorithms have specific constraints, such as generating false faces of a specific size and resolution. An affine transformation and blur function are applied to the synthesised faces to match the source face to the target in the original video. As shown in the given figure 1, we can see the difference between original and deepfake faces.

Figure: 1. Original vs Deep Fake [8]

The detection method utilizes the Haar wavelet transform to analyze the blur inconsistency and detect Deep Fake forgery. Using the Haar wavelet transform function, it compares the blurred synthesised ROI with the surrounding environment. By analyzing edge types and sharpness, the method determines whether a face image has been blurred and to what extent [9]. To combat the growing number of deepfakes on the internet, major tech companies are actively investigating methods to detect deepfakes [10]. Also, Google has released a free dataset for the public as a contribution to the deepfake detection. The Deepfake Detection Challenge initiated by major tech companies and the release of a free dataset by Google is mentioned as examples of industry efforts in addressing the deepfake issue [4], combines deep learning techniques like recurrent neural networks (RNN), convolutional neural networks (CNN), and long short-term memory to provide a thorough research for deepfake identification. [9] [11] [12].

2. Literature Survey

G. Lee and M. Kim. [2] proposed a technique for Using a DNN to calculate the rate of change of computer vision features based on the difference between a given amount of frames and frames.

The method is the first real-time facial reproduction system that only needs monocular RGB input, according to J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner [3].

According to S. Lyu's article[13] Predict that a number of upcoming technology advancements will enhance the phoney videos' production effectiveness and visual quality.

Mohammed Akram Younus and Taha Mohammed Hasan. [10] DeepFakes, a novel method for spotting false faces created artificially, have been proposed. Given that DeepFake can only generate face photos in set sizes and low resolutions, which must then be blurred and modified to match the faces that will be replaced in the original film, In the generated DeepFake movies, These additive blur and ROI changes produce distinct artefacts that may be efficiently recorded by spotting discrepancies.

Dolhansky, Brian and Howes, Russ and Pflaum, Ben and Baram, Nicole and Ferrer, Cristian [4] introduced a preview of the DFDC dataset that will be made available later this year with the goal of encouraging researchers to familiarise themselves with the data, providing preliminary findings, and comparing those findings to suggested baselines.

R. Saravana Ram, M. Vinoth Kumar, Tareq M. Al-shami, Mehedi Masud, Hanan Aljuaid and Mohamed Abouhawwash in [5] suggested extracting features from the input deepfake image using fuzzy clustering.

Kandasamy V, Hubálovsk, and Trojovsk [14] revealed the Deep learning approach with two levels for detecting deepfake photos and videos. To extract features from face images, the recommended SAE technique is employed.

Abdulqader M. Almars [9] gave a thorough explanation of the architecture, tools, and performance of the existing deepfake approaches. It also emphasised the publicly available datasets used by the scientific community and sorted them by source, technique, and dataset.

The proposed technique by Aarti Karandikar, Vedita Deshpande, Sanjana Singh, Sayali Nagbhidkar, and Saurabh Agrawal [11] focuses on facial modification for forgery detection and leverages transfer learning on the VGG-16 model to train the dataset.

As correctly mentioned by authors in the papers [15,16,17] the fisherface Linear binary pattern histogram using the DBN classifier (FF-LBPH DBN) technique was implemented as a detection technique for deepfake images. The planned work was carried out much quickly, and it was quite good at distinguishing between phoney and real images.

Author has seen, [12] An very accurate bidirectional recurrent neural network model may identify bogus news.

3. Proposed System

Deepfake detection is used in various real-life applications such as social media platforms, news verification, law enforcement, entertainment industry, political campaigns, online identity verification, and online reputation management. It helps identify and flag manipulated content, verify the authenticity of videos and images, analyze evidence in legal proceedings, enforce copyright and protect intellectual property, combat political misinformation, verify online identities, and manage online reputation. Deepfake detection is a dynamic field that requires continuous research and updates to stay ahead of evolving deepfake techniques.

