



Audio copy-move forgery detection with decreasing convolutional kernel neural network and spectrogram fusion

Canghong Shi¹ · Xin Qiu¹ · Min Wu¹ · Xianhua Niu¹ · Xiaojie Li² · Sani M. Abdullahi³

Received: 25 March 2025 / Accepted: 28 June 2025 / Published online: 12 July 2025
© The Author(s) 2025

Abstract

One of the most common forms of audio forgery is copying and moving certain audible segments of audio to other locations in the same audio. The audio features of the pasted regions in such audio forgeries become very dissimilar to the audio features of the copied segments after post-processing. This dissimilarity makes detecting such tampering a major challenge. To address this problem, we propose a robust audio copy-move forgery detection method using a Decreasing Convolutional Kernel Neural Network (DCKNN), data augmentation, and digital fusion. In the proposed algorithm, Mel spectrogram and Hilbert–Huang spectrogram of the audio are extracted, and then they are fused by weighting coefficients, which are gained through extensive experiments. New spectrogram images are generated by weighted fusion, and these spectrogram images are used to train the proposed DCKNN model. The trained DCKNN can effectively detect copy-move forgery. The DCKNN model consists of a combination of four convolutional groups, each with different sensitivities to the two audio categories. We solve the problem of different sensitivities by sequentially lowering the parameters of the convolutional layers in the four convolutional groups, thus obtaining high accuracy in audio classification. The experimental results show that the proposed scheme is robust to most typical post-processing operations, including additive noise, compression, median filtering, resampling, re-quantization, and low-pass filtering, etc al. In addition, our method shows better performance in the detection of forged audio with multiple attacks. Compared to the state-of-the-art algorithms, the proposed algorithm has advantages in terms of accuracy, precision, and F1 score.

Keywords Audio copy-move-forgery detection · Convolutional neural network · Figure fusion · Mel spectrogram · Hilbert–Huang spectrogram

Introduction

With the rapid development of science and technology, audio has become an integral part of our daily lives. Currently, the most widely used audio editing software, such as Adobe Audition CC allows people without specialized knowledge to modify audio. Since the software can perform a series of post-processing operations (adding Gaussian white noise, audio compression, median filtering, resampling, re-quantization,

and low-pass filtering) to cover up traces of forgery, it makes it difficult to detect the authenticity of audio. Especially when audio is used as evidence in court, ensuring its authenticity becomes crucial. For example, audio used as evidence in court contains the phrase: “I will not say it again, I’m a criminal.” A forger could alter the audio to read “I will not say it again, I am not a criminal.” by copying the “not” part and purposefully placing it between “I” and “am” making a complete semantic change. Figure 1 depicts an example of a copy-move forgery operation [1, 2], the forged audio is created by copying the first segment of the original audio and moving it to the sixth segment (highlighted in red in the figure). Figure 1a, b compares the original and forged audio waveforms.

The challenge of detecting copy-move forgery arises due to the interception of often short-duration segments from the same audio file. Furthermore, the forger employs various post-processing operations on the tampered audio, including

✉ Xiaojie Li
lixiaojie000000@163.com

¹ School of Computer and Software Engineering, Xihua University, Chengdu 610039, Sichuan, China

² College of Computer Science, Chengdu University of Information Technology, Chengdu 610225, Sichuan, China

³ Maersk McKinney Moller Institute, University of Southern Denmark, Campusvej, 5230 Odense, Denmark

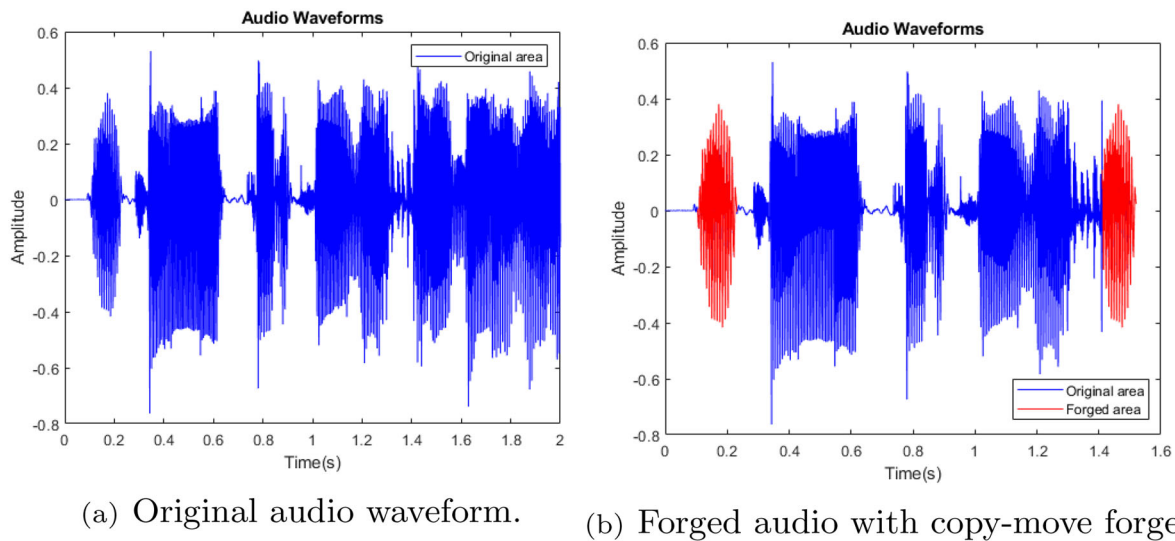


Fig. 1 Copy-move audio comparison figure

the addition of noise, median filtering, compression, low-pass filtering, re-quantization, resampling, multiple attacks, and so on. Figure 2 shows the waveforms of the tampered audio after different attacks.

In order to test the authenticity of the audio, the researcher conducted the study from several angles. For example, the identification of microphones of recording devices [3], embedding digital watermarks for copyright protection [4, 5].

However, research in the area of audio replication mobile forgery has been slightly slower. Early approaches to detecting copy-move forgeries relied on manual segmentation and handcrafted features. For example, calculating the similarity in the same window to determine whether a segment is a forgery is less robust as the audio waveforms are distorted when subjected to post-processing attacks [6], resulting in inconsistencies in the waveforms of the copy-move segments. Methods based on pitch and resonance peaks or formant structure require Voice Activity Detection (VAD) to segment audio before feature extraction [7–10], making them vulnerable to unsegmented or weakly voiced tampering. Other techniques utilized Mel spectrograms [11, 12] or Constant-Q features combined with handcrafted similarity measures [13–15], but often suffered from high computational costs or sensitivity to signal degradation.

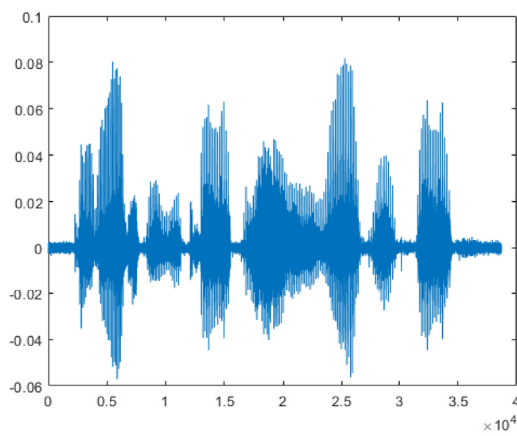
To overcome these limitations, recent studies have turned to deep learning [16, 17] and image-based representations of audio. By transforming audio into time-frequency spectrograms—such as Mel, Constant-Q Transform (CQT), or Hilbert–Huang transforms—researchers can leverage mature image analysis techniques, including CNNs [18] and Transformers [19], for forgery detection. However, many of these models either ignore the complementary nature of different

spectrogram types or rely on fixed convolutional structures, limiting their robustness under hybrid attack conditions [20].

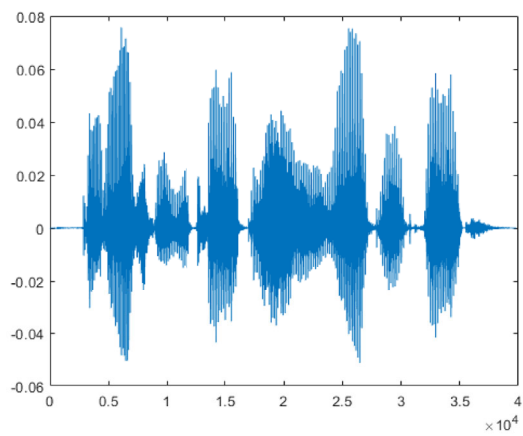
In this work, we propose a novel segmentation-free audio copy-move forgery detection framework based on Decreasing Convolutional Kernel Neural Networks (DCKNN). To enhance feature richness and resilience, we fuse Mel and Hilbert–Huang spectrograms, which offer complementary time-frequency characteristics. Unlike existing CNN-based models, our DCKNN architecture employs progressively shrinking convolutional kernels and a variety of activation and pooling strategies. This design improves the model's ability to generalize across different forgery types while maintaining a compact parameter.

The proposed method is tested for performance under multiple datasets and compared with mainstream methods [10, 11, 13, 15, 18] in the field. The contributions of the proposed method are summarized below.

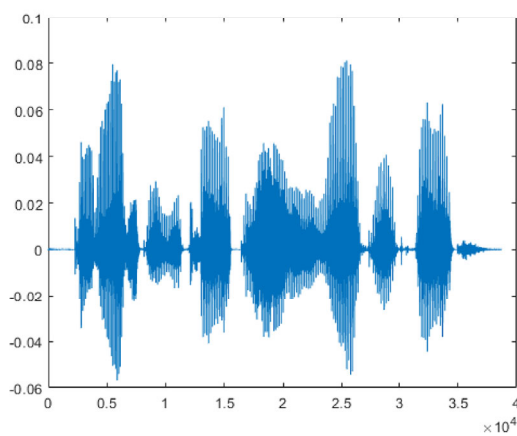
- (i) We propose the DCKNN model, where the convolutional kernels range from large to small, allowing the network to extract better global features using large convolutional kernels in the face of audio with post-processing. Moreover, the DCKNN model uses fewer parameters than the model proposed by [18] with the same number of layers.
- (ii) We propose an image fusion method for the network feature map input, where two spectral images are weighted and fused. This method highlights important information on the spectral images while filtering out unimportant details.
- (iii) We used two publicly available audio datasets to test the performance of the DCKNN model and compare it to other methods. From the comparison results, the pro-



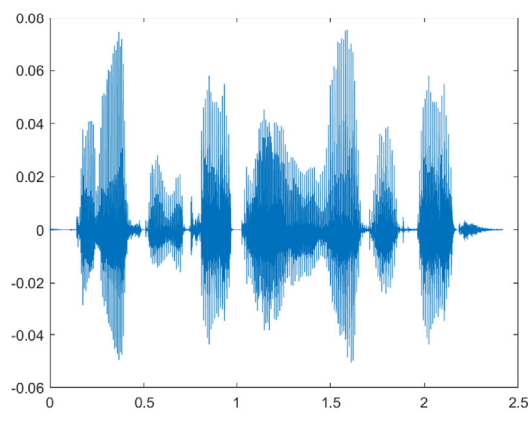
(a) With 20dB Gaussian noise.



