Journal of Big Data

RESEARCH Open Access

A novel approach for detecting deep fake videos using graph neural network

M. M. El-Gayar[1,6*], Mohamed Abouhawwash[2,3], S. S. Askar[4] and Sara Sweidan[5,6]

*Correspondence:
mostafa_elgayar@mans.edu.eg

[1] Department of Information Technology, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt
[2] Department of Computational Mathematics, Science and Engineering (CMSE), College of Engineering, Michigan State University, East Lansing, MI 48824, USA
[3] Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt
[4] Department of Statistics and Operations Research, College of Science, King Saud University, P.O. Box 2455, 11451 Riyadh, Saudi Arabia
[5] Artificial Intelligence Department, Faculty of Computer and Artificial Intelligence, Benha University, Banha, Egypt
[6] Faculty of Computer Science and Engineering, New Mansoura University, Gamasa, Egypt

Abstract

Deep fake technology has emerged as a double-edged sword in the digital world. While it holds potential for legitimate uses, it can also be exploited to manipulate video content, causing severe social and security concerns. The research gap lies in the fact that traditional deep fake detection methods, such as visual quality analysis or inconsistency detection, need help to keep up with the rapidly advancing technology used to create deep fakes. That means there's a need for more sophisticated detection techniques. This paper introduces an enhanced approach for detecting deep fake videos using graph neural network (GNN). The proposed method splits the detection process into two phases: a mini-batch graph convolution network stream four-block CNN stream comprising Convolution, Batch Normalization, and Activation function. The final step is a flattening operation, which is essential for connecting the convolutional layers to the dense layer. The fusion of these two phases is performed using three different fusion networks: FuNet-A (additive fusion), FuNet-M (element-wise multiplicative fusion), and FuNet-C (concatenation fusion). The paper further evaluates the proposed model on different datasets, where it achieved an impressive training and validation accuracy of 99.3% after 30 epochs.

Keywords: Graph neural network, Convolutional neural network, Deepfake video detection, Multi-task cascaded convolutional neural network, Mini-GNN

Introduction

The rise of deep fake technology has opened a new frontier in the digital world, enabling the creation of convincing synthetic video content. While this advancement offers potential for positive applications, it poses significant risks to information security and integrity. "Deep fake videos"—artificially synthesized video content manipulated using deep learning methodologies—pose significant threats to information integrity and security. The ability to manage videos can lead to misinformation, identity theft, and other forms of cybercrime. This has spurred a growing need for effective detection techniques to counteract the misuse of deep fake technology. Detecting deep fakes is complex due to the sophistication of contemporary AI-driven synthesis techniques. Traditional detection methods, such as those based on visual quality or inconsistencies, become less effective as deep fake technology evolves [1–3].

Deepfakes, artificial intelligence-based synthetic media where individuals in existing images or videos are replaced with someone else's likeness, have become increasingly prevalent. The rapid advancement in deepfake technologies has made it increasingly challenging to distinguish between natural and manipulated media, posing significant threats to information credibility, privacy, and security. While various deepfake detection methods have been proposed, many need help with overfitting issues, high computational complexity, and lack of generalizability across different datasets and deepfake techniques. Additionally, most current methods focus primarily on video and image-based deepfakes and overlook the potential use of other forms of media, like audio and text. Given these challenges, there is an urgent need for a robust, efficient, and comprehensive deepfake detection method that can effectively handle various media types and deepfake techniques. Furthermore, as deepfake technologies continue to evolve, it is crucial that our detection methods also adapt. Deep fakes can now convincingly mimic facial expressions, lip movements, and even voices, making them virtually indistinguishable from real videos. Furthermore, the wide variety of deep fake generation methods and their continual improvement make developing a universally applicable detection algorithm challenging. There is also the challenge of dataset imbalance, as the quantity of genuine videos vastly outweighs that of deep fakes, leading to biased detection results. Traditional Convolution Neural Networks (CNNs), while powerful image and video analysis tools, have certain limitations when applied to deep fake detection. One such limitation is their limited temporal context. CNNs analyze each video frame independently, not considering the temporal correlations between different frames. This means they might miss out on material inconsistencies in deepfakes that could be detected by considering multiple frames in context. Additionally, CNNs are susceptible to adversarial attacks. These attacks introduce subtle perturbations into an image or video designed to fool CNN and cause it to misclassify the content. This vulnerability can be exploited to create undetected deepfakes that pass through CNN-based detection systems. Moreover, CNNs are prone to overfitting, especially when trained on limited data. This could lead to poor generalization, causing the CNN to fail when encountering new or different types of deepfakes. The issue of overfitting could be more problematic in the context of deepfake detection due to the relative scarcity of deepfake videos for training purposes.

In this paper, we address these challenges by introducing a novel approach combining GNN and CNNs' strengths. Our proposed model exploits GNN's ability to capture spatial–temporal information and CNN's capability to extract visual features from each frame. We further enhance the model's robustness to adversarial attacks and prevent overfitting by employing three different fusion strategies and a mini-batch technique. This paper presents a novel, efficient model for these challenges, utilizing minibatch GNNs (miniGNNs). Comparable to CNNs, miniGNNs can efficiently train the network for deep fake video detection on a downsampled graph (or topological structure) in a minibatch manner. Additionally, the model trained can be employed directly to predict new data. Through our newly introduced miniGNNs, we aim to conduct a detailed comparison between CNNs and GNNs (both qualitatively and quantitatively). CNNs and GNNs are recognized for their ability to extract and symbolize information from deep fake videos, albeit from different vantage points—for instance, spatial–temporal features of CNNs, graph (or relational) representations of GNNs, etc. This naturally motivates us

to use them jointly, exploring various fusion strategies and enhancing their applicability for deep fake video detection [4–6]. More specifically, the main contributions of this paper are threefold:

- We systematically analyze CNNs and GNNs, focusing on deep fake video detection. To the best of our knowledge, this is the first instance where the potential and drawbacks of GNNs have been scrutinized within the research community, especially compared with CNNs. Unlike many other types of neural networks, GNNs are naturally invariant to the order of input data points.
- We propose a novel supervised version of GNNs, which we call miniGNNs. As the name suggests, miniGNNs can be trained in a minibatch manner, striving to find a more robust and superior local optimum. Unlike traditional GNNs, our miniGNNs can train the networks using a training set and facilitate a straightforward inference of large-scale, out-of-sample data using the trained model.
- We develop three fusion strategies, including additive fusion (FuNet-A), element-wise multiplicative fusion (FuNet-M), and concatenation fusion (FuNet-C), to achieve better deep fake video detection results by integrating features extracted from CNNs and our miniGNNs, in an end-to-end trainable network.

This manuscript introduces a novel model to detect deep fake videos that address these challenges. We propose a model that combines the strengths of GNN and CNN to enhance detection accuracy. GNN allows us to exploit the spatial–temporal information of the video content, which is often overlooked by other deep fake detection methods. On the other hand, CNN enables us to extract visual features from each frame effectively. We further enhance the model by integrating the two phases using three different fusion networks, allowing the model to handle a wider range of deep fake techniques. We also address the dataset imbalance problem by applying a mini-batch technique, ensuring a balanced sample of genuine and counterfeit videos in each batch. Our proposed model was evaluated on different datasets and achieved an impressive training and validation accuracy of 99.3% in just 30 epochs, demonstrating its effectiveness in detecting deep fake videos. This paper comprehensively describes our model, discusses the fusion strategies, and presents the evaluation results, contributing a new perspective to the ongoing discourse on deep fake detection.

The remainder of this article is organized as follows. Section Literature review provides an in-depth examination of existing literature related to this topic. The proposed model is meticulously explained in section Methodology and proposed model. A comprehensive set of experiments and their corresponding analyses form the crux of section Configuration of experimental results. The article reaches its culmination in section Conclusion, where we offer concluding thoughts and allude to potential avenues for future research.

Literature review

Over the past few years, there has been a concerted effort by various authors to develop methods for identifying DeepFakes. Early studies focused on identifying visual inconsistencies within individual frames, with some methods utilizing biological signals and

others delegating feature extraction to CNNs. In one promising approach, authors proposed using capsule networks with dynamic routing to achieve highly accurate results. Other successful methods have focused on localizing altered regions, particularly the face. Given the potential for DeepFakes to spread false information, there is a growing concern about the need to detect them accurately. To this end, some authors have suggested tracking facial landmarks to learn behaviors typical of specific individuals, which can then be used to distinguish between authentic and fake video content [7–9].

Although CNNs have recently advanced, they have also automatically made it easier to generate fake visual images, commonly called DeepFakes. There are numerous similar techniques, including the face-swap approach employed on social media applications like Snapchat, which is a quick but subpar method. However, when faces are added to learned frames from source videos or collections of images, Generative Adversarial Networks (GANs) provide more desirable output, and both the FakeApp and Faceswap Github host methods for public use. In Face2Face, facial expressions are reenacted from source to target frames by an algorithm, allowing for merging videos with fabricated sound data to create entirely fake material. Recent studies demonstrate that algorithms can also produce speech that convincingly resembles a target speaker based on their text or words [10–14].

Despite achieving high accuracy, modern techniques for DeepFake detection still need to provide a comprehensive solution that can withstand various video modifications, particularly regarding voice biometrics. Consequently, they must be more reliable for detecting DeepFakes in the field.