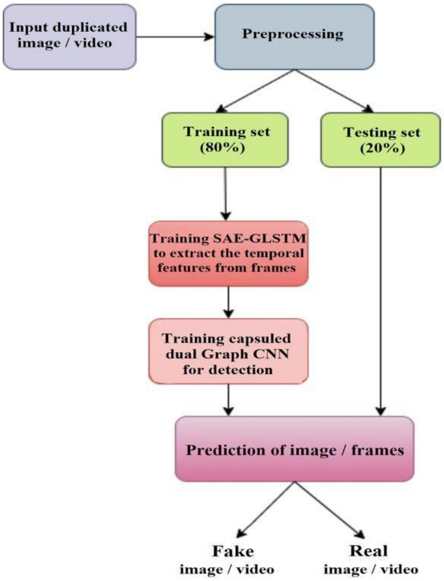
Input duplicated image / video
Preprocessing
Training set (80%)
Testing set (20%)
Training SAE-GLSTM to extract the temporal features from frames
Training capsuled dual Graph CNN for detection
Prediction of image / frames
Fake image / video
Real image / video

Figure: 2. Preprocessing System for Deepfake[5]

Deepfake detection can consist of several components and techniques working together. By looking at the given figure 2, A high-level description of a hypothetical system is given below:

Preprocessing: The system starts by preprocessing the input data, which could be a video, image, or audio file. This step may involve resizing, normalizing, or filtering the data to ensure consistency and compatibility with the subsequent analysis steps.

1. Feature Extraction: In this stage, relevant features are extracted from the input data. These features can include visual cues, audio characteristics, and metadata. Deep neural networks and other advanced approaches or convolutional neural networks (CNNs) can be employed to extract discriminative features.
2. Training and Model Development: A machine learning model is trained on a large and diverse dataset that contains both real and deepfake samples. The model learns to differentiate between genuine and manipulated media by analyzing the extracted features. Various techniques like Depending on the situation, supervised learning, unsupervised learning, or semi-supervised learning can be used, the availability of labeled data.
3. Detection Algorithm: The trained model is then used as the core detection algorithm. It takes the extracted features of an unknown sample as input and calculates a confidence score or

probability indicating the likelihood of it being a deepfake. The algorithm should be designed to detect both known and unknown types of deepfakes.

4. Multimodal Analysis: To enhance the accuracy and reliability of the detection system, multimodal analysis can be employed. This involves analyzing multiple modalities such as video, audio, and metadata simultaneously. Inconsistencies or discrepancies across different modalities can indicate the presence of a deepfake.
5. Post-processing and Fusion: The outputs from different detection algorithms or modalities can be fused or combined to generate a final decision. Fusion techniques can include weighted averaging, ensemble methods, or decision-level fusion. This step helps improve the overall accuracy and reliability of the system.
6. Real-time Implementation: To detect deepfakes in real-time scenarios, the system should be optimized for efficiency and speed. Techniques like parallel computing, hardware acceleration (e.g., GPUs), or model compression can be utilized to achieve real- time performance.
7. Continuous Monitoring and Updates: Deepfake techniques are evolving rapidly, so the system should be regularly updated with new training data and models. Continuous monito to stay effective in detecting the latest threats.
8. User Interface and Reporting: The system can provide a user-friendly interface for users to interact with and analyze the results. It can generate detailed reports highlighting the detected deepfake indicators, confidence scores, and any additional information that aids in further investigation.
9. ring and research on emerging deepfake techniques will enable the system to stay effective in detecting the latest threats.
10. User Interface and Reporting: The system can provide a user-friendly interface for users to interact with and analyze the results. It can generate detailed reports highlighting the detected deepfake indicators, confidence scores, and any additional information that aids in further investigation.

4. Implementation Details

Implementing deepfake detection using computer vision with improved quality and speed involves several key steps and considerations. Consider the following implementation details:

Dataset Collection and Preparation: Gather a diverse and high-quality dataset consisting of both real and deepfake samples. Ensure that the dataset covers various deepfake generation techniques, resolutions, lighting conditions, and camera angles. Preprocess the data by resizing, normalizing, and augmenting the images or videos as needed.