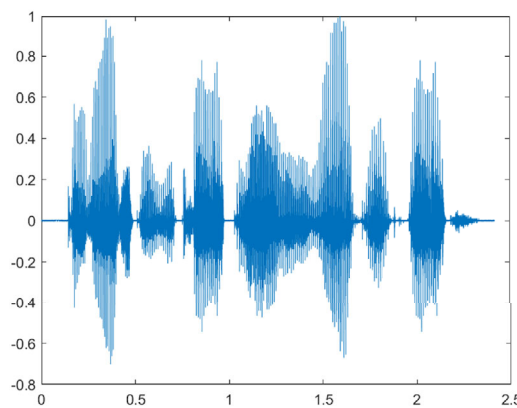
(b) With 32kbps MP3 compression.



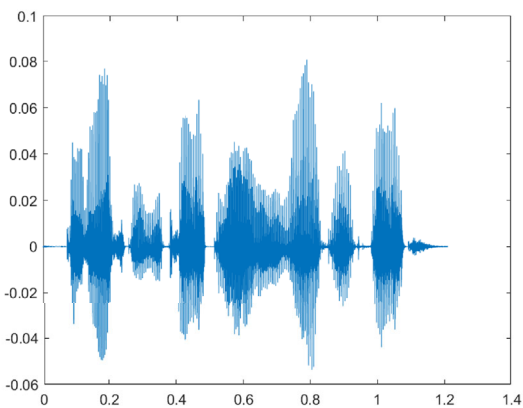
(c) With median filtering.



(d) With low-pass filtering.



(e) With re-quantization.



(f) With resampling.

Fig. 2 Waveforms of the same audio tampered with under different attacks. **a** Shows the forged audio with 20 dB Gaussian white noise added. **b** Shows the forged audio with a median filtering attack. **c** Shows forged audio with a 32 kbps MP3 compression attack. **d** Shows the

forged audio with a low-pass filtering attack. **e** Shows the forged audio with a re-quantization attack. **f** Shows the forged audio attacked by resampling

posed method outperforms the traditional methods and other neural network models.

Related work

In this section, we introduce the related techniques used in the experiments: figure fusion, convolutional neural networks, and data augmentation.

Analysis of related work

Recent research in audio copy-move forgery detection has progressed along multiple methodological streams.

In the available traditional methods, Yan [10] proposed a method based on pitch and resonance peak extraction from voiced segments. Using Dynamic Time Warping (DTW) to compute similarity, this approach is effective for clean signals. However, it relies heavily on accurate VAD and segmentation, making it vulnerable to unsegmented forgeries and degradation from post-processing. Huang [7] employed audio activity detection algorithms to distinguish between voiced and unvoiced segments within the audio. Subsequently, audio features were extracted using the Discrete Fourier Transform (DFT) and Mel-frequency cepstral coefficients (MFCC). They organized each detail based on this data and conducted comparisons between each segment and its neighboring regions in the list to identify potential instances of audio copy-move forgery. However, both algorithms require a segmentation between voiced and unvoiced segments prior to detection. In addition, Su [13] introduced a segmentation-free method using Constant Q Cepstral Coefficients (CQCC) and a sliding window mechanism. By comparing windowed feature vectors, this method improves over segmentation-dependent approaches. However, its exhaustive pairwise comparison results in a high computational cost. To solve the problem of detection duration and segmentation of audible segments, some researchers have introduced machine learning algorithms into the field.

In the machine learning algorithm, Xie [14] proposed to first perform VAD on the audio to extract multiple features of the voiced segments, and then fuse the detection results of the extracted gamma-pass features, Mel cepstrum coefficients features, fundamental features, and DFT coefficients features using a C4.5 decision tree to achieve forgery detection of the audio. Su [15] first subjected the audio to VAD, extracted Constant Q Spectral Sketches (CQSS) features, classified copy-move forgeries by using Support Vector Machines, and optimised the features using Genetic Algorithms. The final trained support vector machine then implements the audio for forgery detection. This method is effective in categorising audio and approximating areas of audio forgery. However, it requires manual selection of features for training, and the

effectiveness of this training is particularly dependent on the selection of features. The machine learning approach does avoid the problems of audible segment segmentation and window length, but it manually selects features, and this approach fails when the features are not selected correctly. In order to explore new methods, some researchers have introduced traditional methods in the field of image and deep learning methods to the field of audio forgery detection.

In the traditional approach in the field of image, Ustubioglu [11] proposed a keypoint detection-based algorithm utilizing Scale Invariant Feature Transform (SIFT) key points is employed to assess the authenticity of the audio Mel spectrogram. The process involves creating Mel spectrograms from the audio, obtaining SIFT key points from each RGB color channel, and subsequently performing feature matching on the key points acquired from each channel. The identified key points are then designated as image sub-blocks, representing forged blocks. Similarly, Beste [12] proposed to transform the audio into a super-resolution spectrogram, which uses Center Surround Extrema to obtain the key points, binary robust independent elementary features to represent the key points, and then a sorted point identification clustering structure algorithm for keypoint matching classification. Both of the above methods can be applied to audio forgery detection, but this method also brings the problem of excessive time-consuming detection. In order to address the problem of excessive time-consuming audio forgery detection using images, researchers have invoked image classification networks in the image domain.

In deep learning algorithms, Arda [18] proposed a CNN-based model that uses Mel spectrograms to classify audio as authentic or forged. While CNNs exhibit strong feature extraction capabilities, this model lacks spectrogram-level feature fusion and is limited by a static network architecture, reducing its effectiveness under complex post-processing scenarios. In this section, we provide a focused analysis of these representative methods, which are also included in our experimental comparisons. Similarly, there are Vision Transformer [19] and Capsule Networks [17]. However, the use of feature maps in these deep learning methods is singular. A single feature map is not guaranteed to be effective against multiple types of attacks.

To address this problem, we propose a robust audio copy-move forgery detection method using Decreasing Convolutional Kernel Neural Networks (DCKNN), data augmentation, and spectrogram fusion.

Figure fusion

Figure fusion, a technique aimed at combining two or more distinct images into a unified representation, is a process crucial for consolidating information from multiple sources into a single cohesive image. Currently, there are several methods

of image fusion as follows [21]: Pixel-level fusion involves the fusion of pixel values at the same location from different images using various methods such as averaging, weighted averaging, or selecting the maximum value. Feature-level fusion involves extracting features from each image and then combining them. Typically, the combined features include texture, color, and edges. Each image is divided into several regions based on image region fusion. Different fusion methods are used to select regions for fusion. Based on deep learning fusion, a deep learning model is used to learn feature representations from the input feature graph, and the learned feature representations are fused. Figure fusion is applied to a wide range of fields and plays an extremely important role in the medical field, the military field, and the artistic field.

The rationale for transforming audio signals into fused spectrograms lies in their complementary robustness to post-processing. Mel spectrograms excel in frequency localization under noise [18], while Hilbert–Huang spectrograms decompose signals into intrinsic time-frequency components via Empirical Mode Decomposition (EMD). However, Mel spectrograms may overlook transient phase changes, and Hilbert–Huang spectrograms suffer from mode mixing in complex signals [20]. By fusing these representations with weighted coefficients (“[Proposed approach](#)”), our method preserves global frequency patterns and local time-frequency dynamics, enabling DCKNN to detect subtle tampering traces even under hybrid attacks.

In this study, we apply pixel-level fusion of images for audio authentication. The exact process of pixel-level fusion is shown in Fig. 3. It can be seen that we are using two images to generate a new image by weighted summing. The new image retains the important information of the two images according to different weights. This method has good robustness and resistance to post-processing attacks and has significant advantages in audio authenticity detection [22, 23].

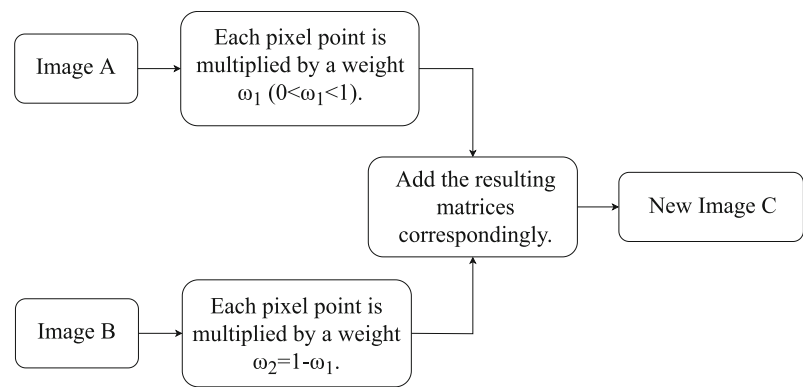
Convolutional neural network

Convolutional neural networks (CNNs) are widely used for image classification and recognition due to their weight sharing and hierarchical feature extraction capabilities [24]. While earlier CNN models faced interpretability challenges, recent advances in explainable AI (XAI) techniques—such as gradient-based class activation mapping (Grad-CAM) [25, 26] and self-attention visualization [27]—provide transparent insights into feature learning mechanisms. These methods enable the visualization of discriminative regions in input spectrograms, bridging the gap between model decisions and human interpretability. A typical CNN architecture comprises convolutional layers, pooling layers, activation functions, and normalization operations.

The convolutional model proposed by [18] forged audio copy movement is an infinite stack of convolutional and pooling layers; such a network model would make the network parameters large and suffer from under-training in the face of small sample training. And, since the network model only has convolutional and pooling layers, this will lead to a weak generalisation of the network. The use of capsule networks proposed by [19] and the use of vision Transformer networks proposed by [17] still suffer from the problem of a large number of training parameters, even though they introduce a variety of nonlinear transformations. Compared to the above three network models, our proposed model not only introduces multiple nonlinear transformations to enhance the generalisation ability of the model but also reduces the number of training parameters by decreasing the convolution kernel sequentially. Moreover, the proposed network model can be seen to be feasible through experiments.

Data augmentation

In recent years, data augmentation techniques have become a key strategy for mitigating the scarcity of labeled data and enhancing model generalization in audio and speech recognition research [28]. Salamon and Bello [29] pioneered the application of audio data augmentation (e.g., time shifting, pitch shifting, and time stretching) to environmental sound classification. By augmenting the training set with deformed log-Mel spectrograms and combining CNN and RNN models, they significantly improved the performance of deep convolutional neural networks. Rezaul et al. [30] integrated augmentation techniques into the classical MFCC feature extraction pipeline. Employing various data preprocessing and augmentation operations on small audio datasets—such as random cropping and background noise addition—in conjunction with CNN and RNN models, they substantially enhanced the robustness and accuracy of music and environmental sound classification. These augmentation methods not only expand the dataset size but also increase feature input diversity. For sound recognition, Hizlisoy et al. [31] extended the original audio using augmentation and combined multiple low-level features (e.g., MFCC, spectral contrast) to construct hybrid feature vectors for classifier input. The augmented data significantly enhanced the representational capacity of the training data, improving its discriminative power for diverse sounds and leading to higher test set accuracy. In language recognition, Alemu et al. [32] augmented an original 8-hour audio dataset to approximately 24 h using techniques like volume perturbation and time shifting, based on a CNN model. By combining Mel spectrograms with MFCC hybrid features, they achieved higher accuracy during both training and validation phases for languages such as Amharic, Geez, and Tigrigna. Within the domain of audio forgery detection, Üstübioglu et al. [18] employed a

Fig. 3 Schematic of weighted fusion

Mel spectrogram-based CNN approach. They augmented the training data to simulate multiple copy-paste and other tampering scenarios, thereby improving the model's sensitivity and discriminative capability for detecting audio copy-paste forgeries.