Montserrat et al. [14] introduced a straightforward yet potent tactic that leverages the combined strength of convolutional neural networks (CNN), recurrent neural networks (RNN), and the DFDC dataset to attain the most impressive results. The system operates efficiently on a single GPU and swiftly processes video in less than eight seconds. Even though its main focus is on identifying modifications to facial features, it does not scrutinize the accompanying audio content, a potential area for enhancement that could significantly boost detection accuracy in future research. The ultimate objective of this study is to equip journalists, both locally and globally, with the means to uncover DeepFake videos. Despite its efficiency and speed, it exclusively focuses on identifying facial modifications, overlooking the potential use of accompanying audio content for detection. This leaves open the question of how integrating audio analysis could enhance detection accuracy.

Gu et al. [15] provided a unique Region-Aware Temporal Filter (RATF) module that dynamically builds various temporal filters for forged regions to detect deepfake videos. They break up the video into different snippets to capture the long-term temporal irregularity, and we suggest the Cross-Snippet Attention (CSA) method to encourage cross-snippet interaction. Outstanding performance is shown by the proposed framework on four widely used benchmarks (FF++, Celeb-DF, DFDC, and WildDeepfake). Despite demonstrating remarkable performance on four widely used benchmarks, the authors should have discussed the method's resistance to future, more sophisticated DeepFake techniques or its adaptability to other forms of media beyond video.

Wodajo and Atnafu [16] suggested using a convolutional vision transformer. Two parts comprise the Convolutional Vision Transformer: The convolutional neural network

(CNN) and the Vision Transformer (ViT). The CNN retrieves learnable features, and the ViT uses a mechanism called attention to categorize those learned as input. They got 91.5 percent accuracy, an AUC value of 0.91, and a loss value of 0.32 after training our model on the DFDC dataset. While achieving impressive accuracy, the method's applicability to other manipulation techniques and the impact of varying dataset characteristics on its performance still need to be explored.

Kumar et al. [17] conducted an exhaustive analysis of various neural network techniques applicable to the classification of highly compressed DeepFakes. They demonstrated that the proposed metric learning approach could effectively perform this classification. Employing a triadic network structure in the measured learning process proved particularly beneficial when evaluating the authenticity of a smaller quantity of video frames. A notable downside, however, is the method's failure to generalize across different datasets, a fact attributed to the absence of adjustment of unsupervised features to harmonize the feature space between the source and target datasets. Such adaptability could render the model more resilient and less reliant on labels. Despite promising results, the method needed help to generalize across different datasets. This could be attributed to adjusting unsupervised features to harmonize the feature space between other datasets. This limitation highlights the need for a more flexible model that is less reliant on labels.

Elhassan et al. [18] developed and implemented a model known as the Deep-Fake Identification Technique with Mouth Features (DFT-MF). Utilizing a machine learning approach, this model identifies DeepFake videos by selectively focusing on, analyzing, and confirming lip or mouth movements. However, the method's limitation lies in its narrow focus on the mouth area, neglecting the broader facial and body movements in its analysis. However, its narrow focus on the mouth area ignores more general facial and body movements, possibly leaving some deepfake manipulations undetected.

Ahmed et al. [19] delved into using advanced CNN amplification techniques to achieve real-time reconstruction of DeepFake imagery via devices such as video and surveillance cameras. The research effectively merges DeepFake's amplified configuration with targeting, elevating the accuracy rate to 95.77%. However, a minimal discrepancy exists in the estimated costs of implementing this technology. While achieving a high accuracy rate, the research does not delve into the potential discrepancies in the estimated costs associated with implementing this technology, which could impact its adoption.

Gandhi et al. [20] proposed a methodology based on adversarial techniques to improve DeepFake images and evade traditional DeepFake detection methods. Their method incorporates both the Fast Gradient Sign Method (FGSM) and the L2 norm-based attack by Carlini and Wagner in both Blackbox and Whitebox scenarios. However, a notable limitation of this approach is the substantial computational resources needed to manipulate a single image. Further exploration is necessary to broaden the application of these enhanced adversarial attacks to various other domains. However, the significant computational resources required to manipulate a single image may limit its scalability and applicability to different domains.

Das et al. [21] investigated a study to pinpoint the weaknesses and deficiencies within the existing DeepFake detection framework. Their theoretical and empirical scrutiny of ideal traditional datasets and systems revealed that the integration of Face-Cutout

can escalate the overall data variance and mitigate clustering problems while achieving a LogLoss reduction ranging from 15.2% to 35.3% across different datasets. Future research directions involve investigating the application of this reinforcement principle on a broader spectrum of DeepFake datasets. While their integration of Face-Cutout improved data variance and mitigated clustering problems, the research did not explore its application to a wider range of Deepfake datasets.

Suratkar et al. [22] introduced a novel approach, presenting a study that employs transformational learning within CNN. This strategy involves assigning weights to the upper echelons of pre-trained deep CNNs, resulting in superior outcomes in reduced training durations compared to CNN models trained on nonlinear mapping weights for DeepFake video detection. Despite this advancement, the model's efficacy could be augmented by integrating ConvLstm2D (Tensorflow) layers and supplying the network with image sequences rather than isolated images. This modification could address temporal discrepancies in DeepFake videos and feature distortions. Despite its reduced training duration, the model could benefit from integrating ConvLstm2D layers and supplying the network with image sequences rather than isolated images to address temporal inconsistencies in Deepfake videos.

El Rai et al. [23] proposed a technique for distinguishing genuine videos or images from DeepFakes by creating a novel convolutional neural network called the Patch & Pair CNN (PPCNN). In this method, instead of processing the entire face, the face is segmented into smaller frames prior to the face pairs being input into the network. Although PPCNN has demonstrated its effectiveness in detecting DeepFake videos from the same dataset, enhancing its generalizability with a dual-branch learning framework could improve its performance on DeepFake videos stemming from different sources. While PPCNN has demonstrated effectiveness in detecting Deepfake videos from the same dataset, its performance on Deepfake videos from different sources may be improved by enhancing its generalizability with a dual-branch learning framework.

Li et al. [24] draw attention to an escalating issue with partial facial alterations in Deepfake videos, which only execute modifications at the video level, disregarding the manipulation of all faces within the forged videos. The study addresses these Deepfake challenges by amalgamating the input face and video instances and treating them as bags and instances within this learning framework. In contrast to the conventional Multi-Instance Learning (MIL) approach, which typically follows a linear path from instance consolidation to instance projection and then to bag prediction, a novel concept called "sharp MIL" (S-MIL) is introduced. S-MIL directly establishes a pathway from instance consolidation to bag prediction. However, it's worth noting that the FFPMS dataset used in this context hasn't undergone comprehensive testing across various platforms and DeepFake detection methods. Despite the introduction of the "sharp MIL" (S-MIL) concept, the FFPMS dataset used in this context hasn't undergone comprehensive testing across various platforms and DeepFake detection methods, leaving a gap in understanding its performance potential.

Zhang et al. [25] have innovated a trailblazing phantom feature extraction method to enhance the identification of face swap images created from their original counterparts via DeepFake. This method leverages deep learning and Error Level Analysis (ELA) to detect variances in image coding associations. After this, a CNN extracts

these phantom features and ascertains image authenticity. Although the proposed approach effectively sees image manipulation under lossy compression conditions, its performance dwindles in low-quality or lossless coding scenarios. While it shows promise in detecting manipulated images under lossy compression, its efficacy significantly diminishes in low-quality or lossless coding scenarios. This limitation could hinder the method's applicability across digital platforms and image qualities.

Vizoso et al. [26] delve into how significant media entities like the Wall Street Journal, Washington Post, Reuters, and influential internet companies including Google, Facebook, and Twitter, respond to the rise of DeepFakes, viewing them as a new form of disinformation. The research underscores methods of DeepFake detection and contemplates the potential influence of DeepFake on democratic procedures and national security. Nonetheless, the study reveals a notable Western-centric cultural inclination in the digital platforms and media samples analyzed, which may pose issues in extrapolating the findings to similar entities in different cultural settings. However, their research exhibits a Western-centric bias, which may limit the generalizability of their findings to non-Western contexts. This raises the question of how DeepFakes and their implications are perceived and managed in diverse socio-cultural environments.

A highly effective deepfake detection model is suggested by Tran et al. [27] for manipulated video, guaranteeing model correctness while maintaining the proper weight. A high-performance and lightweight model using CNN network was used as the basis. The DFDC dataset yielded an AUC and F1-score of 0.958 and 0.9243, respectively, and Celeb-DF v2 with 26M parameters yielded an AUC and F1-score of 0.978 and 0.9628, respectively. However, the research does not discuss how the model would perform on deepfake techniques that are not video-based. Thus, the model's effectiveness may be limited to video deepfakes. The issue of overfitting is particularly problematic in the context of deepfake detection due to the relative scarcity of deepfake videos for training purposes. Moreover, this research is unsuitable for tasks requiring relational reasoning, i.e., understanding and utilizing relationships between different entities.