1. Model Selection and Training: Choose a suitable deep learning architecture for deepfake detection, such as a CNN or a combination of CNN and RNNs for temporal analysis. Consider using pre-trained models like ResNet, Inception, or EfficientNet and fine-tune them on the deepfake detection task. Employ transfer learning to leverage features learned from large-scale datasets like ImageNet.
2. Feature Extraction: Extract meaningful features from the input data using computer vision techniques. This may involve face detection, facial landmark detection, texture analysis, or motion analysis. Select features that capture relevant information for differentiating between real and deepfake content.
3. Ensemble Methods: Train multiple deepfake detection models with different architectures or trained on different subsets of the dataset. Combine their predictions through ensemble methods like majority voting or averaging to improve detection accuracy. Experiment with different ensemble techniques to find the optimal combination.
4. Hardware Acceleration: Utilize GPUs or other hardware accelerators to accelerate the execution of deep learning models. This can be done using deep learning frameworks like

TensorFlow or PyTorch, which provide GPU support. Ensure that the hardware infrastructure is properly set up and configured for efficient parallel processing.

5. Model Optimization: Apply model optimization methods for reducing computational load and memory requirements. This may include model quantization to reduce the precision of model weights and activations, or model compression techniques to reduce the number of parameters. Utilize optimized libraries or frameworks like TensorFlow Lite or TensorRT to improve the speed of execution.
6. Real-time Implementation: Optimize the system for real-time performance by minimizing latency. Use efficient data structures and algorithms to optimize the processing of intermediate results. Employ techniques like batch processing, multi-threading, or asynchronous processing to parallelize computations and reduce latency.
7. Continuous Improvement: Stay updated with the latest research advancements in deepfake generation and detection. Continuously evaluate and refine the deepfake detection system by incorporating new techniques, datasets, and model updates. Regularly monitor performance metrics and incorporate feedback to enhance the quality and speed of detection.
8. Evaluation and Testing: Evaluate the performance of the deepfake detection system using appropriate parameters including F1 score, recall, accuracy, and precision. Conduct extensive testing on diverse datasets, including unseen deepfake variations, to assess the robustness and generalization capability of the system.
9. Deployment and Integration: Integrate the deepfake detection system into the desired application or platform. Ensure that the system is scalable, reliable, and user-friendly. Consider factors like system requirements, user interfaces, and deployment options (cloud-based, on-premise, or edge computing) based on the specific use case and deployment environment.

It's important to note that the implementation details can vary depending on the specific deepfake detection approach, available resources, and the chosen technologies. Experimentation, fine-tuning, and adapting the implementation to suit the specific requirements and constraints of the system are necessary to achieve optimal results.

A mathematical expression representing the process of deepfake detection using computer vision and algorithms:

Let:

X be the input data (preprocessed images/videos).

Y be the ground truth labels (0 for real content, 1 for deepfake content).

(x) be the feature extraction function that maps the input data to a set of features.

θ be the parameters of the detection model.

The deepfake detection model can be represented as a function $h(X;\theta)$ that takes the preprocessed data X and the model parameters θ as inputs and outputs the probability of the input being a deepfake, as mentioned in eq 1.

$$h(X;\theta) = P(Y = 1|X;\theta) \qquad \text{Eq(1)}$$

To train the model, we use a labeled dataset as depicted in eq 2.

$$D = \{\,(X1, Y1), (X2, Y2)\ \dots \dots .\, (Xn, Yn)\} \qquad \text{Eq(2)}$$

The purpose is to discover the best parameters θ * for minimising classification error. This is accomplished by minimising the loss function. $(\theta)$with respect to θ as mentioned in eq 3:

$\theta * = argmin\theta\ (\theta)$ Eq(3)

The choice of the loss function depends on the specific approach and can vary. Commonly used loss functions for binary classification include cross-entropy loss or binary logistic loss.

Once the model is trained, it can be used to classify new, unseen data by calculating the probability of being a deepfake using the learned parameters which is depicted in eq 4:

$(Y = 1|X; \theta *)$ Eq(4)

The threshold for classifying an input as a deepfake can be adjusted based on the desired balance between false positives and false negatives. Applying intraframe noise for deepfake detection involves extracting the noise patterns within individual frames and utilizing them to identify potential manipulations.

To apply intraframe noise method to detect a deepfake :-

Here's a step-by-step explanation of how to use intraframe noise for this purpose:

1. Frame Extraction and Preprocessing:
   - Extract frames from both authentic and suspect videos.
   - Preprocess frames by resizing them to a consistent resolution and converting them to grayscale. Grayscale simplifies the analysis while retaining essential information for noise detection.

2. Noise Extraction:
   - Choose a denoising technique (e.g., Gaussian blurring, median filtering) to create an estimate of the noise-free frame.
   - Subtract the denoised frame from the original frame to obtain the intraframe noise. This step highlights the noise patterns within the frame.