Collectively, these studies demonstrate that data augmentation effectively exposes models to broader sample variations, enhancing feature learning capabilities. This technique successfully addresses two critical challenges: model overfitting from limited data and reduced generalization capacity due to class imbalance.

Proposed approach

The objective of our work is to develop a model capable of efficiently detecting audio copy-move forgery, which consists of a feature fusion and a sequential DCKNN. This section provides a detailed description of our proposed algorithm. Firstly, we fuse different spectrograms of the same audio using image weighting to create a new image, which is then used as the input feature map for the DCKNN. To enhance the generalization ability of the network, we introduce various nonlinear variations in this convolutional neural network, and we have designed the convolutional kernel part to sequentially decrease in size, allowing the network to achieve a balance of sensitivity between the two labels.

Spectrogram feature generation

Mel spectrogram feature generation

In this proposed approach, the Fast Fourier Transform (FFT) was used to generate the Mel spectrogram of the audio signal. Firstly, the audio was pre-emphasised to highlight the high-frequency part of the audio, then the FFT was performed on the audio, and the transformed signal was passed through the Mel filter bank. The equation for the interconversion of Mel frequency to linear frequency is shown in Eq. (1).

$$Mel(f) = 2595 \times \lg \left(1 + \frac{f}{700} \right) \quad (1)$$

where f is the frequency of the audio. The result obtained by Eq. (1) is subjected to logarithmic and Discrete Cosine Transform (DCT). Finally, the Mel-spectrogram image is obtained.

The audio ("sa2_3-5.wav") from the TIMIT audio database is firstly subjected to various post-processing, and then the Mel spectrogram images after various attacks are obtained by the appeal method. Figure 4 compares Mel spectrograms under different post-processing attacks. The red boxes highlight forged segments, demonstrating minimal distortion despite global noise interference.

In Fig. 4, we can see that even after various post-processing operations on the audio, the similarity of the repetitive parts of the Mel spectrogram remains high. This result highlights the robustness of the Mel spectrogram and its ability to withstand post-processing attacks.

Generation of the Hilbert–Huang spectrogram

This paper employs Empirical Mode Decomposition (EMD) and the Hilbert Transform to derive the Hilbert–Huang spectrogram. The process is outlined as follows:

Firstly, the audio signal $x(t)$ is decomposed into a set of Intrinsic Mode Function (IMF) components and a residual term using EMD, as shown in Eq. (2):

$$x(t) = \sum_{i=1}^n C_i(t) + R_n(t) \quad (2)$$

Where $C_i(t)$ denotes the i -th IMF component, and $R_n(t)$ represents the residual term.

Next, the Hilbert Transform is applied to each IMF component $C_i(t)$ to obtain its instantaneous amplitude and instantaneous phase, resulting in the Hilbert–Huang spectrogram, as described in Eq. (3):

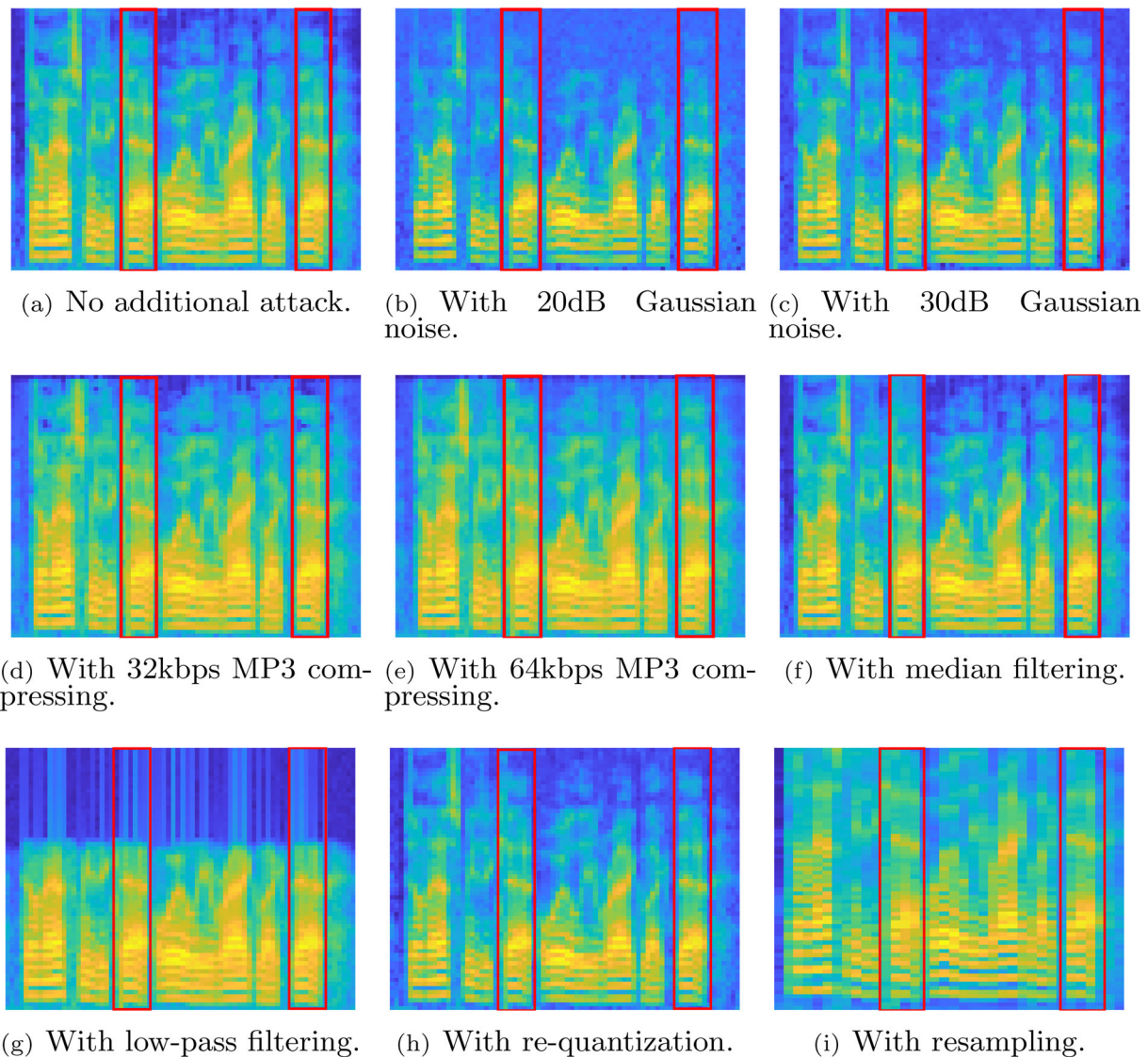


Fig. 4 Comparison of Mel spectra with various post-processes. **a** Shows the Mel spectrogram that forged audio is not under attack. **b** Shows the Mel spectra of audio attacked by 20 dB Gaussian white noise. **c** Mel spectra of audio attacked by 30 dB Gaussian white noise are shown. **d** Shows the Mel spectrum of audio attacked by 32 kbps MP3 compression. **e** Shows the Mel spectrum of an audio attack by 64 kbps

MP3 compression. **f** Shows the Mel spectrum of an audio attack by median filtering. **g** Shows the Mel spectrum of audio subjected to a low-pass filtering attack. **h** Shows the Mel spectrum of audio attacked by re-quantization. **i** Shows the Mel spectrum of an audio attacked by resampling

$$z_i(t) = a_i(t)e^{j\theta_i(t)} \quad (3)$$

Where $a_i(t)$ denotes the instantaneous amplitude, $\theta_i(t)$ denotes the instantaneous phase, and j is the imaginary unit. Through the Hilbert Transform, the complex signal $z_i(t)$ is derived, where:

$$a_i(t) = |z_i(t)| = \sqrt{C_i(t)^2 + \hat{C}_i(t)^2} \quad (4)$$

$$\theta_i(t) = \arctan\left(\frac{\hat{C}_i(t)}{C_i(t)}\right) \quad (5)$$

Here, $\hat{C}_i(t)$ represents the Hilbert Transform of $C_i(t)$.

Finally, the Hilbert–Huang spectrograms of all the IMF components are merged to produce the final Hilbert–Huang spectrogram. This comprehensive spectrogram captures the time-frequency characteristics of the audio signal, making it useful for forgery detection tasks.

We first perform various post-processing operations on a piece of audio ("sa2_3-5.wav") in the TIMIT database, and then obtain the Hilbert–Huang spectrogram images of each post-processing operation using the appeal method. Figure 5 shows the Hilbert–Huang spectrogram image of the same

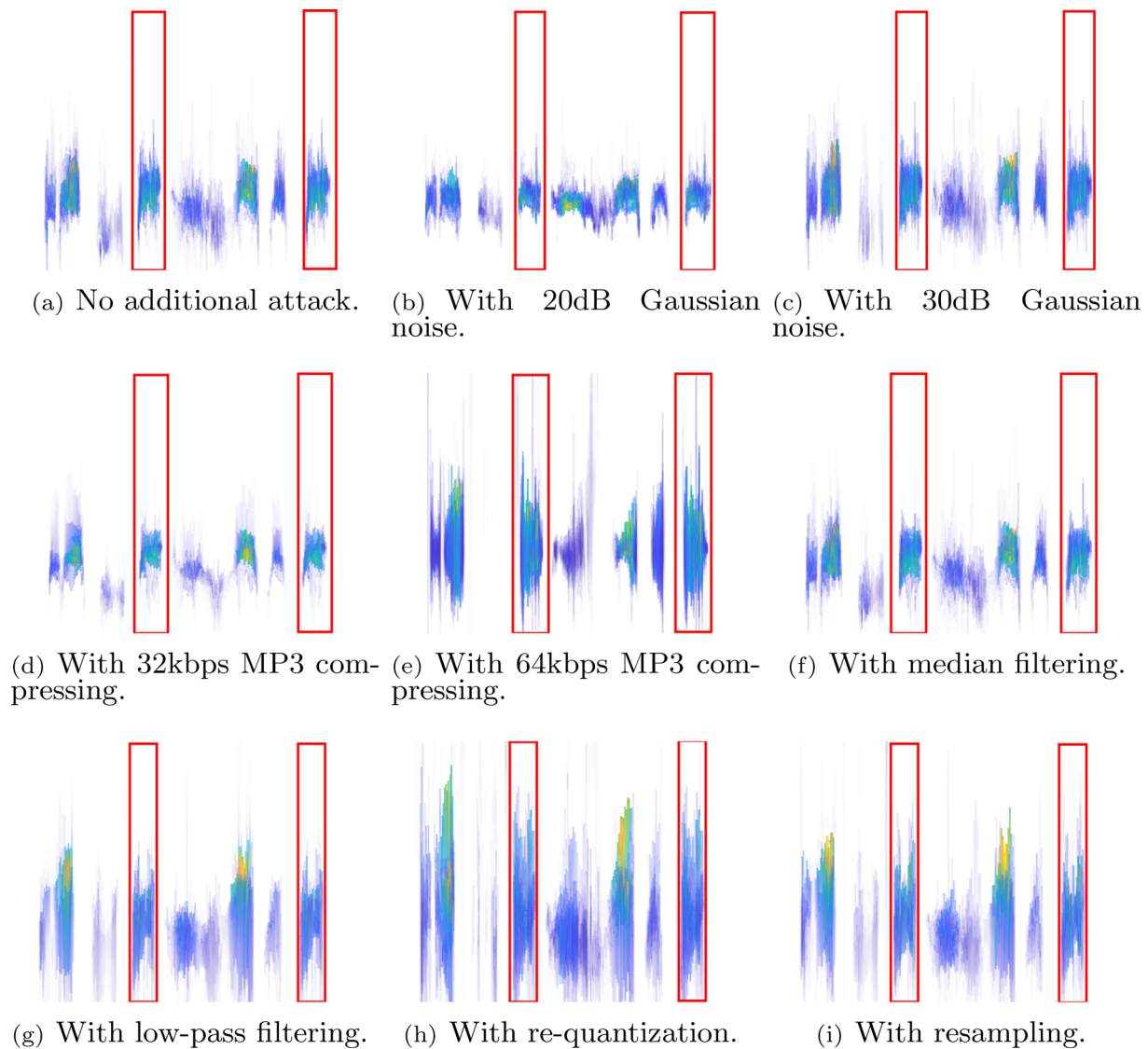


Fig. 5 Comparison of Hilbert–Huang spectra with various post-processes. **a** Shows the Hilbert–Huang spectrogram of forged audio not under attack. **b** Hilbert–Huang spectrogram showing audio attacked by 20 dB Gaussian white noise. **c** Hilbert–Huang spectrogram showing audio attacked by 30 dB Gaussian white noise. **d** Shows the Hilbert–Huang spectrum of audio attacked by 32 kbps MP3 compression. **e** The