Jiang et al. [28] present a novel learning framework, Multiple Graph Learning Neural Networks (MGLNN). This framework is engineered to enable data classification using various graph-based perspectives. MGLNN's primary objective is to amalgamate multi-graph learning with diverse graph structures, thereby identifying the most suitable graph structure that enhances the learning process of GNN. The MGLNN framework is demonstrated to be versatile, catering to multiple graphs utilizing any specified GNN model. Furthermore, the MGLNN model was trained and optimized using a comprehensive approach. Experimental findings from various datasets suggest that MGLNN outshines several comparative methodologies in semi-supervised classification tasks. While the framework outperforms several comparative methodologies in semi-supervised classification tasks, its applicability to other tasks or domains still needs to be explored. Understanding how this framework can be adapted to different use cases beyond semi-supervised classification is imperative. Moreover, this research is unsuitable for tasks requiring relational reasoning, i.e., understanding and utilizing relationships between different entities. Furthermore, this research is unsuitable for simultaneously detecting spatial and temporal dynamics.

Roy et al. [29] endorsed the application of WilDect-YOLO, a high-precision real-time object detection system, for identifying endangered wildlife species. Incorporating spatial pyramid pooling, DenseNet, and a redesigned path aggregation network has significantly bolstered the network's overall performance. The proposed model distinguishes itself by outstripping contemporary advanced models, boasting a mAP and F1 score of 96.89% and 97.87%, respectively, all at a remarkable detection speed of 59.2 frames per second. However, the research must explore how this system would perform in other object detection scenarios. The potential application and performance of WilDect-YOLO in detecting non-wildlife objects or in varied environments is an area that warrants further investigation. Moreover, this research is unsuitable for simultaneously detecting spatial and temporal dynamics. Furthermore, this research is unsuitable for tasks requiring relational reasoning, i.e., understanding and utilizing relationships between different entities.

Hu et al. [30] proposed a novel Deepfake detection method that takes advantage of the inherent flaws in Deepfake videos—unspecific face part discrepancies called Mover. Real faces can be restored quickly, whereas false faces are more difficult to repair due to Mover's randomly selected regions of interest (ROIs), which masks regions of interest (ROIs) and restore faces to learn generic features. Four publicly available Deepfake video datasets FF++, CDF, WildDF, and DFDCP, are used to assess the proposed technique. Numerous tests using industry-recognized criteria show that Mover is quite successful. While the method performs well on several Deepfake video datasets, it fails to discuss the implications of different ROI selection strategies on detection performance, limiting our understanding of optimizing the method for diverse scenarios.

Hussain and Ibraheem [31] proposed a new approach to CNNs in conjunction with the Jaya algorithm optimization, which they provided for precisely identifying deepfake videos. The DFDC dataset and the Celeb-DF dataset, two publicly accessible datasets, are used to assess the methodology. On both datasets, the methodology performs at the cutting edge. With strong F1 scores suggesting a high precision and recall for recognizing deepfake movies, their method achieves an accuracy rate of 99.3% and 97.6% On the DFDC and the Celeb-DF datasets, respectively. In addition, their strategy is more resistant to adversarial attacks than current cutting-edge techniques. While achieving high accuracy rates on the DFDC and the Celeb-DF datasets, the research needs to discuss how the model would perform against new or emerging manipulation techniques. This suggests testing the model against a wider range of deepfake techniques. Moreover, this research is unsuitable for tasks requiring relational reasoning, i.e., understanding and utilizing relationships between different entities.

Using MesoNet and a preprocessing module, Xia et al. [32] suggested a method for detecting Deepfake videos. In order to improve the discriminating between multi-color channels, a preprocessing module is first developed to preprocess the clipped face photos. The traditional MesoNet is then fed with the previously processed photos. The proposed method's detection performance is tested on two datasets; it outperforms existing approaches in terms of AUC on FaceForensics++ (0.974) and Celeb-DF (0.943). Although the method performed well on two datasets, the research does not explore the impact of the preprocessing module on different types of deepfake detection tasks, limiting the method's potential applicability to a broader range of deepfake scenarios.

To summarize this section, the literature on deepfake detection methods is extensive and diverse. Earlier works predominantly revolved around traditional machine-learning techniques and focused on specific cues like facial distortions, lighting inconsistencies, and blink rates. However, these methods often struggle with new, more sophisticated deepfake techniques. More recent studies have leaned towards deep-learning-based approaches. Convolutional neural networks (CNNs), Recurrent Neural Networks (RNNs), and auto-encoder architectures have been employed, often achieving high accuracies. Nevertheless, these approaches commonly face challenges related to overfitting, high computational complexity, and limited generalizability across different deepfake generation methods and datasets. Several research efforts have also explored the use of transfer learning and multimodal detection techniques. While these methods have shown promise, their performance can be heavily dependent on the quality and diversity of training data, and they often struggle to adapt to new deepfake techniques.

In our paper, we propose a GNN-based deepfake detection method that aims to overcome these limitations:

- Overfitting: GNN's reduced complexity compared to deep models like deep CNNs makes it less prone to overfitting. We also employ regularization techniques to improve the model's generalization capabilities.
- Computational Complexity: By leveraging the relational data handling capacity of GNNs, we are able to achieve high detection accuracy without the need for highly complex models, therefore reducing computational requirements.
- Generalizability: Our model is trained and tested across several diverse datasets, ensuring that it performs well under a variety of conditions and deepfake techniques.
- Adaptability: The flexible and scalable nature of GNNs allows our model to adapt to new deepfake techniques, providing a future-proof solution to deepfake detection.

Methodology and proposed model

This section presents a novel model to address these challenges, utilizing minibatch graph neural networks (termed miniGNNs). Comparable to CNNs, miniGNNs can efficiently train the network for deep fake video detection on a down-sampled graph (or topological structure) in a minibatch manner. Additionally, the model trained can be employed directly to predict new data. In addition, we develop three fusion strategies to enhance the detection of deepfake videos; these strategies encompass additive fusion (FuNet-A), element-wise multiplicative fusion (FuNet-M), and concatenation fusion (FuNet-C). These fusion techniques aim to improve the results of deep fake video detection by integrating features extracted from both CNNs and our miniGNNs within a network that can be trained end-to-end. The proposed model is illustrated in Fig. 1. A graph is an intricate nonlinear construct that encapsulates one-to-many associations within a non-Euclidean realm. Regarding our scenario, the relationships among spectral signatures form an undirected graph. Assume an undirected graph, $G = (V, E)$, where V and E correspond to the sets of vertices and edges respectively. In the context of our work, the set of vertices is comprised of image pixels, while the set of edges is formed by the similarities between any pair of vertices, namely v_i and v_j. The adjacency matrix, represented as A, defines the relationships or connections (edges) between nodes/vertices within the

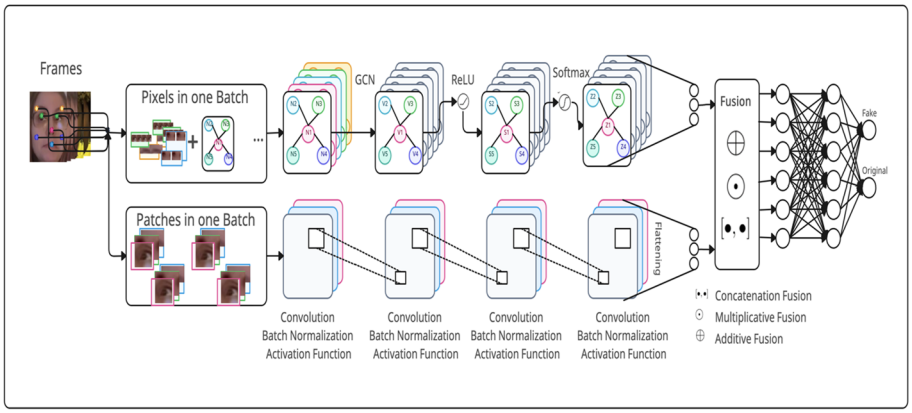

Fig. 1 Architecture of proposed methodology

graph network. Each value in A is commonly determined using a Radial Basis Function (RBF), which evaluates the similarity between node features to establish the strength of connections between nodes.

$$A_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) \tag{1}$$

where σ is a parameter to control the width of the RBF. The vectors x_i and x_j denote the spectral signatures associated with the vertexes v_i and v_j. Given two functions, f and g, their convolution operation can be represented as

$$f(t) * g(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2}$$

where τ indicates the shift distance and * is the convolution operator symbol. The Fourier transform of the convolution of f and g is equal to the pointwise multiplication of their individual Fourier transforms. This relationship can be written as:

$$\mathcal{F}[f(t) * g(t)] = \mathcal{F}[f(t)].\mathcal{F}[g(t)] \tag{3}$$

where F denotes the Fourier transform and · indicates pointwise multiplication. The inverse Fourier transform (denoted by $\mathcal{F}^{-1}$) of the convolution of f and g is equal to 2π times the multiplication of their respective inverse Fourier transforms. Therefore, the convolution operation on a graph can be transformed into the Fourier domain to define the Fourier transform ($\mathcal{F}$) or identify a set of basis functions.

$$\mathcal{F}^{-1}[f(t) * g(t)] = 2\pi\mathcal{F}^{-1}[f(t)].\mathcal{F}^{-1}[g(t)] \tag{4}$$

Deriving facial frames

Our novel miniGCN approach leverages graph convolutional networks to effectively identify manipulated facial regions in video content. We first isolate facial frames using multi-task cascaded convolutional neural networks (MTCNN). These facial images are then converted into visual embeddings and partitioned into patches represented as

graph nodes (as illustrated in Algorithm 1). Edges between nodes are constructed based on feature vector similarity using K-Nearest Neighbors. An aggregation and update function iteratively adjusts edge weights as graphs pass through the miniGCN layers. A vital aspect is the integration of the pyramid Resnet architecture, allowing the model to maintain a small spatial size at increasing depths for enhanced extraction of distinguishing facial features. This specialized design enables robust detection of subtle visual artifacts in deepfakes through comprehensive multiscale facial analysis. Our approach achieves state-of-the-art performance in identifying manipulated videos by combining MTCNN facial detection, graph representations, and pyramid Resnets in the miniGCN framework. MTCNN surpasses other facial detection techniques in its ability to precisely identify subtle facial landmarks like eyes, nose, and mouth in video frames. Unlike Haar cascade and Viola-Jones, MTCNN excels at extracting fine-grained facial details, even under challenging lighting variations and occlusion conditions. This robust performance stems from MTCNN's cascaded architecture that captures facial information coarse to finely across multiple networks. MTCNN achieves unparalleled accuracy in delineating facial boundaries and distinguishing the face from its surroundings by thoroughly analyzing various visual features from different scales. These capabilities make MTCNN an ideal choice for preprocessing video to detect reliably manipulated faces indicative of deepfakes. Its precision in modeling facial geometry ensures comprehensive analysis of visual artifacts within isolated facial regions.