3. Feature Extraction:
   - Calculate statistical features from the intraframe noise. Some features to consider:
   - Mean: The average value of noise in the frame.
   - Standard Deviation: Measures the spread of noise values around the mean.
   - Variance: Indicates the amount of noise variation in the frame.
   - Skewness: Describes the asymmetry of the noise distribution.
   - Kurtosis: Measures the tail behavior of the noise distribution.
   - Entropy: Represents the randomness of noise values.
   - Higher-Order Moments: Higher-order statistical moments can capture finer noise characteristics.

4. Training a Classifier:
   - Prepare a labeled dataset containing features extracted from both authentic and deepfake frames.
   - Train a binary classifier (e.g., logistic regression, random forest, support vector machine, or a deep learning model) using the noise features as inputs and the label (real or fake) as the output in Eq(2)

5. Validation and Hyperparameter Tuning:
   - Split the dataset into training and validation sets to tune the hyperparameters of the classifier.
   - Use techniques like cross-validation to ensure robust performance estimation.

6. Evaluation and Testing:
   - Test the trained classifier on a separate testing dataset to evaluate its performance.
   - Calculate metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to assess the model's effectiveness. Eq(3)

7. Adversarial Considerations:
   - Deepfake creators might attempt to manipulate noise patterns to evade detection.
   - Incorporate techniques from adversarial machine learning to enhance your model's resistance to such manipulations.

8. Threshold Selection:
   - Determine an appropriate threshold for the classifier's output probability or score to decide whether a frame is real or fake.
   - The threshold impacts the trade-off between false positives and false negatives.

9. Deployment and Monitoring:
   - Integrate the trained model into a deepfake detection system.
   - Continuously monitor the model's performance and update it as new deepfake techniques emerge.

It's important to note that while intraframe noise analysis can be effective, it's most effective when used in conjunction with other deepfake detection techniques. Combining multiple approaches increases the overall reliability and accuracy of the detection system. Additionally, the success of this method depends on the quality of the noise estimation and the chosen statistical features, so experimentation and refinement are essential for optimal results.

It is important to note that the specific implementation and mathematical expressions may vary depending on the chosen algorithms and techniques used for feature extraction and classification. The above representation provides a general framework for understanding the process of deepfake detection using computer vision and algorithms.

5. Result

The result of the deepfake detection system can be a binary classification indicating whether the input media (image or video) is classified as a deepfake or genuine. This result is based on the analysis and prediction made by the trained deep learning model or ensemble of models.

The system may provide a confidence score or probability along with the classification result, indicating the level of certainty in the prediction. A higher confidence score suggests a higher likelihood of the input being classified correctly.

Additionally, the system may generate a detailed report highlighting the detected deepfake indicators or features that contributed to the classification. This information can be useful for further investigation or analysis.

The overall result of the system can be evaluated using accuracy, precision, recall, and F1 score are all performance indicators. These metrics assess the efficiency and reliability of the deepfake detection system in correctly identifying deepfakes while minimizing false positives or false negatives.

It is important to note that the accuracy and performance of the system can vary depending on the training dataset's quality and variety, the chosen detection techniques, and the implementation details. Regular monitoring, evaluation, and continuous improvement are crucial to ensure the system's effectiveness in detecting evolving deepfake techniques.

6. Conclusion

Choosing a suitable deep learning architecture, such as CNN or a hybrid of CNN and RNN, is important for deepfake detection. Leveraging pre-trained models like ResNet, Inception, or EfficientNet through transfer learning can provide a good starting point. Feature extraction techniques, including face detection, facial landmark detection, texture analysis, and motion analysis, help capture relevant information for differentiation Ensemble methods, such as combining predictions from multiple models through majority voting or averaging, can enhance detection accuracy. Leveraging GPUs or other hardware accelerators, along with model optimization techniques like quantization and compression, improves the speed of execution. Real-time implementation can be achieved by minimizing latency and utilizing efficient algorithms.

Continuously updating the system with the latest research advancements, evaluating performance metrics, and incorporating feedback are crucial for continuous improvement. Evaluating the system using appropriate metrics and extensive testing on diverse datasets ensures robustness and generalization capability.