Hilbert–Huang spectrum of an audio attacked by 64 kbps MP3 compression is shown. **f** Shows the Hilbert–Huang spectrum of an audio attack by median filtering. **g** Shows the Hilbert–Huang spectrum of audio subjected to a low-pass filtering attack. **h** Shows the Hilbert–Huang spectrum of audio attacked by re-quantization. **i** Shows the Hilbert–Huang spectrum of an audio attack by resampling

audio under different attacks. The forged parts are marked with red rectangles.

From Fig. 5, we can see that even after various post-processing operations on the audio, the similarity of the repetitive parts of the Hilbert–Huang spectrogram remains high. This observation highlights the robustness of the Hilbert–Huang spectrogram and its ability to withstand post-processing attacks.

Spectrogram figure fusion

The Mel spectrogram and Hilbert–Huang spectrogram of audio are chosen for image fusion in this paper. From Fig. 4, it can be seen that in the presence of an attack, the change in the replicated moving segment is almost minimal. This indicates that the Mayer spectrogram has strong robustness. The Hilbert Huang spectrogram decomposes the signal into

time-frequency localized components, which can well reflect the dynamic changes of the signal in the time-frequency domain. The two spectrograms have different characteristics, and image fusion, can make the new image have its unique characteristics, which makes the new image we obtain have stronger robustness. The specific steps of image fusion are described in detail below.

Firstly, the two figures F_1 and F_2 are resized to the same dimensions, following which image fusion is conducted using Eq. (6).

$$F_{final}(x, y) = w_1 F_1(x, y) + w_2 F_2(x, y) \quad (6)$$

where w_1 is the weight of the first spectrogram and w_2 is the weight of the second spectrogram. F_1 is the first spectrogram and F_2 is the second spectrogram. (x, y) is the corresponding position of the image pixel. F_{final} is the spectral image obtained after the fusion.

The two weights add up to one. The larger the weight corresponding to the image, the greater the effect on the new image. We perform a large number of experiments with different weight combinations to finally determine an optimal weight combination.

The audio ("sa2_3-5.wav") was selected from the TIMIT database, and its Mel spectrogram and Hilbert–Huang spectrogram were fused to produce a new spectrogram using the weighted fusion method. Figure 6 illustrates the fusion results with different weights, represented as (a) $w_1 = 0.6$, $w_2 = 0.4$, (b) $w_1 = 0.7$, $w_2 = 0.3$, (c) $w_1 = 0.8$, $w_2 = 0.2$, and (d) $w_1 = 0.9$, $w_2 = 0.1$, respectively.

From Fig. 6 we can see that the effect of the two spectrograms is different under different weight combinations. It can be seen that the Hilbert–Huang spectrogram can highlight important parts of the Mel spectrogram. Where for the w combinations, we performed a number of experiments to find the optimal w . The w combination selection process is explained in the next subsection.

This method highlights the important features that are common to both spectra and retains the different important features through figure fusion. The obtained figure has multiple important features, making it robust and resistant to attacks.

Weights selection

In this section, we show how we choose the right combination of weights. With different weights, the proposed method has different performance under different attacks. Therefore, choosing the right combination of weights is beneficial for our subsequent detection. We perform experiments with different post-processing attacks under different weight combinations. We use the trained model with appropriate weights

to classify the audio data in the TIMIT database, as shown in Fig. 7.

As shown in Fig. 7, the network performance of the proposed method in this paper is better when the weights of the Mel spectrogram become larger and larger under different combinations of weights. However, when the weight of the full Mel spectrogram is 1 and the weight of the Hilbert–Huang spectrogram is 0, the network's overall performance decreases. Thus, this proposed image fusion technique proves to be useful. Different combinations of w make the network exhibit different performances under different attacks. As shown in Fig. 7, when detecting the attack mode as additive noise, using the weight set of $w_1 = 0.7$ and $w_2 = 0.3$ results in an F1 score of 94% and an accuracy rate of 93%. To detect attacks using median filtering, a weight combination of $w_1 = 0.8$ and $w_2 = 0.2$ can achieve an F1 score of 90% and an accuracy of 91%. For detecting compressed audio attacks, a weight combination of $w_1 = 0.9$ and $w_2 = 0.1$ can achieve an F1 score and accuracy rate of 95%. When the audio is not subjected to post-processing attacks, using a weight combination of $w_1 = 0.8$ and $w_2 = 0.2$ results in a recall of 81% and the remaining metrics can reach 90%. In summary, if a large number of attacks of a certain type are expected to be detected, a more appropriate weight combination can be chosen based on the type of attack. If unsure which type of attack will be detected, the weight combination of $w_1 = 0.8$ and $w_2 = 0.2$, which is stronger in terms of synthesis, can be used. Therefore, after performing a comprehensive analysis, we use the weight combination of $w_1 = 0.8$ and $w_2 = 0.2$ for our experiments.

Model parameter setting

Architecture overview

Audio copy-move-paste forgery detection is a binary classification problem for determining whether a piece of audio is forged or original. To solve the audio forgery detection problem, [18] proposed network model appears to classify the original audio as forged audio if the original audio is added in the prediction phase. This is related to the input feature figure, a single network structure with large parameters and weak generalisation. Therefore, as shown in Fig. 8, we propose to use a new feature map, which consists of the fusion of two spectrograms, which makes the new feature map more robust, and input it into the proposed convolutional model with image weighted fusion method. The model consists of four multi-module convolutional groups of the network model. Among the four convolutional group network modules, our core is to make the convolutional kernel decrease sequentially to reduce the training parameters. The specific details of the DCKNN model reduction are shown in Fig.

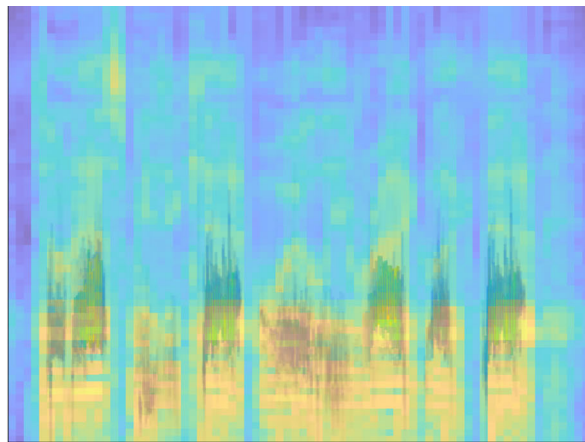
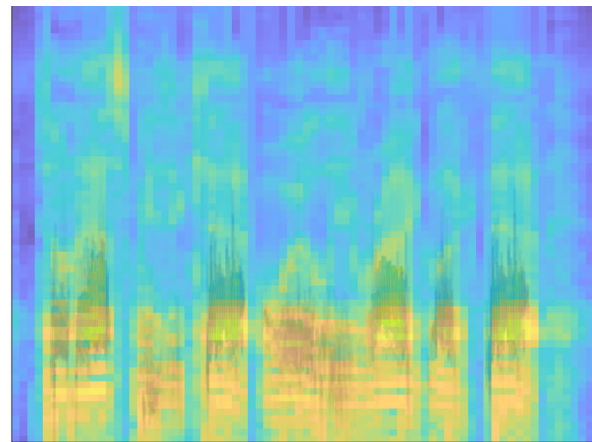
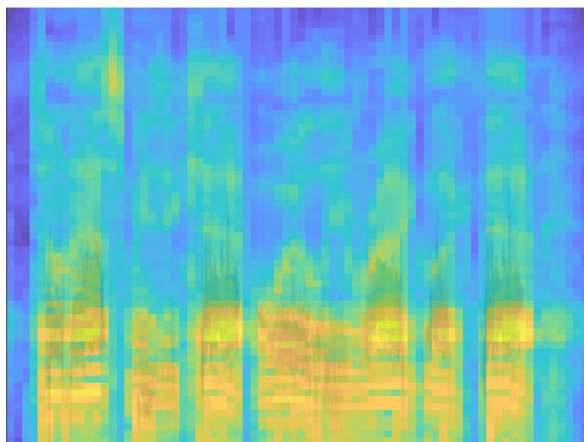
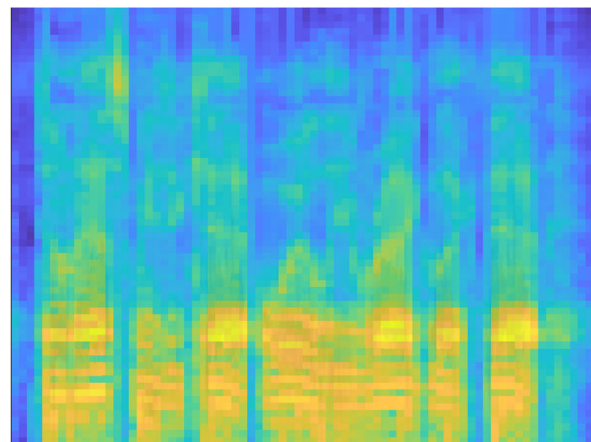
(a) $w_1 = 0.6, w_2 = 0.4$ (b) $w_1 = 0.7, w_2 = 0.3$ (c) $w_1 = 0.8, w_2 = 0.2$ (d) $w_1 = 0.9, w_2 = 0.1$

Fig. 6 Viewable view of fusion with different weights. **a** Shows the fusion result with fusion weights of $w_1 = 0.6, w_2 = 0.4$. **b** Shows the fusion result with fusion weights of $w_1 = 0.7, w_2 = 0.3$. **c** Shows the

fusion results with fusion weights of $w_1 = 0.8, w_2 = 0.2$. **d** Shows the fusion results with fusion weights of $w_1 = 0.9, w_2 = 0.1$

9. In order to improve the generalisation of the network, we introduce a variety of nonlinear variations.

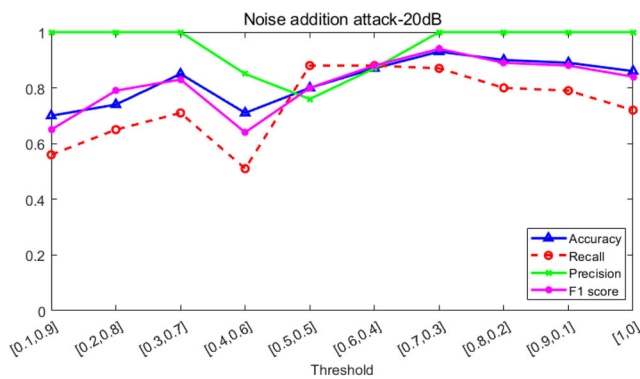
Given a fused spectrogram image $F \in 3 \times H \times W$, feature extraction of the feature map is performed by the first convolutional block, which is used as input to the second convolutional block. After passing the second convolutional block, the network will already be able to correctly classify the original audio, but will incorrectly classify the fake audio as original. Therefore, we introduced the third and fourth convolutional blocks to further reduce the convolutional kernel to make the sensory field smaller, thus further extracting the local feature information in the feature map. The results of the fourth convolutional group are then fed into the fully connected module to obtain the category probabilities through the softmax activation function. Finally, the category proba-

bilities are inputted into the classifier to correctly categorize the two types of audio.