Algorithm 1 MTCNN preprocessing configurations

```
Input: Video_path, detector, max_frames
Output: face_frame in output_file
Start Procedure
cap = cv2.videoCapture(video_path)
 While cap.isOpened ()
   ret, frame = cap.read()
   IF not ret:
      break
   End IF
   frame_count +=1
   IF max_frames and frame_count > max_frames:
     Break
   End IF
   Faces = detector.detect_faces(frame)
   For i, face in enumerate(faces):
        x, y, width, height = face['box']
        face_frame = frame[y:y+height, x:x+width]
        cv2.imwrite(output_file, face_frame)
   End For
  End While
  cap.release()
  End Procedure
```

Transition from image to visual representations

The Mini-GCN requires graphs as input, so we transform images into equivalent graphs with nodes and edges using an image-to-visual embedding process. Each facial frame of

dimensions $224 \times 224 \times 3$ is divided into N patches using patch embeddings to provide a compact representation. The image-to-graph process involves:

- Expanding the patch window so adjacent windows overlap by half.
- Padding the feature map with zeros to maintain resolution.
- Applying 4 depthwise convolution layers with 1-pixel padding between fully connected layers and ReLU activation as shown in GCN architecture.
- The convolution layer accepts an input size of $224 \times 224 \times 3$ with a specified stride, kernel size, padding, and number of kernels.

GCN Architecture

Input: Processed Trained Frames (PTCXRIMG)
Output: Trained Model (Net)
Procedure GCN_Architecture:
Initialize Proposed_Model = GCN.CreateModel()
Proposed_Model.add(Input layer)
Proposed_Model.add(Convolution block)
Proposed_Model.add(Normalization layer)
Proposed_Model.add(ReLU layer)
Proposed_Model.add(Pooling layer)
Proposed_Model.add(Dropout layer)
Proposed_Model.add(Convolution block)
Proposed_Model.add(Normalization layer)
Proposed_Model.add(ReLU layer)
Proposed_Model.add(Pooling layer)
Proposed_Model.add(Dropout layer)
Proposed_Model.add(Flatten layer)
Proposed_Model.add(Convolution block)
Proposed_Model.add(Normalization layer)
Proposed_Model.add(ReLU layer)
Proposed_Model.add(Pooling layer)
Proposed_Model.add(Dropout layer)
Proposed_Model.add(Flatten layer)
Proposed_Model.add(FullyConnected layer)
Proposed_Model.add(Sigmoid layer)
Proposed_Model.add(Classification layer)

Define training options (Opt) with:
Initial_Learning_Rate = 0.0002
Initial_Drop_Rate = 0.6
Batch_Size = 128
Max_Epochs = 30

Net = TrainNetwork(PTCXRIMG, Model, Opt)

Return Net

End Procedure

Graph neural network

GNNs inherently operate on nodes rather than images. Therefore, the patches derived from converting images to visual embeddings transform nodes that the network can effectively process as part of a graph structure. These patches, originating from individual facial frames and represented as patch embeddings, serve as the nodes within this graph. Every patch is treated as a distinct node, and connections between nodes

are established using the K-Nearest Neighbor method, employing the feature vectors associated with each node. Consequently, this process yields a graph, which serves as input to the aggregation and update function, responsible for iteratively adjusting edge weights. The pyramid ResNet architecture incorporates multiscale image characteristics while maintaining a compact spatial size as the layer depth increases. This characteristic dramatically enhances the model's capacity to discern distinctive features. In addition, the MTCNN is leveraged to extract facial frames from video content by identifying facial landmarks such as nose, eyes, and mouth. Figure 2 depicts the process for generating batches in the proposed miniGCNs. Like CNNs, this batching approach samples nodes to form each batch; however, a key difference is that after each sample, the graph or adjacency matrix within the produced batch must be reconstructed according to the connections in graph G. In a data preparation phase, CNNs process input patches individually to create single-instance encoded labels as output. Conversely, Graph Convolutional Networks (GCNs) operate on individual pixel samples alongside an adjacency matrix representing relationships between samples. This matrix must be calculated before training begins. Furthermore, CNNs effectively extract local spatial and spectral details from HS images at the feature representation phase. Meanwhile, thanks to the adjacency matrix, GCNs leverage their graph structure to model spatial connections between proximate, intermediate, and distant samples. Also, as deep learning models at the network training phase, CNNs typically employ mini-batch training strategies. In contrast, GCNs necessitate full-batch training since all samples must be fed to the network simultaneously to properly account for their graph-based relationships, as defined by the adjacency matrix.

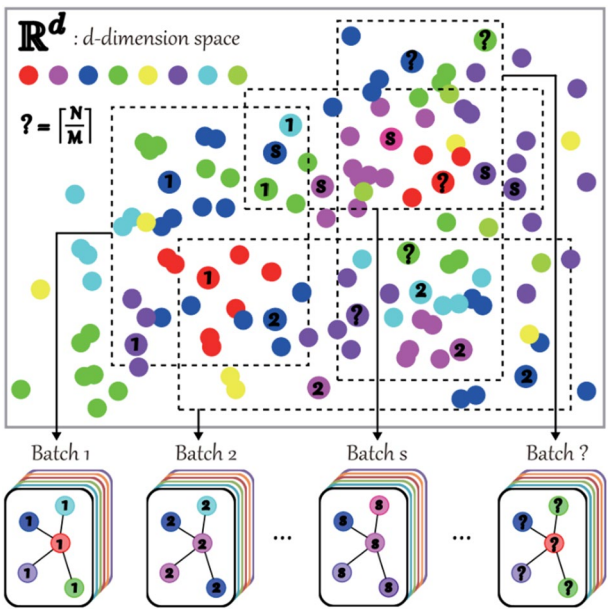

Fig. 2 Showcases an essential process within the miniGCNs' operation: extracting sub-graphs or 'batches' from an encompassing graph, denoted as G

Algorithm 2 Pruned net

```
Input: Original Network (Net), Pruning Percentage (PP), Maximum Pruning (MP)
Procedure Prune_Network:
   Train and evaluate Net on dataset B
   While PP is less than or equal to MP do
      1. Calculate the total number of filters in each hidden layer.
      2. Identify and remove a percentage of filters in each hidden layer
         based on the highest Average Percentage of Zeros (APoZ).
      3. Retrain the pruned network on dataset Net, and select
         the optimal weights after pruning.
      4. Increase PP by one (PP = PP + 1).
   End While
   Return MP + 1
End Procedure
```

Constructing the graph

The graph construction process involves extracting image patches from facial regions and converting them into feature vectors $x_i \in R^d$ of dimension d, where $x = [x_1, x_2, \ldots, x_n]$,$i = [1, 2, 3, \ldots, n]$ for n patches. The feature vectors are treated as unordered nodes as $U = [u_1, u_2, \ldots, u_n]$. To form connections, we find the K nearest neighbors $\beta(u_i)$ for each node u_i. Edges e_{ij} are then created connecting u_j to u_i for all u_j belonging to β (u_i). The result is a facial graph $G' = (U,E)$ with nodes U and edges E established based on feature vector similarities of image patches. This graph representation, tailored to facial geometry, allows capturing subtle manipulated artifacts by analyzing relationships between visual features extracted from key facial regions.

- Each patch is transformed into an M-dimensional feature vector yi, constituting the set of unordered nodes U.
- For each node u_i, we find the K nearest neighbors $\beta(u_i)$ based on feature similarities.
- An edge e_{ij} is created from neighbor u_j to node u_i.
- All edges constitute the set E, representing the graph as $G' = (U,E)$.
- This entire graph creation process is denoted as $G' = G(y)$.

Manipulating the graph structure

The initial facial graph $G' = G(y)$ is constructed from image feature vectors y as described earlier. This graph is then passed through graph convolutional layers to allow information sharing between connected nodes. Specifically, the graph convolution operation consists of two components:

- Aggregation (L_{agg}): Node features are aggregated by combining features from neighboring nodes using learnable aggregation weights.
- Update (L_{udate}) : The aggregated neighbor features are used to update the features of each node through a set of learnable update weights.