Integrating the deepfake detection system into the desired application or platform requires scalability, reliability, and user-friendliness. Consider system requirements, user interfaces, and deployment options based on the specific use case and environment.

Remember that the implementation details provided here serve as a general guide. Adjustments and experimentation may be necessary to optimize the deepfake detection system based on your specific needs and available resources. Stay informed about the latest developments in deepfake generation and detection to ensure your system remains effective against evolving deepfake techniques.

7. Future Scope

The future scope for deepfake detection involves the development of advanced algorithms and techniques to detect increasingly sophisticated deepfake videos, images, and audio. This includes improving detection algorithms through machine learning and AI, incorporating multimodal analysis across different sources, creating benchmark datasets for evaluation, focusing on explainable AI to provide transparency, promoting collaboration and standards, working on real-time detection systems, countering anti- forensic techniques, raising awareness through education, and staying ahead of deepfake technology advancements.

References

[1] Donie O'Sullivan, CNN Updated 12:31 PM EDT, Fri May 24, 2019

[2] Lee, G. and Kim, M., 2021. Deepfake detection using the rate of change between frames based on computer vision. Sensors, 21(21), p.4364.

[3] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C. and Nießner, M., 2016. Face2face: Real-time face capture and reenactment of rgb videos. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2384-2395).

[4] Dolhansky, B., Howes, R., Pflaum, B., Baram, N. and Ferrer, C.C., 2019. The deepfake detection challenge (dfdc) preview dataset. arXiv preprint arXiv:1910.08854.

[5] Saravana Ram, R., Vinoth Kumar, M., Al-shami, T.M., Masud, M., Aljuaid, H. and Abouhawwash, M., 2023. Deep Fake Detection Using Computer Vision-Based Deep Neural Network with Pairwise Learning. Intelligent Automation & Soft Computing, 35(2).

[6] Balasubramanian, S.B., Prabu, P., Venkatachalam, K. and Trojovský, P., 2022. Deep fake detection using cascaded deep sparse auto-encoder for effective feature selection. PeerJ Computer Science, 8, p.e1040.

[7] Shad, H.S., Rizvee, M., Roza, N.T., Hoq, S.M., Monirujjaman Khan, M., Singh, A., Zaguia, A. and Bourouis, S., 2021. Comparative analysis of deepfake Image detection method using convolutional neural network. Computational Intelligence and Neuroscience, 2021.

[8] Video personalization using deepfake Technology, Aug 5, 2023

[9] Almars, A.M., 2021. Deepfakes detection techniques using deep learning: a survey. Journal of Computer and Communications, 9(5), pp.20-35.

[10] Khalil, H.A. and Maged, S.A., 2021, May. Deepfakes creation and detection using deep learning. In 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 1-4). IEEE.

[11] Karandikar, A., Deshpande, V., Singh, S., Nagbhidkar, S. and Agrawal, S., 2020. Deepfake video detection using convolutional neural network. International Journal of Advanced Trends in Computer Science and Engineering, 9(2), pp.1311-1315.

[12] Jiang, T., Li, J.P., Haq, A.U. and Saboor, A., 2020, December. Fake news detection using deep recurrent neural networks. In 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP) (pp. 205-208). IEEE.

[13] Lyu, S., 2020, July. Deepfake detection: Current challenges and next steps. In 2020 IEEE international conference on multimedia & expo workshops (ICMEW) (pp. 1-6). IEEE.

[14] Kandasamy, V., Hubálovský, Š. and Trojovský, P., 2022. Deep fake detection using a sparse auto encoder with a graph capsule dual graph CNN. PeerJ Computer Science, 8, p.e953.

[15] Suganthi, S.T., Ayoobkhan, M.U.A., Bacanin, N., Venkatachalam, K., Štěpán, H. and Pavel, T., 2022. Deep learning model for deep fake face recognition and detection. PeerJ Computer Science, 8, p.e881.

[16] Younus, M.A. and Hasan, T.M., 2020, April. DeepFake Detection Method Based on Haar Wavelet Transform Effective and Fast. In 2020 International Conference on Computer Science and Software Engineering (CSASE) (pp. 186-190). IEEE.

[17] Rafique, R., Gantassi, R., Amin, R., Frnda, J., Mustapha, A. and Alshehri, A.H., 2023. Deep fake detection and classification using error-level analysis and deep learning. Scientific Reports, 13(1), p.7422.