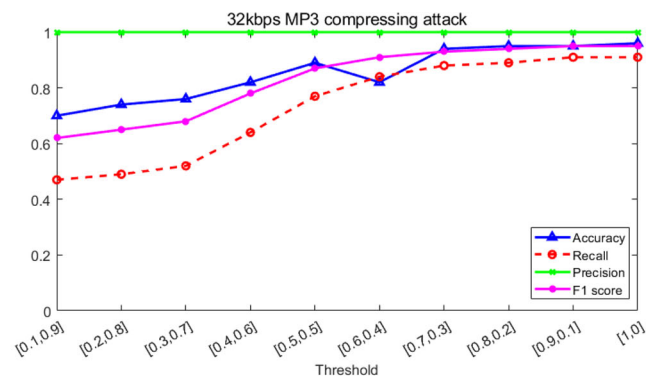
Core network module

From a given fused spectrogram image I , it is sequentially subjected to convolution, pooling and non-linear operations to generate a feature map containing texture-related features. The convolution kernels we use are sequentially decreasing, which allows the sense field of the convolutional layers to go from large to small and from global to local. At the same time, the number of filters is incrementally increased to produce more texture-related feature maps.

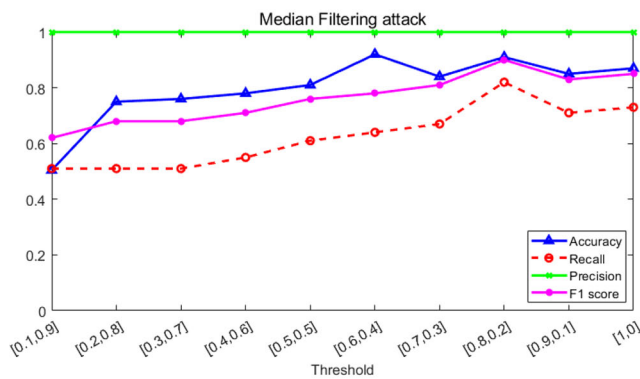
Specifically, the given image is passed through the first convolutional block to extract the feature texture map as



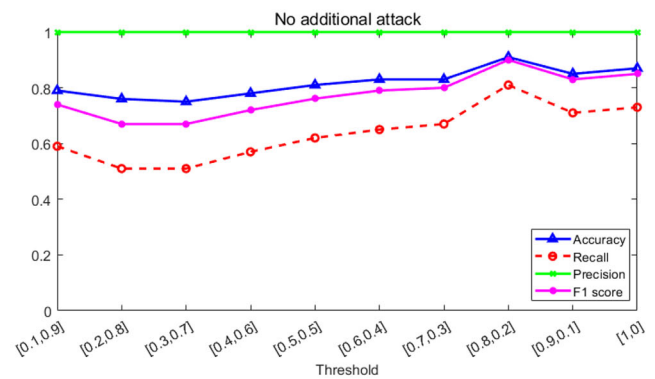
(a) With 20dB Gaussian white noise.



(b) With 32kbps MP3 compressing.



(c) With Median filtering.



(d) No additional attack.

Fig. 7 Performance comparison of networks trained using fused spectrograms with different weights. **a** Effect of different fusion weights on network performance under 20 dB Gaussian white noise. **b** Demonstrates the effect of different fusion weights on the network performance

under a 32 kbps MP3 compression attack. **c** Demonstrates the effect of different fusion weights on network performance under a median filtering attack. **d** Demonstrates the impact of different fusion weights on network performance under no attack

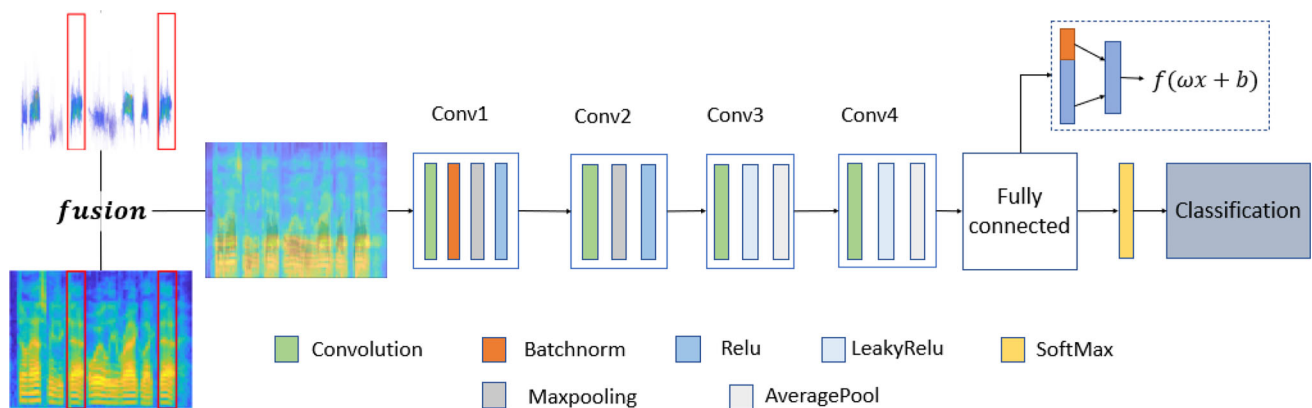


Fig. 8 The proposed DCKNN model structure diagram. It consists of another image fusion module and four convolutional groups. Image fusion combines the input Mel spectrogram and Hilbert–Huang spec-

trogram with corresponding weights. The convolution kernels of the four convolutional groups are sequentially reduced and finally input to the fully connected layer to complete the classification task

#	Name	Type	Activations	Learnable Properties	States
1	imageinput 300×300×3 images with 'zero-center' normalization	Image Input	300(S) × 300(S) × 3(C) × 1(B)	-	-
2	conv 10 10×10 convolutions with stride [3 3] and padding 'same'	2-D Convolution	100(S) × 100(S) × 16(C) × 1(B)	Weights 10 × 10 × 3 × 16 Bias 1 × 1 × 16	-
3	batchnorm Batch normalization	Batch Normalization	100(S) × 100(S) × 16(C) × 1(B)	Offset 1 × 1 × 16 Scale 1 × 1 × 16	TrainedMean 0 × 0 TrainedVariance 0 × 0
4	maxpool 5×5 max pooling with stride [1 1] and padding 'same'	2-D Max Pooling	100(S) × 100(S) × 16(C) × 1(B)	-	-
5	relu ReLU	ReLU	100(S) × 100(S) × 16(C) × 1(B)	-	-
6	conv_1 32 7×7 convolutions with stride [2 2] and padding 'same'	2-D Convolution	50(S) × 50(S) × 32(C) × 1(B)	Weights 7 × 7 × 16 × 32 Bias 1 × 1 × 32	-
7	maxpool_1 5×5 max pooling with stride [1 1] and padding 'same'	2-D Max Pooling	50(S) × 50(S) × 32(C) × 1(B)	-	-
8	relu_1 ReLU	ReLU	50(S) × 50(S) × 32(C) × 1(B)	-	-
9	conv_2 64 5×5 convolutions with stride [1 1] and padding 'same'	2-D Convolution	50(S) × 50(S) × 64(C) × 1(B)	Weights 5 × 5 × 32 × 64 Bias 1 × 1 × 64	-
10	leakyrelu Leaky ReLU with scale 0.01	Leaky ReLU	50(S) × 50(S) × 64(C) × 1(B)	-	-
11	avgpool2d 5×5 average pooling with stride [1 1] and padding 'same'	2-D Average Pooling	50(S) × 50(S) × 64(C) × 1(B)	-	-
12	conv_3 128 3×3 convolutions with stride [1 1] and padding 'same'	2-D Convolution	50(S) × 50(S) × 128(C) × 1(B)	Weights 3 × 3 × 64 × 128 Bias 1 × 1 × 128	-
13	leakyrelu_1 Leaky ReLU with scale 0.01	Leaky ReLU	50(S) × 50(S) × 128(C) × 1(B)	-	-
14	avgpool2d_1 5×5 average pooling with stride [1 1] and padding 'same'	2-D Average Pooling	50(S) × 50(S) × 128(C) × 1(B)	-	-
15	fc_1 32 fully connected layer	Fully Connected	1(S) × 1(S) × 32(C) × 1(B)	Weights 32 × 320000 Bias 32 × 1	-
16	fc 2 fully connected layer	Fully Connected	1(S) × 1(S) × 2(C) × 1(B)	Weights 2 × 32 Bias 2 × 1	-
17	softmax softmax	Softmax	1(S) × 1(S) × 2(C) × 1(B)	-	-
18	classoutput crossentropyx	Classification Output	1(S) × 1(S) × 2(C) × 1(B)	-	-

Fig. 9 Network parameter details. The figure shows some of the necessary data for the network and the important parameters of the network, including the image input size, the size of the four convolutional kernels reduced in sequence, and the hyperparameters of the other modules

shown in Fig. 10a. This convolutional block consists of a 10×10 convolutional layer, a batch normalization layer, and a maximum pooling layer, which is then activated by a Relu function, a process given in detail by Eq. (7).

$$F_{out} = \sigma(\text{Max}(\text{BN}(C_j(F_i)))) \quad (7)$$

$$F_{out} = \delta(\text{Ave}(C_j(F_i))) \quad (8)$$

where F_i is the i -th feature map of the input in the batch, C_j is the i -th convolutional layer, BN is the batch normalization layer, Max is the maxpooling layer, Ave is the average-pooling layer, σ is the Relu activation function, and δ is the LeakyRelu activation function.

The output image size of the convolution layer for each convolution block is available through Eq. (9).

$$I' = \frac{(I - K + 2P)}{S} + 1 \quad (9)$$

where I' is the size of the output feature map, I is the size of the input feature map, K is the size of the filter, P is the amount of boundary padding, and S is the step size.

The second convolution block removes the batchnorm layer and reduces the convolution kernel to 7×7 , leaving the rest unchanged. With the second convolutional block, the feature texture map can be further extracted. If we use the feature maps extracted by these two convolutional blocks, the network will classify them all as raw. Therefore, we need to extract the local feature maps from the extracted global feature maps and, at the same time, increase the number of local feature maps.