This localized filtering and feature aggregation between neighboring nodes enables the model to jointly analyze relationships among facial patches, identifying subtle inconsistencies in manipulated images.

$$G'' = F\left(G', L\right) = update(aggregate(G', L_{agg}), L_{update}) \tag{5}$$

As shown in Eq. (5), graph convolution involves two key steps—aggregation and update, parameterized by learnable weights L_{agg}, and L_{update}. The aggregation function calculates node representations by aggregating features from neighboring nodes. This captures local contextual information around each node. The update function then combines these aggregated neighbor features to compute updated representations for each node. Mathematically, this update process can be expressed as:

$$x_i' = h\left(x_i, g\left(y_i, \beta(x_i), L_{agg}\right), L_{update}\right) \tag{6}$$

where $\beta\left(y_i\right)$ signifies the neighboring nodes of y_i, while g and h represent the aggregation and update functions, respectively. By aggregating local neighbor information and updating node features accordingly, the model iterates through multiple graph convolution layers to jointly analyze relationships between facial patches. This allows identifying subtle inconsistencies and manipulated features in graphs constructed from deepfake images. In the case of max pooling aggregation, it selects the maximum value among the patches with high magnitudes. Each neighbor's vector undergoes processing through a fully connected layer and is subsequently aggregated using max pooling, which aids in minimizing information loss. It is shown as:

$$g(*) = x_i'' = \left[x_i, \max\left(\left\{x_j - x_i | \in \beta(x_i)\right\}\right)\right] \tag{7}$$

The core of our model is the max-relative graph convolution operation $g(*)$ defined in Eq. (7). This computes node features by aggregating information from neighboring nodes using max pooling, capturing useful neighborhood characteristics. The multi-top update approach further distinguishes real and fake samples by dividing aggregated features into tops, iteratively adjusting the tops with new weights, and concatenating the results to update nodes concurrently. Specifically, the combined features x_i'' are split into t tops $[top^1, top^2, \ldots, top^t]$ and updated each iteration before concatenation. The max pooling aggregation, max-relative graph convolution, and multi-top update enable comprehensive analysis of subtle differences between real and manipulated faces. This specialized graph learning approach is crucial for detecting the distinct artifacts introduced in deepfake creation. Our model leverages these techniques to analyze facial regions and discern manipulated videos comprehensively.

$$x_i' = \left[top^1 L_{update}^1, top^2 L_{update}^2, \ldots, top^t L_{update}^t\right] \tag{8}$$

Figure 3 provides insight into the evolution of relationships over time. It depicts an image graph constructed from two sequential moments—an initial state and a later stage. In the beginning, as seen in 3(b), connections are tightly bound by similarities in characteristics. Nodes affiliate most closely with those sharing kindred qualities. This reflects a network organized principally by content-based affinities. However, as shown

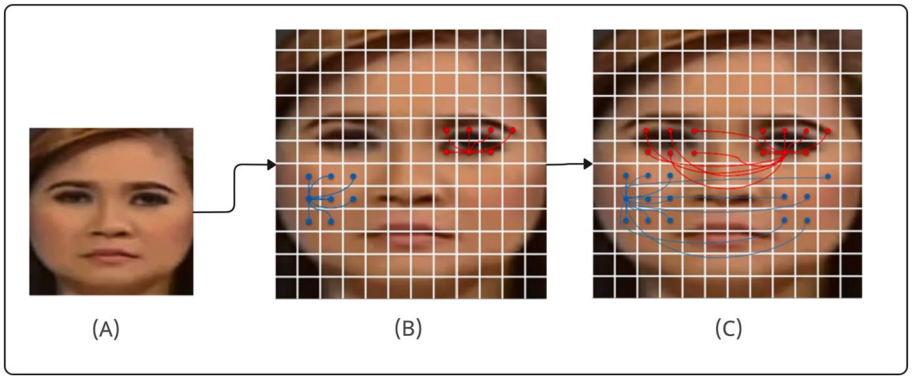

Fig. 3 Presents a graph where nodes of identical colors signify their immediate adjacency to a particular node. **A** Input image. **B** Graph connectivity at the first sequential block. **C** Graph connectivity at the 12th sequential block

moving to 3(c), the passage of time brings continual reassessment and recalibration of relations. This signals a shift—from a structure staked first to likenesses evident soon, to connections cutting deeper to likenesses revealed through slow unfolding. The graph transitions from particularized to integrated, from initially bounded to comprehending the more profound totality of its depiction. In this way, Fig. 3 hints at how relationships, given time, may mature beyond surface comparisons to tap into latent unities underneath. It suggests how networks evolve from fragmentary to holistic as broader truths emerge through progressive intertwining over sequential moments.

Model architecture

The proposed method uses a hierarchical structure that leverages multiscale image properties and reduces spatial size as layers get deeper. This helps the model identify unique attributes more effectively. Hierarchical structures improve accuracy and reduce parameters, leading to robust models. They are well-suited for image datasets since they can capture distinct sample characteristics while reducing complexity. Image patches are converted into graphs within the miniGNN. The model consists of concatenated and abbreviated blocks as shown in Fig. 1. Each block contains a GraphNet subblock and an FFN subblock. GraphNet includes convolutional layers with batch normalization and ReLU activation, a graph convolution layer, and another convolutional layer. Linear layers are used before and after graph convolution to combine node features and enhance diversity. ReLU activation is used after graph convolution to minimize interference between layers.

Different neural networks can extract unique representations from this data: CNNs extract spatial-spectral features while GCNs model interrelationships between samples. However, no single model captures all useful information. We propose an intuitive fusion of CNNs and GCNs to boost discriminative power. Unlike traditional GCNs, our miniGCNs can be trained incrementally and integrated into CNNs. The resulting end-to-end fusion network, FuNet, combines the benefits of both architectures. We consider three fusion strategies:

- Additive fusion (A): The CNN and GCN outputs are summed. This simple linear combination exploits the complementary nature of the two representations.
- Multiplicative fusion (M): The outputs are element-wise multiplied, allowing them to modulate each other and encode more complex interactions.
- Concatenative fusion (C): The outputs are concatenated, allowing the model to learn the most informative combination of features.

Our proposed approach optimally rearranges layers to discriminate real from fake data effectively. This arrangement consists of: (A) Convolutional layers to extract features. (B) Batch normalization for stability. (C) ReLu activation to address the vanishing gradient problem, enabling better model fitting with few additional resources while reducing overfitting. (D) Dropout for regularization. (E) The softmax function activates the Fully connected layer, which probabilistically converts the output vector for classification. (F) The softmax function, applied in the last layer, provides a probability distribution when comparing real and fake samples. This reveals how confident the model is in its predictions. Table 1 summarizes the critical configuration of the graph neural network (GNN) model, showing the underlying design philosophy. The dimensional parameters signify the following:

- 'FD'—the feature dimension, capturing the essential properties of the data
- 'HD'—the hidden dimension ratio in the neural network, determining its representational power
- 'N'—the number of neighbors each graph node is connected to, shaping the network topology
- 'H x W'—the width and height of input images, specifying the spatial extent of the data

Configuration of experimental results

This section outlines the datasets, performance metrics, and experimental protocol used to rigorously evaluate the proposed approach. We conduct extensive experiments on various large-scale datasets to test the model thoroughly. The datasets provide real-world examples for testing the ability of the approach to generalize. Rigorous experimentation

Table 1 Detailed settings of graph neural network

Stages	Output size	Parameter
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	FD = 80 HD = 4 N = 9
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	FD = 160 HD = 4 N = 9
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	FD = 400 HD = 4 N = 9
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	FD = 600 HD = 4 N = 9

Table 2 List of deepfake datasets

Dataset	Release year	No. Fake–No. Real	Source
FaceForensics++ [33]	2019	4K–1K	Youtube
DFDC [34]	2020	> 100K for both	Celebrities
Celeb-DF [35]	2019	5639–590	Youtube

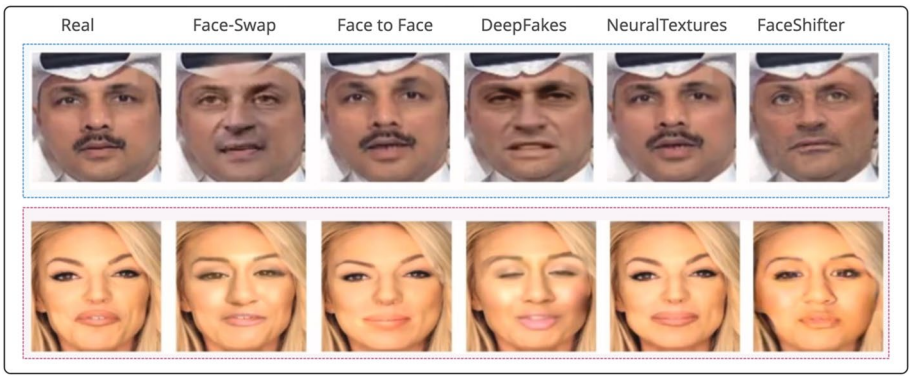

Fig. 4 Representative images from the FaceForensics++ Dataset

exposes strengths and weaknesses, informing areas for improvement. The performance metrics quantify how well the model achieves its objectives, revealing the true impact of various design decisions. Only through careful metric selection can we properly evaluate different approaches. The details of the experiments—models compared, training parameters, preprocessing steps, etc.—are described to ensure reproducibility and fair comparison. This demonstrates the robustness and versatility of the proposed approach under different settings. By testing on multiple datasets with varying characteristics, using well-suited performance metrics, and delineating experimental details precisely, we aim to provide an objective and comprehensive evaluation of the model's effectiveness. Only through such rigorous experimentation can we gain valuable insights and effectively benchmark against other methods.

Datasets

This section evaluates the performance of the proposed method using three distinct datasets: FaceForensics++ (FF++), DFDC, and Celeb-DF, as outlined in Table 2. FaceForensics++ dataset is renowned for its complexity, comprising 1000 authentic YouTube videos featuring various faces, accessories, lighting conditions, and angles. This diversity presents a formidable challenge in distinguishing accurate content from deepfake material. Some samples of FaceForensics++ are shown in Fig. 4. Various deep learning and computer graphics techniques were used to manipulate genuine videos, introducing intricate variations. The DFDCP dataset, on the other hand, encompasses 5000 videos featuring both authentic and driven content, all performed by professional actors. The creation of fake videos involved the utilization of deepfake and GAN techniques. This dataset captures a spectrum of acquisition scenarios, lighting conditions, poses, and diversities, encompassing gender, age, and skin tones, rendering it highly representative.

Fig. 5 Exemplary instances from the DeepFake Detection Challenge (DFDC) Dataset

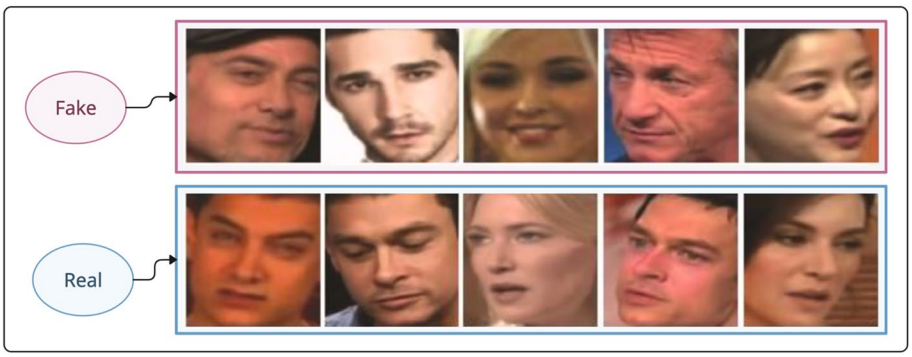

Fig. 6 Selected examples from the Celeb-DF Dataset

Table 3 Tuning parameters

Hyperparameters	Value
Learning Rate	0.0002
Optimizer	AdamW
Batch Size	128
Dropout Rate	0.6
Epochs	30

Some DFDC samples are showcased in Fig. 5. Lastly, the Celeb-DF dataset includes 590 real and 5639 fake celebrity videos sourced from YouTube. Some samples of Celeb-DF are shown in Fig. 6. The actual videos exhibit variations in face size, orientation, lighting, and background—reflecting real-world complexity. The deepfake creation method optimizes facial brightness/contrast, minimizing discrepancies—making the manipulated videos visually deceptive.

Experiment procedures

The research methodology involved carefully studying facial frames extracted using the MTCNN algorithm resizing them to a resolution of 128×128 pixels. Across all experiments, an 80:20 training–testing data split was maintained, with 80% of frames allocated for model training and the remaining 20% assigned for performance evaluation. The proposed model was built using the PyTorch deep learning framework. Table 3 lists the

tuning parameters selected during the training phase. The entire experimentation was conducted on a powerful computing cluster featuring 4 NVIDIA Tesla V100 16G GPUs for accelerating calculations, 192 GB RAM for handling extensive data, and 48 processor cores clocked at 2.10 GHz for executing operations quickly. Additionally, most videos in the Celeb-DF and DFDC datasets are relatively short, typically lasting just a few seconds. Therefore, we suggest using 40 frames per video as this duration matches the videos' lengths and ensures sufficient training duration.

Evaluation of the proposed model's performance

To establish the effectiveness of the proposed deepfake identification technique, thorough experiments were carried out utilizing the FF++, Celeb-DF, and DFDC datasets. Each portion of these datasets includes two types of samples: real ones with unmodified recordings representing one class and fake ones with manipulated content signifying the other class. The proposed model's performance was evaluated using 20% of each dataset's samples kept separate during training. The results of these experiments are discussed in depth in the subsequent subsections. They aim to demonstrate how well the model achieves its intended purpose by accurately classifying real and fake samples across different datasets. In a confusion matrix, each column represents instances of a predicted class, while each row signifies instances of an actual class as shown in Fig. 7. Unmodified images are referred to as the positive class, whereas manipulated images are identified as the negative class. The evaluation criteria for detection and classification incorporate several common statistical metrics, defined by Eqs. (1)–(5) below.

- True Positive (TP): This refers to the number of positive samples (unmodified images) correctly identified in the dataset.
- False positive (FP): This signifies the count of negative samples (manipulated images) incorrectly classified as positive in the dataset.
- True Negative (TN): This represents the total of negative samples (manipulated images) accurately recognized in the dataset.

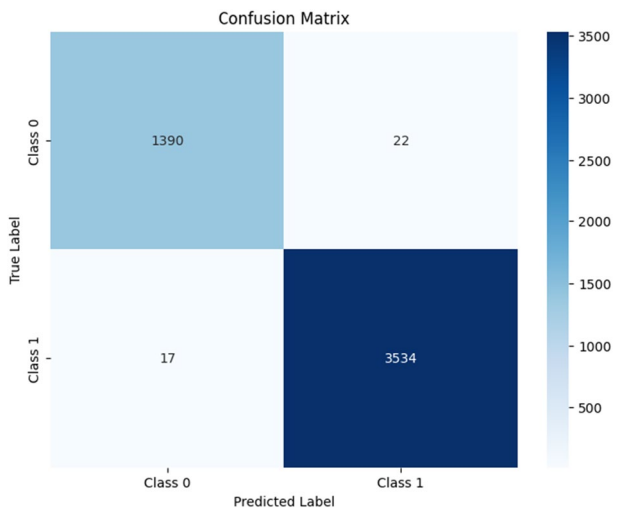

Fig. 7 Confusion matrix

- False Negative (FN): This indicates the number of positive samples (unmodified images) inaccurately classified as negative in the dataset.
- Accuracy: As defined in Eq. 9, this represents the proportion of correctly identified samples across the entire dataset.

$$Accuracy = \frac{Tp + Tn}{Tp + Fp + Tn + Fn} \tag{9}$$

- Precision: As detailed in Eq. 10, this is the proportion of correctly identified positive samples relative to all samples labeled as positive.

$$Precision = \frac{Tp}{Tp + Fp} \tag{10}$$

- Recall: As specified in Eq. 11, this is the proportion of correctly identified positive samples relative to all actual positive samples.

$$Recall = \frac{Tp}{Tp + Fn} \tag{11}$$

- F1 Score: As prescribed in Eq. 12, this harmonizes precision and recall by taking their harmonic mean, weighing both metrics equally.

$$F1 - score = \frac{2 \times (precision \times recall)}{precision + recall} \tag{12}$$

Discussion

This experiment was devised to assess the effectiveness of the proposed method in classifying various categories of deepfakes. The model was systematically tested on each subset of the FF++ dataset and exhibited exceptional performance in detecting Deepfakes. This underscores its robust capability to discern face swaps generated through deep learning techniques. Notably, the model excelled in identifying distinguishing characteristics between manipulations and static textures. The detection accuracy was relatively