With the third convolutional block, the local feature map can be initially extracted, as shown in Fig. 10b. This convolutional block consists of 3×3 convolutional layers, the LeakyReLU activation function, and the average pooling layer, a process given in detail by Eq. (8). The formula for LeakyReLU is shown in Eq. (10), where a takes the value 0.01.

$$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases} \quad (10)$$

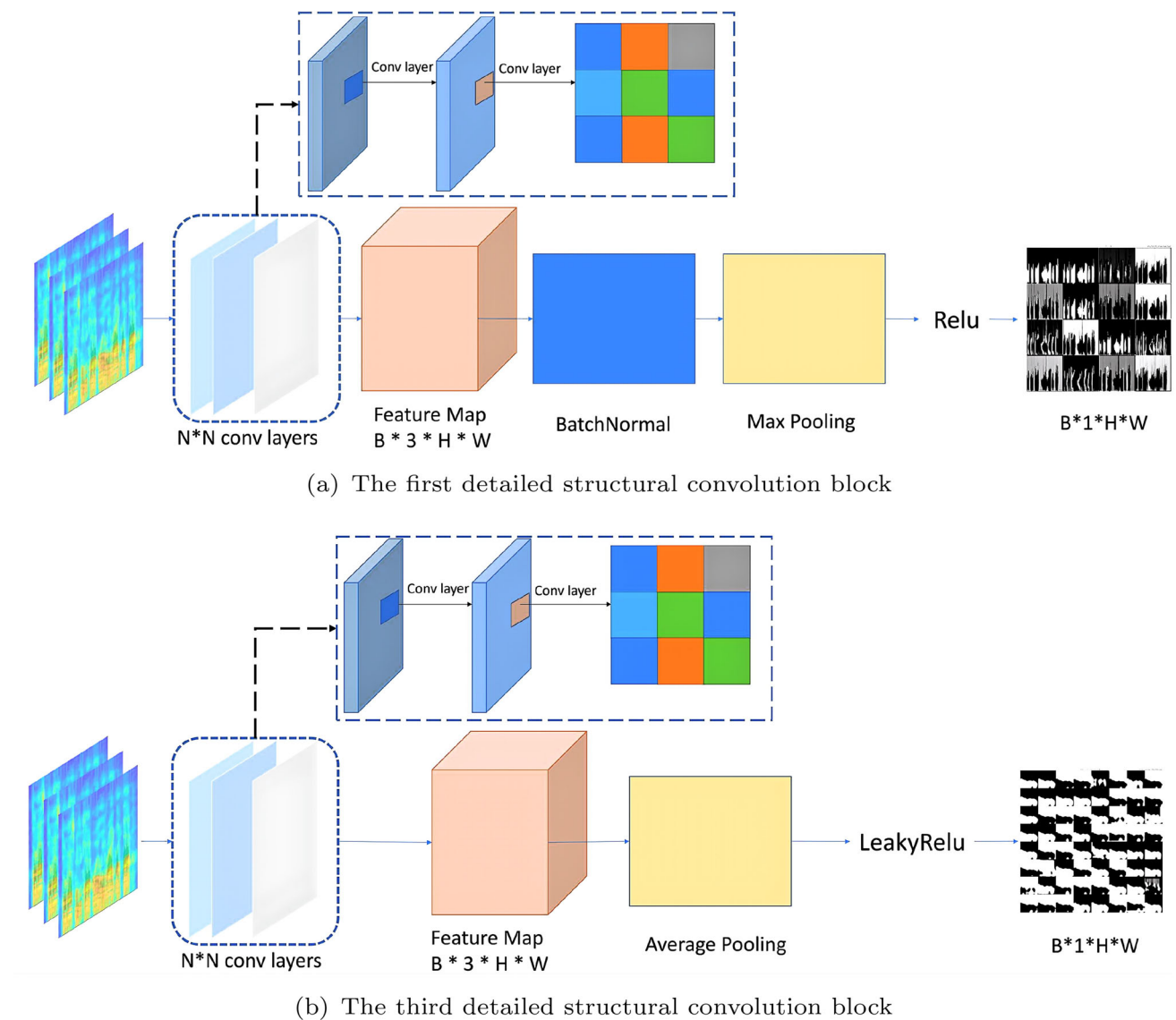


Fig. 10 Detailed framework for convolutional blocks. **a** shows the detailed connectivity of the modules in the first convolutional groups. **b** Shows the detailed connections of the modules in the third and fourth convolutional groups

The convolution kernel in the convolutional layer of the fourth convolutional block is reduced to 3×3 , and the rest have the same structure as the third convolutional block. The third and fourth convolutional blocks use different pooling and activation functions than the previous two. This is because the averaging pool allows better extraction of global features from the extracted local feature maps. The use of the LeakyReLU activation function contributes to the stability and generalisation performance of the network.

Finally, we feed the result of the convolution into the fully connected module and output the result via Eq. (11). Where the fully connected module consists of two fully connected layers, we set the parameters of the first fully connected layer to 32 neurons, and the second fully connected layer contains two neurons corresponding to two categories. The use of two fully connected layers gives the model better expressive and

improves the generalisation ability of the model. The first fully connected layer is processed as shown in Eq. (11).

$$p_{w,b}(x) = \sum_{i=1}^n w_i x_i + b_i \quad (11)$$

where $p_{w,b}(x)$ is the output, w_i is the weight matrix of the first fully connected layer, x_i is the input vector, and b_i is the bias vector of the first fully connected layer.

The second fully connected layer is processed differently from the first one, as compared to the first fully connected layer, the output of the second one needs to be activated by a softmax function once and thus categorized as shown in Eq. (12).

$$p_i(x) = f \left(\sum_{i=1}^n w_i p_{w,b}(x) + b_i \right) \quad (12)$$

where $p_i(x)$ is the categorical probability output, w_i is the weight matrix of the second fully connected layer, $p_{w,b}(x)$ is the input vector, b_i is the bias vector of the second fully connected layer, and f is the softmax activation function.

Our choice of optimizer is the Adam optimizer, and the loss function chosen is the cross-entropy loss function used for the binary classification problem, shown in Eq. (13).

$$L = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (13)$$

where N is the number of samples, y_i is the true label of the i -th sample, and p_i is the probability predicted by the model that the i -th sample belongs to a positive class.

The training set and test set required for the CNN were created using the appeal method from two databases. However, deep learning neural networks require a significant amount of data to tune the parameters, or else overfitting of the trained network may occur. To address this issue, we utilized data augmentation techniques. Data augmentation is a commonly used method in deep learning to increase the amount of data by deriving it from the original data. In the study, the image enhancement technique was employed to randomly flip the vertical and horizontal orientation of the image. These image-processing methods were utilized to generate the required amount of data.

The datasets acquired in the experiments were classified into two categories: forged and original. The training sessions amounted to 216, with a batch size of 64 for the validation dataset.

Experimental results

In this section, a detailed analysis of the experimental data is presented, and a comparison with mainstream research methods in the field is conducted to validate the feasibility of the proposed method. This evaluation considers not only the labeling of audio as forged but also the classification accuracy of the original audio. The experimental setup comprises an Intel Core i3-12100F 64-bit processor, 16GB RAM, a 1060 Ti graphics card, and the Windows 10 operating system. Model training was conducted using MATLAB 2023a software.

Dataset

We used two publicly available audio datasets to test the performance of the proposed method. The first one is the TIMIT audio database containing 4416 audio files. The audio

in the TIMIT audio dataset has an approximate length of 5 s, an audio sampling rate of 16 kHz for adults, 70% of the male recordings, and 30% of the female recordings, and an audio sampling rate of 16 kHz, and the language is English. The second one is the Arabic audio dataset with an audio length of about 3 s, recorded by a male, with an audio sampling rate of 48 kHz, and the language is Arabic.

In order to make the forgery method more realistic, both the original audio data and the forged audio data have been subjected to a variety of common post-processing operations, each of which corresponds to the numbering and strength of the attack as shown in Table 1, including noise addition, MP3 compression, median filtering, re-sampling, and low-pass filtering. In addition to this, randomly selected combinations of appealing attacks are also applied to the audio, such as adding MP3 compression after applying Gaussian white noise, and adding median filtering attack after applying MP3 compression attack.

Performance indicators

Our proposed method classifies not only the forged audio but also the original audio. From Eq. (14), we know that the number of labels in both test sets must be set to the same size to ensure reliable accuracy of the results.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

where TP is the number of samples correctly predicted, FN is the number of samples incorrectly predicted to be in the negative category, TN is the number of samples correctly predicted to be in the positive category, and FP is the number of samples incorrectly predicted to be in the positive category.

Our proposed method is different from traditional methods in terms of detection data. In traditional methods, they don't need to detect the original audio, and they don't need to check whether they misclassify the original audio or not. But in our proposed method, we classify the original audio and the forged audio after mixing them. So we also detect the original audio in the traditional way to ensure the consistency of the experiment and count the results. According to the recall rate formula in Eq. (15), the recall rate is equivalent to the check-all rate. However, for our task requirements, we need to focus on the precision Eq. (16), which represents the correct classification. Therefore, recall is not an important metric for the performance of our proposed model. In this context, we use the comprehensive performance metric F1

Table 1 Correspondence between the attack type labels used in the chart and the attack types and a detailed description of each attack type

Index	Attack type	Details
1	Gaussian noise addition	Gaussian noise with SNR of 20 dB is added
2	Gaussian noise addition	Gaussian noise with SNR of 30 dB is added
3	MP3 compression	MPEG 1 Layer III (MP3) compression at 32 kbps
4	MP3 compression	MPEG 1 Layer III (MP3) compression at 64 kbps
5	Median filtering	The recording is filtered by a median filtering
6	Resampling	The recording is first down-sampled at 8 kHz, and then up-sampled at 16 kHz
6	Re-quantization	Quantises the audio from 16-bit floating point to 8-bit and then to 16-bit.
7	Low-pass filtering	The recording is filtered using a low-pass filtering with a cutoff frequency of 4 kHz
8	MP3 compression and Median filtering	MP3 compression at 64 kbps followed by median filtering.
9	MP3 compression, Median filtering and Gaussian noise addition	MP3 compression at 64 kbps, median filtering, and Gaussian noise with 20 dB SNR.
10	MP3 compressing and Gaussian noise addition	MP3 compression at 64 kbps and Gaussian noise with 20 dB SNR.
11	Resampling and Re-quantization	Downsampled at 8 kHz, then upsampled at 16 kHz, then quantized from 16-bit floating point to 8-bit, then to 16-bit.
12	MP3 compression, Gaussian noise addition and Resampling	64 kbps MP3 compression, Gaussian noise with 20 dB SNR, and resampling from 8 kHz to 16 kHz.
13	Median filtering, Gaussian noise addition and Low-pass filtering	Median filtering, Gaussian noise with 20 dB SNR, and low-pass filtering at 4 kHz.
14	No additional attack	No attacks added to the audio

score to measure the merits of the proposed algorithm, and the formula for F1 score is shown in Eq. (17).

$$F1 = \frac{2 \times P \times R}{P + R} \quad (17)$$

where P is the precision rate and R is the recall rate.

Results with data augmentation

As we know, training convolutional neural networks requires a large amount of data. If the amount of data is too small, it tends to cause an overfitting phenomenon in the network. Therefore, we need to use data augmentation techniques, including rotation and stretching operations, to increase the amount of data from the existing data.

We compared the performance with and without the use of data enhancement techniques on the Arabic audio Corpus [33] audio dataset. The results of the comparison are shown in Table 2, where it can be seen that after applying the data augmentation technique, the performance of the model is significantly improved except for the recall. The decrease in recall is a small decrease in precision on the positive categories, but the other performance improvements are small increases in accuracy on the negative categories. Overall, the

use of data augmentation techniques improves the robustness of the proposed method.

Visualisation of spectrograms under attack

In this subsection, we will show the visual effect of the same attack (adding 20 dB Gaussian white noise) with different weight combinations and the visual effect of different attacks with the same weight combinations ($w_1 = 0.8$, $w_2 = 0.2$). As shown in Fig. 11, the first column shows the images fused with different weights. The second column shows the initial feature image obtained from the input image. The third column shows the image after passing through the convolutional layer. The fourth column shows the categories labeled by the network for the audio. The fifth column shows the actual labeling of the audio.

Figure 11a shows the visual effect of adding 20db Gaussian white noise with different weight combinations of $w_1 = 0.7$, $w_2 = 0.3$; $w_1 = 0.8$, $w_2 = 0.2$; $w_1 = 0.9$, $w_2 = 0.1$. Figure 11b Shows the visual effects under the same weight combinations for the three attacks: double hybrid attack with MP3 compression and median filtering, triple hybrid attack with MP3 compression, median filtering, and added noise, and double hybrid attack with MP3 compression and added noise. It can be clearly seen that the network's processing of the audio spectrogram, in turn, becomes better when more

Table 2 Ablation experiments

Categories	Accuracy	Precision	Recall	F1 score
Without data augmentation	0.88	0.73	0.95	0.83
With data augmentation	0.91	0.88	0.91	0.89

post-processing is applied. This explains the fact that the network's detection accuracy for unattacked audio is lower than the detection accuracy for attacked audio.

Results with post-processing operations

In this subsection, we analyze the performance of the proposed method on two different datasets. Table 3 shows the performance of the proposed method in this paper under different attacks on the Arabic speech corpus. From Table 3, we can see that the accuracy of the proposed method under various attacks exceeds 89% and can reach up to 99%. The F1 score also exceeds 89% and can reach up to 99% as well. This indicates that the proposed method has a good performance on the Arabic speech corpus.

Table 4 shows the performance of the proposed method on the TIMIT dataset under different attacks. As we can see from Table 4, the performance of the proposed method varies under different compression attacks. The accuracy under 32 kbps compression can reach 95% and the F1 score can reach 94%. Under 64 kbps compression, the accuracy can reach 83% and the F1 score can reach 84%. Under other attacks, the accuracy and F1 score are over 87%. This shows that the proposed method has good performance on the TIMIT dataset.

Comparison with other detection methods

This subsection compares the performance of the proposed method with four main literature methods on copy-move datasets. In the "Introduction", we discussed the literature on detecting copy movement in audio. This paper compares four main literature methods: Pitch and Formant [10], SIFT [11], GA-SVM [15], SW-CQCC [13], and CNN [18]. We compare performance with other research methods using the TIMIT audio dataset and Arabic Speech corpus on the same machines, and our method uses a combination of weights of $w_1 = 0.8$ and $w_2 = 0.2$. The performance metrics we used include accuracy, precision, recall, and F1 score.

Accuracy under different attacks

In the first part of the comparison experiment, we compare the accuracy of the proposed method with other methods on the TIMIT dataset. As shown in Fig. 12, the highest accuracy rate of CQSS-GA-SVM among mainstream literature meth-

ods is close to 92%. However, the accuracy of our proposed method reaches up to 98%. Moreover, the accuracy rate of the proposed method is increasing while the accuracy rates of all other methods are decreasing when the number of attacks increases.

We have compared the accuracy of the proposed method with other methods in the mainstream literature on the Arabic speech corpus. As shown in Fig. 13, although the CQSS-GA-SVM method performs the best in the existing literature with an accuracy rate of nearly 93%, our proposed method achieves the highest accuracy rate of 99% under the same conditions. Similarly, as the number of attacks increases, the accuracy of the proposed method not only does not decrease but also increases, while the accuracy of the other methods generally decreases. This indicates that the proposed method has stronger robustness and adaptability when facing more attacks.

Recall under different attacks

In the second part of the comparison experiments, we compare the recall of the proposed method with other methods on the TIMIT dataset and the Arabic speech corpus. As shown in Fig. 14, the recall of our proposed algorithm is higher than that of the other methods in the vast majority of post-processing operations. Except for the cases of median filtering and no attack, the recall of the proposed method is higher than that of mainstream literature methods, reaching up to 99%.

The recall test results on the Arabic speech corpus, as shown in Fig. 15, show that the recall of our proposed algorithm is higher than that of other methods in the vast majority of post-processing operations. Except for the case of 64 kbps MP3 compression and no attack, the recall of the proposed method is higher than the mainstream literature methods, up to 99%.

F1 score under different attacks

In the third part of the comparison experiments, we compare the F1 score with mainstream literature methods on the TIMIT dataset and the Arabic speech corpus. As shown in Fig. 16, our proposed algorithm achieves a higher F1 score than the mainstream literature methods in most of the post-processing. Moreover, the more post-processing attacks the

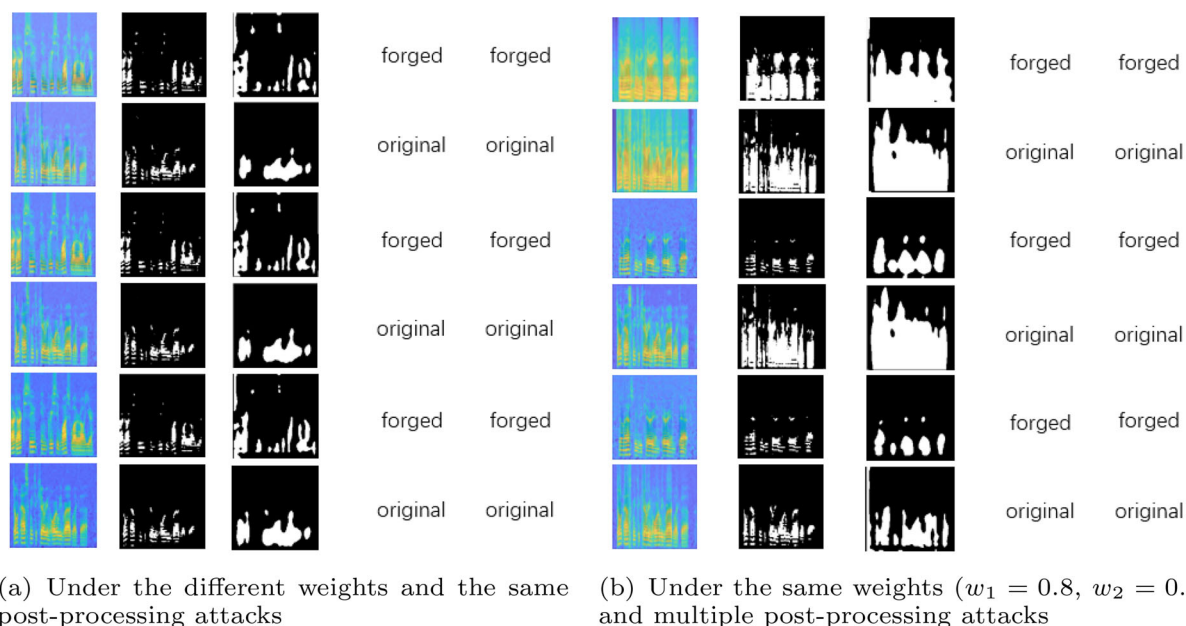


Fig. 11 Output of the fused spectrogram in the network. **a** Shows a comparison of the network's output for the same audio with different fusion weights and adding the same attacks. **b** Shows a comparison of the network's output for the same audio with the same fusion weights and with different attacks added

Table 3 Test results of the proposed method on the Arabic speech corpus

Post-processing	Accuracy	Precision	Recall	F1 score
Noise addition attack-20 dB	0.92	0.93	0.92	0.92
Noise addition attack-30 dB	0.93	0.90	0.95	0.93
32 kbps MP3 compressing	0.94	0.91	0.98	0.95
64 kbps MP3 compressing	0.94	0.95	0.94	0.94
Median filtering	0.96	0.99	0.93	0.96
Resampling	0.89	0.90	0.92	0.91
Re-quantization	0.92	0.93	0.95	0.94
Low-pass filtering	0.92	0.89	0.93	0.91
MP3 compressing and median filtering	0.991	1	0.992	0.996
MP3 compressing, median filtering and Gaussian noise addition	0.982	1	0.969	0.984
MP3 compressing and Gaussian noise addition	0.980	1	0.965	0.982
MP3 compressing, Gaussian noise addition and resampling	0.985	1	0.972	0.986
Median filtering, Gaussian noise addition and low-pass filtering	0.982	1	0.982	0.991
No additional attack	0.91	0.88	0.91	0.89

audio is subjected to, the higher the F1 score of our proposed method.

The F1 score was compared with mainstream literature methods on an Arabic speech corpus. As shown in Fig. 17, our proposed algorithm achieves higher F1 scores than mainstream literature methods for most of the post-processing. Under the 64 kbps MP3 compression attack, the F1 score of CQSS-GA-SVM is the same as that of the proposed method, and lower than that of the proposed method under all other attacks. In addition, the more post-processing attacks the

audio is subjected to, the higher the F1 score of our proposed method.

Robustness experiments with different methods

In this subsection, we compare the proposed method with other methods on the TIMIT dataset and the Arabic Speech Corpus and statistically determine the accuracy and F1 score of adding single and multiple post-processes to the audio.