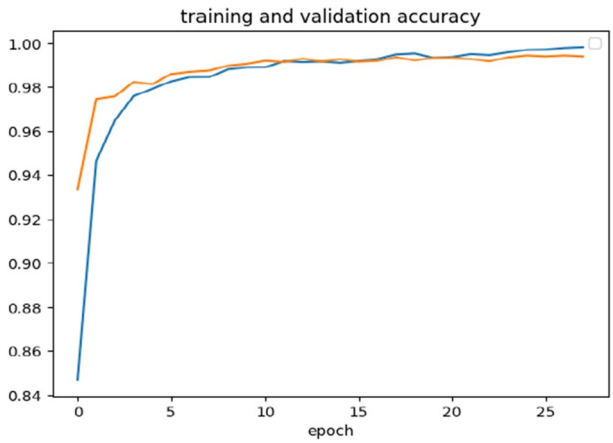

Fig. 8 Chart of DFDC training and validation accuracy

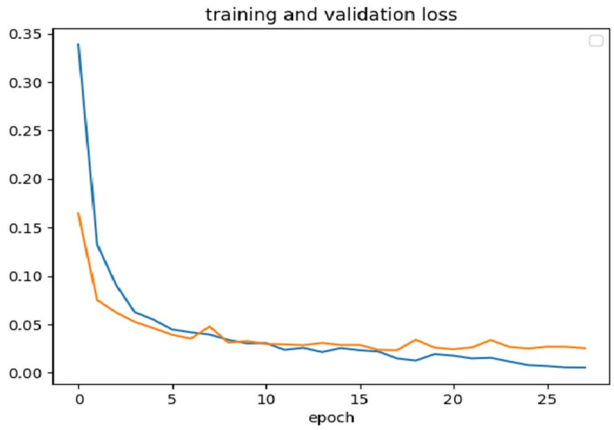

Fig. 9 Chart of DFDC training and validation loss

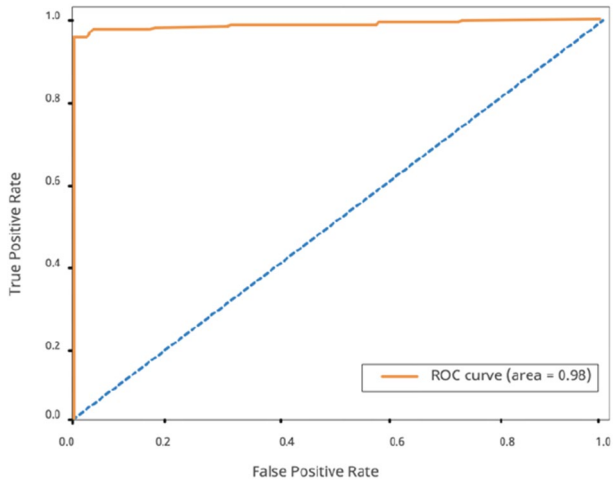

Fig. 10 Chart of Celeb-DF ROC curve

lower for Face2Face and NeuralTextures, measuring 92.49% and 95.09%, respectively. These categories employed expression swapping to create faces with minimal semantic alterations, challenging their detection. This underscores the intricacies involved in identifying this manipulation. The results demonstrated remarkable efficacy, achieving an accuracy level of 99.3% and a validation loss of under 5%, as depicted in Figs. 8 and 9, respectively.

The model's capacity for deepfake detection was further evaluated by pitting real Celeb-DF samples against a wide array of fake samples. The model achieved a remarkable accuracy rate of 98.9% and an AUC of 0.98, as depicted in Fig. 10. It's worth noting that Celeb-DF grapples with a significant class imbalance, featuring 590 real videos compared to 5639 fake ones. However, the proposed method relies on identifying sample characteristics associated with color and texture alterations. Despite this class imbalance, the model demonstrated its ability to differentiate between highly realistic swapped faces within the Celeb-DF dataset. These swapped faces exhibited minimal

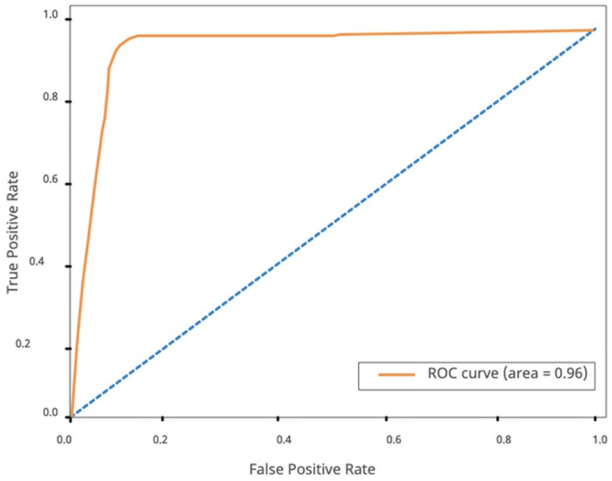

Fig. 11 Chart of DFDC ROC curve

Table 4 Evaluating the effectiveness of various activation functions

Activation Function	Accuracy on DFDC	Accuracy on FF++	Accuracy on Celeb-DF
GeLU	90.19	91.88	91.45
Hswich	87.9	95.4	93.6
LeakyReLU	85.1	71.3	83.7
ReLU	92.4	97.16	95.8

color discrepancies and reduced temporal flickering, posing a considerable challenge for forgery detection.

The effectiveness of the proposed approach in detecting deepfakes within the DFDC dataset was assessed by pitting authentic samples against manipulated ones. The results showcased exceptional performance, with an accuracy rate of 99.3% and an AUC of 0.96, as illustrated in Figs. 8 and 11, respectively. Notably, a substantial portion of the DFDC videos were recorded in dimly lit conditions, posing a considerable challenge. However, the method could distinguish between actual and manipulated samples, even in low-light scenarios and when individuals were captured inside profiles. This capability holds significance, given that many videos within the dataset feature actors engaged in conversations while facing each other in profile view. Despite the dataset's diversity and these challenging variations, the proposed model demonstrated remarkable proficiency in detecting deepfake artifacts.

Numerous experiments were conducted to assess the impact of different activation functions, aiming to comprehend their influence on the proposed technique. These experiments involved subjecting the model to various activation functions, and the outcomes have been meticulously recorded in Table 4. Notably, among the activation functions examined, the ReLU function employed in the proposed method displayed superior performance across both datasets. An intriguing characteristic of ReLU is its adjustable parameter, which can be fine-tuned during the training process to enhance

Table 5 Assessment of performance across varying graph convolution methods

Graph Convolution	Accuracy on DFDC	Accuracy on FF++	Accuracy on Celeb-DF
GraphSAGE	61.6	79.9	75.4
Edge GraphConv	73.19	92.3	88.9
GIN	43.9	56.11	51.4
MR GraphConv	99.3	97.17	92.2

Table 6 Empirical analysis of different detection methods on various datasets

Refs.	Detection method	Dataset	Result
Xia et al. [32]	MesoNet and a preprocessing module	FF++	AUC = 0.974
		Celeb-DF	AUC = 0.943
Agarwalet al. [7]	Temporal, behavioral biometric with CNN	FF++, DFDC-P and Celeb-DF	Less susceptible to counterattacks and generalizes effectively
Hussain and Ibraheem [31]	CNNs in conjunction with the Jaya algorithm optimization	DFDC	Accuracy rates = 98.3%
		Celeb-DF	Accuracy rates = 97.6%
Wodajo and Atnafu [16]	Convolutional Vision Transformer	DFDC	Accuracy = 91.5% AUC = 0.91 Loss value = 0.32
Kharbat et al. [5]	Region-Aware Temporal Filter (RATF)	FF++ and Celeb-DF	Outstanding performance
Proposed Model	Multi Fusion between GNN and CNN	FF++	Accuracy rate = 95.09%
		DFDC	Accuracy rate = 99.3% AUC = 0.96
		Celeb-DF	Accuracy rate = 98.9% AUC = 0.98

the network's overall performance. This parameter governs the slope of the function's negative part. The outstanding performance of ReLU in the proposed method can be attributed to its capacity to adaptively learn the negative slope during training, setting it apart from other activation functions.

To evaluate the effect of the graph convolution, many experiments were conducted. These experiments scrutinized the impact of various graph convolutional approaches on the proposed method. The technique was tested with MaxRelative (MR) graph convolution, GraphSAGE, Edge GraphConv, and Graph Isomorphism Network (GIN). As Table 5 shows, MR graph convolution outperformed other variants in accuracy. GNNs' dynamic neighbor shifting helps alleviate over-smoothing, broadening receptive fields substantially. MR GraphConv enhances this by recomputing edges between vertices in each layer's feature space, combining optimal aspects of GraphSAGE, GIN, and Edge GraphConv to surpass individual methods. Rooted in relative positioning, MR GraphConv assimilates more neighborhood information, yielding more expressive node representations encompassing complex graph relationships. Designed for scalability and robustness, MR GraphConv outperformed Edge GraphConv, GraphSAGE, and GIN, which become computationally burdensome with larger graphs. To analyze more extensive networks efficiently, MR GraphConv reduces dimensionality using max-pooling

aggregation. Notably, it uses max-pooling rather than concatenation during neighbor aggregation, shrinking feature vectors and mitigating memory usage.

Empirical analysis of other deep fake methodologies

Table 6 provides a comparative overview of various deepfake detection methods and their performance metrics on different datasets. Xia et al. [32] utilized MesoNet with a preprocessing module on the FF++and Celeb-DF datasets, achieving AUC scores of 0.974 and 0.943, respectively. Agarwal et al. [7] proposed a method combining temporal and behavioral biometric features with a CNN, demonstrating robustness against counterattacks and effective generalization across FF++, DFDC-P, and Celeb-DF datasets. Hussain and Ibraheem [31] employed CNNs alongside the Jaya algorithm optimization on the DFDC and Celeb-DF datasets, reporting accuracy rates of 98.3% and 97.6%, respectively. Wodajo and Atnafu [16] implemented a Convolutional Vision Transformer on the DFDC dataset, achieving an accuracy of 91.5%, an AUC of 0.91, and a loss value of 0.32. Kharbat et al. [5] introduced a Region-Aware Temporal Filter (RATF) method, showing outstanding performance on the FF++and Celeb-DF datasets. Lastly, the proposed model, which is a fusion between GNN and CNN, demonstrated accuracy rates of 95.09%, 99.3%, and 98.9% on the FF++, DFDC, and Celeb-DF datasets, respectively, with corresponding AUC scores of 0.96 and 0.98.

Conclusion

This manuscript introduced a novel, generalized, and interpretable GNN model designed to detect synthetic facial images created through various deceptive techniques. The model's hierarchical structure allows it to capture subtle characteristics within the frames, enhancing feature representation and detection precision. Our GNN model incorporates activation recalibration and variable refinement to optimize performance while emphasizing discriminative features. It further takes advantage of both content and subsurface relationships through graph connections, fostering a comprehensive understanding of the data. The detection process is divided into two phases: a mini-batch graph convolution network stream and a four-block CNN stream that includes convolution, batch normalization, and activation functions. The final step is a flattening operation, which connects the convolutional layers to the dense layer. These two streams are integrated using three different fusion networks: FuNet-A, FuNet-M, and FuNet-C. Extensive testing across four diverse datasets demonstrated the model's remarkable ability to identify various forms of deepfakes, including impersonation and trait or expression mimicking. The model exhibited remarkable adaptability, performing well regardless of the dataset, evaluation type, or source. This signifies that our model effectively grasps the underlying patterns that transcend surface differences in deepfake media. Moving forward, we plan to refine our model continually to enhance its deepfake detection capabilities across various formats. The main goal is to improve the model's ability to discern superficial falsity and deepen its understanding of genuine content, thereby creating a more effective sentinel to safeguard the truth. Additionally, our model demonstrated impressive training and validation accuracy of 99.3% after 30 epochs when evaluated on different datasets. This achievement underscores the potential of our

proposed GNN approach in tackling the growing challenge of deepfake detection. While our current model uses FuNet-A, FuNet-M, and FuNet-C fusion networks, future work will investigate other fusion techniques to determine if they can improve performance. Also, we plan to refine our model for real-time deepfake detection. This is particularly important for applications like live video feeds, real-time broadcasting, and social media platforms. While our model is already interpretable, we aim to improve its explainability further. This can lead to a better understanding of the features and patterns it uses for detection, which can be invaluable for refining the model and developing new detection techniques. As deepfake technology evolves, new detection challenges will arise. We will continually adapt and update our model to ensure it remains effective against the latest deepfake techniques.

Acknowledgements
Researchers Supporting Project number (RSP2024R167), King Saud University, Riyadh, Saudi Arabia.

Author contributions
All authors read and approved the final manuscript.

Funding
This project is funded by King Saud University, Riyadh, Saudi Arabia.

Availability of data and materials
Not applicable. For any collaboration, please contact the authors.

Declarations

Ethics approval and consent to participate
The author confirms the sole responsibility for this manuscript. The author read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 10 September 2023 Accepted: 19 January 2024
Published online: 01 February 2024

References

1. Nguyen TT, Nguyen QVH, Nguyen DT, Nguyen DT, Huynh-The T, Nahavandi S, et al. Deep learning for deepfakes creation and detection: a survey. Comput Vis Image Underst. 2022;223: 103525.
2. Ahmed SR, Sonuc E, Ahmed MR, Duru AD. Analysis survey on deepfake detection and recognition with convolutional neural networks. HORA 2022 - 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings. 2022;
3. Suratkar S, Kazi F. Deep fake video detection using transfer learning approach. Arab J Sci Eng. 2022;48:9727–37. https://doi.org/10.1007/s13369-022-07321-3.
4. Salvi D, Liu H, Mandelli S, Bestagini P, Zhou W, Zhang W, et al. A robust approach to multimodal deepfake detection. J Imaging. 2023;9:122.
5. Kharbat FF, Elamsy T, Mahmoud A, Abdullah R. Image feature detectors for deepfake video detection. Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA. 2019;2019-November.
6. Zhang D, Lin F, Hua Y, Wang P, Zeng D, Ge S. Deepfake video detection with spatiotemporal dropout transformer. MM 2022 - Proceedings of the 30th ACM International Conference on Multimedia. 2022;5833–41. https://doi.org/10.1145/3503161.3547913
7. Agarwal S, Farid H, El-Gaaly T, Lim SN. Detecting deep-fake videos from appearance and behavior. 2020 IEEE International Workshop on Information Forensics and Security, WIFS 2020. 2020. https://arxiv.org/abs/2004.14491v1. Accessed 31 Aug 2023.
8. Nirkin Y, Wolf L, Keller Y, Hassner T. DeepFake detection based on discrepancies between faces and their context. IEEE Trans Pattern Anal Mach Intell. 2022;44:6111–21.
9. Wan D, Cai M, Peng S, Qin W, Li L. Deepfake detection algorithm based on dual-branch data augmentation and modified attention mechanism. Appl Sci. 2023;13:8313.
10. Shobha Rani RB, Kumar Pareek P, Bharathi S, Geetha G. Deepfake video detection system using deep neural networks. 2023 IEEE International Conference on Integrated Circuits and Communication Systems, ICICACS 2023. 2023;
11. Rana MS, Nobi MN, Murali B, Sung AH. Deepfake detection: a systematic literature review. IEEE Access. 2022;10:25494–513.
12. Mary A, Edison A. Deep fake Detection using deep learning techniques: a literature review. 2023 International Conference on Control, Communication and Computing, ICCC 2023. 2023;

13. Gil R, Virgili-Gomà J, López-Gil JM, García R. Deepfakes: evolution and trends. Soft Comput. 2023;27:16. https://doi.org/10.1007/s00500-023-08605-y.
14. Montserrat DM, Hao H, Yarlagadda SK, Baireddy S, Shao R, Horvath J, et al. Deepfakes detection with automatic face weighting. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. 2020:2851–9. https://arxiv.org/abs/2004.12027v2. Accessed 16 Aug 2023.
15. Gu Z, Yao T, Chen Y, Yi R, Ding S, Ma L. Region-aware temporal inconsistency learning for deepfake video detection. IJCAI International Joint Conference on Artificial Intelligence. 2022;2:920–6
16. Wodajo D, Atnafu S. Deepfake video detection using convolutional vision transformer. 2021. https://arxiv.org/abs/2102.11126v3. Accessed 31 Aug 2023.
17. Kumar A, Bhavsar A, Verma R. Detecting deepfakes with metric learning. 2020 8th International Workshop on Biometrics and Forensics, IWBF 2020 - Proceedings. 2020;
18. Elhassan A, Al-Fawa'reh M, Jafar MT, Ababneh M, Jafar ST. DFT-MF: Enhanced deepfake detection using mouth movement and transfer learning. SoftwareX. 2022;19: 101115.
19. Ahmed SRA, Sonuç E. Deepfake detection using rationale-augmented convolutional neural network. Appl Nanosci. 2023;13:1485–93. https://doi.org/10.1007/s13204-021-02072-3.
20. Gandhi A, Jain S. Adversarial perturbations fool deepfake detectors. Proceedings of the International Joint Conference on Neural Networks. 2020. https://arxiv.org/abs/2003.10596v2. Accessed 16 Aug 2023.
21. Das S, Seferbekov S, Datta A, Islam MS, Amin MR. Towards solving the deepfake problem: an analysis on improving deepfake detection using dynamic face augmentation. Proceedings of the IEEE International Conference on Computer Vision. 2021. 2021-October:3769–78. https://arxiv.org/abs/2102.09603v3. Accessed 16 Aug 2023.
22. Suratkar S, Kazi F, Sakhalkar M, Abhyankar N, Kshirsagar M. Exposing Deepfakes using convolutional neural networks and transfer learning approaches. 2020 IEEE 17th India Council International Conference (INDICON). 2020;
23. El Rai MC, Al Ahmad H, Gouda O, Jamal D, Talib MA, Nasir Q. Fighting deepfake by residual noise using convolutional neural networks. 2020 3rd International Conference on Signal Processing and Information Security, ICSPIS 2020. 2020;
24. Li X, Lang Y, Chen Y, Mao X, He Y, Wang S, et al. Sharp multiple instance learning for deepfake video detection. MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia. 2020;1864–72. http://arxiv.org/abs/2008.04585. Accessed 16 Aug 2023.
25. Zhang W, Zhao C, Li Y. A novel counterfeit feature extraction technique for exposing face-swap images based on deep learning and error level analysis. Entropy. 2020;22:249.
26. Vizoso Á, Vaz-álvarez M, López-García X. Fighting deepfakes: media and internet giants' converging and diverging strategies against Hi-tech misinformation. Media Commun. 2021;9:291–300.
27. Tran VN, Lee SH, Le HS, Kwon KR. High performance deepfake video detection on CNN-based with attention target-specific regions and manual distillation extraction. Appl Sci. 2021;11:7678.
28. Jiang B, Chen S, Wang B, Luo B. MGLNN: semi-supervised learning via multiple graph cooperative learning neural networks. Neural Netw. 2022;153:204–14.
29. Roy AM, Bhaduri J, Kumar T, Raj K. WilDect-YOLO: an efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection. Ecol Inform. 2023;75: 101919.
30. Hu J, Liao X, Gao D, Tsutsui S, Wang Q, Qin Z, et al. Mover: mask and recovery based facial part consistency aware method for deepfake video detection. 2023. https://arxiv.org/abs/2305.05943v1. Accessed 31 Aug 2023.
31. Hussain ZF, Ibraheem HR. Novel Convolutional neural networks based Jaya algorithm approach for accurate deepfake video detection. Mesopotamian Journal of CyberSecurity [Internet]. 2023 [cited 2023 Aug 31];2023:35–9. https://journals.mesopotamian.press/index.php/CyberSecurity/article/view/58
32. Xia Z, Qiao T, Xu M, Wu X, Han L, Chen Y. Deepfake video detection based on MesoNet with preprocessing module. Symmetry. 2022;14:939.
33. FF++ Dataset : https://github.com/ondyari/FaceForensics. Accessed 29 Nov 2023.
34. DFDC Dataset : https://ai.meta.com/datasets/dfdc/. Accessed 29 Nov 2023.
35. Celeb-DF Dataset : https://github.com/yuezunli/celeb-deepfakeforensics. Accessed 29 Nov 2023.

Publisher's Note