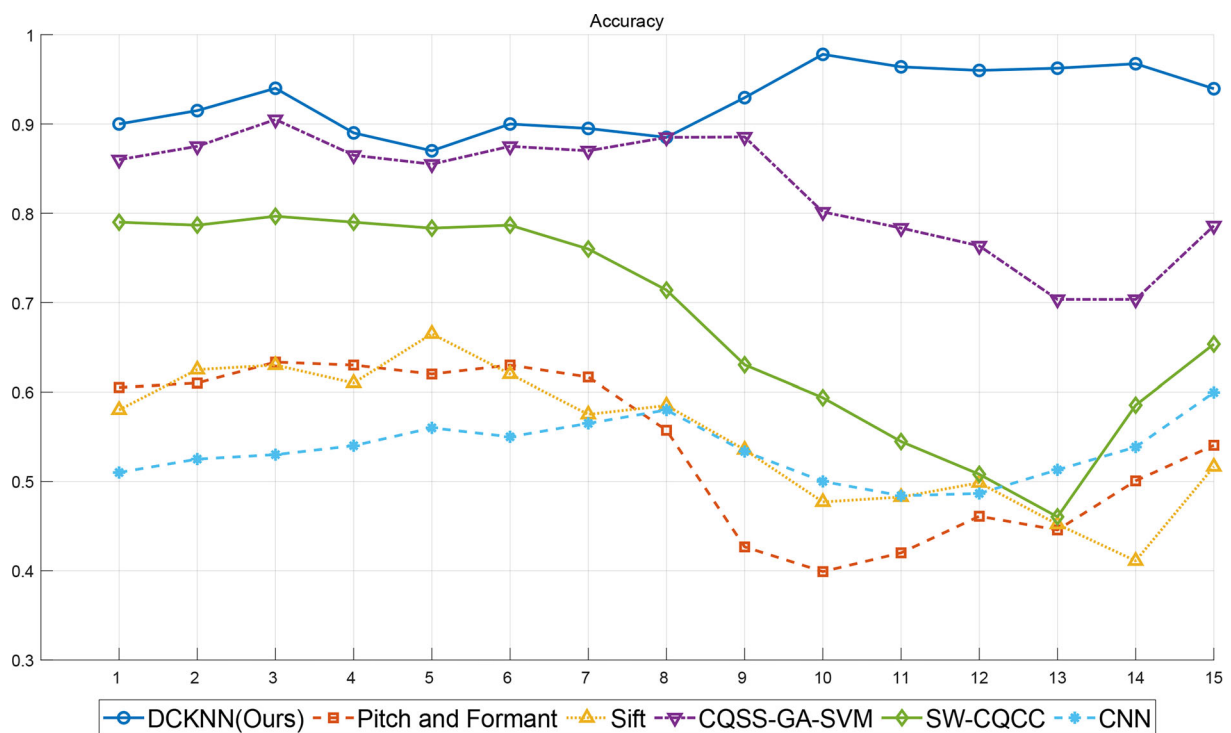


Fig. 12 Comparison of accuracy of the proposed method and other methods on the TIMIT dataset with post-processing

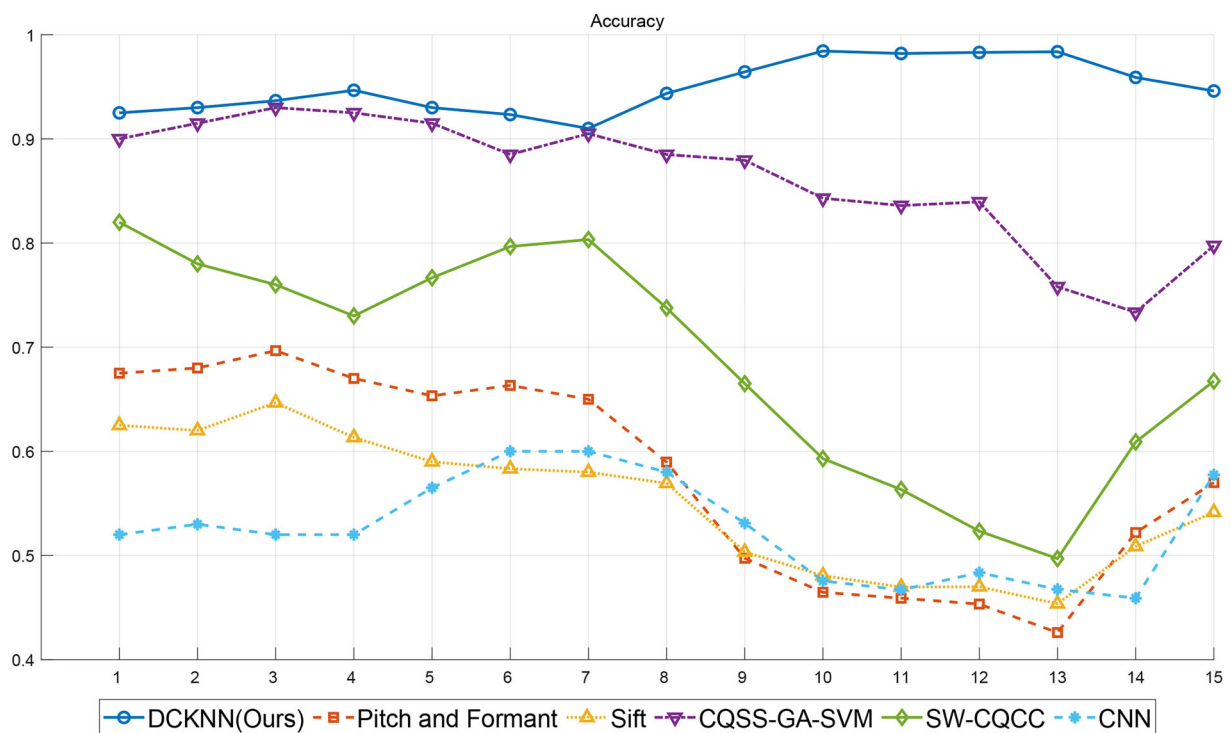
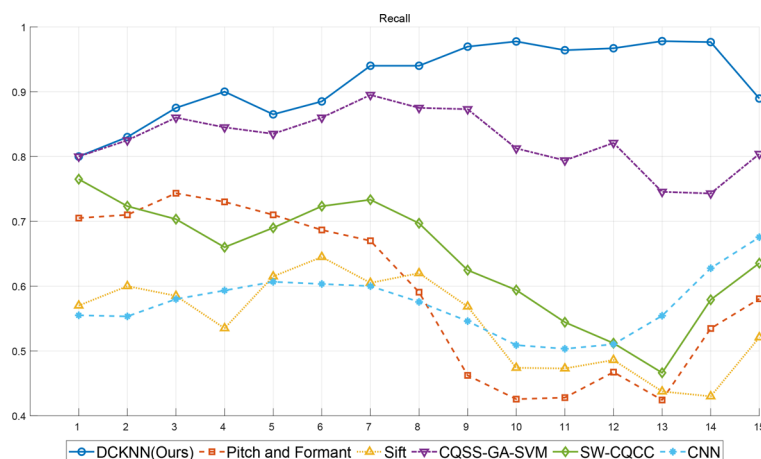


Fig. 13 Comparison of accuracy of the proposed method and other methods on the Arabic speech corpus with post-processing

Table 4 Test results of the proposed method on the TIMIT

Post-processing	Accuracy	Precision	Recall	F1 score
Noise addition attack-20 dB	0.90	1	0.80	0.89
Noise addition attack-30 dB	0.93	0.99	0.86	0.92
32 kbps MP3 compressing	0.95	1	0.89	0.94
64 kbps MP3 compressing	0.83	0.78	0.91	0.84
Median filtering	0.91	1	0.82	0.90
Resampling	0.89	0.89	0.95	0.92
Re-quantization	0.90	0.93	0.93	0.93
Low-pass filtering	0.87	0.87	0.95	0.91
MP3 compressing and median filtering	0.989	1	0.989	0.995
MP3 compressing, median filtering and Gaussian noise addition	0.967	1	0.966	0.983
MP3 compressing and Gaussian noise addition	0.961	1	0.962	0.981
Resampling and re-quantization	0.959	0.972	1	0.986
MP3 compressing, Gaussian noise addition and resampling	0.966	1	0.984	0.992
Median filtering, Gaussian noise addition and low-pass filtering	0.969	1	0.969	0.984
No additional attack	0.91	1	0.81	0.90

Fig. 14 Comparison of recall of the proposed method and other methods on the TIMIT dataset with post-processing

The performance test results of adding single post-processing to the audio in the TIMIT dataset are shown in Table 6, except for the 64 kbps MP3 compression attack, the proposed method outperforms the other methods in the remaining single post-processing operations, with the highest accuracy of 100% and the highest F1 score of 94%. The performance test results of adding multiple post-processing to the audio in the TIMIT dataset are shown in Table 8, where the proposed method has a particularly excellent performance in the face of multiple post-processing. The proposed method achieves an accuracy of 100% under each of the multiple post-processing attacks, and the F1 score exceeds 98% for all of them. With the addition of single or multiple post-processes on the TIMIT dataset, the proposed method shows more robustness than other methods.

The performance test results on the Arabic Speech Corpus with single post-processing are shown in Table 5. The

proposed method outperforms the performance metrics of the other methods under most single post-processing operations, with the highest accuracy reaching 99% and the highest F1 score reaching 96%. The performance of the proposed method is also only 3% less than the performance of the best method under attacks that do not outperform other methods.

The performance test results on the Arabic Speech Corpus with multiple post-processing are shown in Table 7. The proposed method exhibits particularly excellent performance when facing multiple post-processing operations. Under each type of multiple post-processing attack, the accuracy of the proposed method reaches 100% and the F1 score exceeds 98%. The proposed method demonstrates strong robustness on the Arabic Speech Corpus, whether facing single or multiple post-processing operations. Especially when detecting audio with multiple post-processing, the proposed method achieves nearly perfect performance.

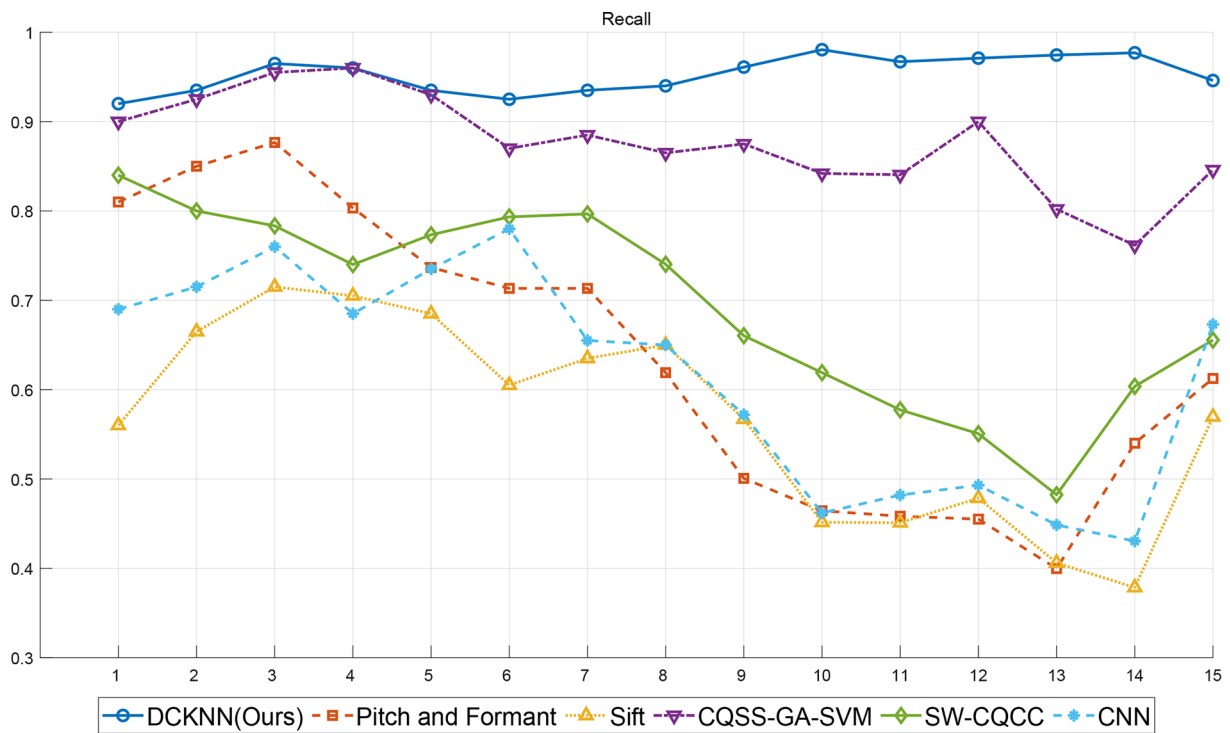


Fig. 15 Comparison of recall of the proposed method and other methods on the arabic speech corpus with post-processing

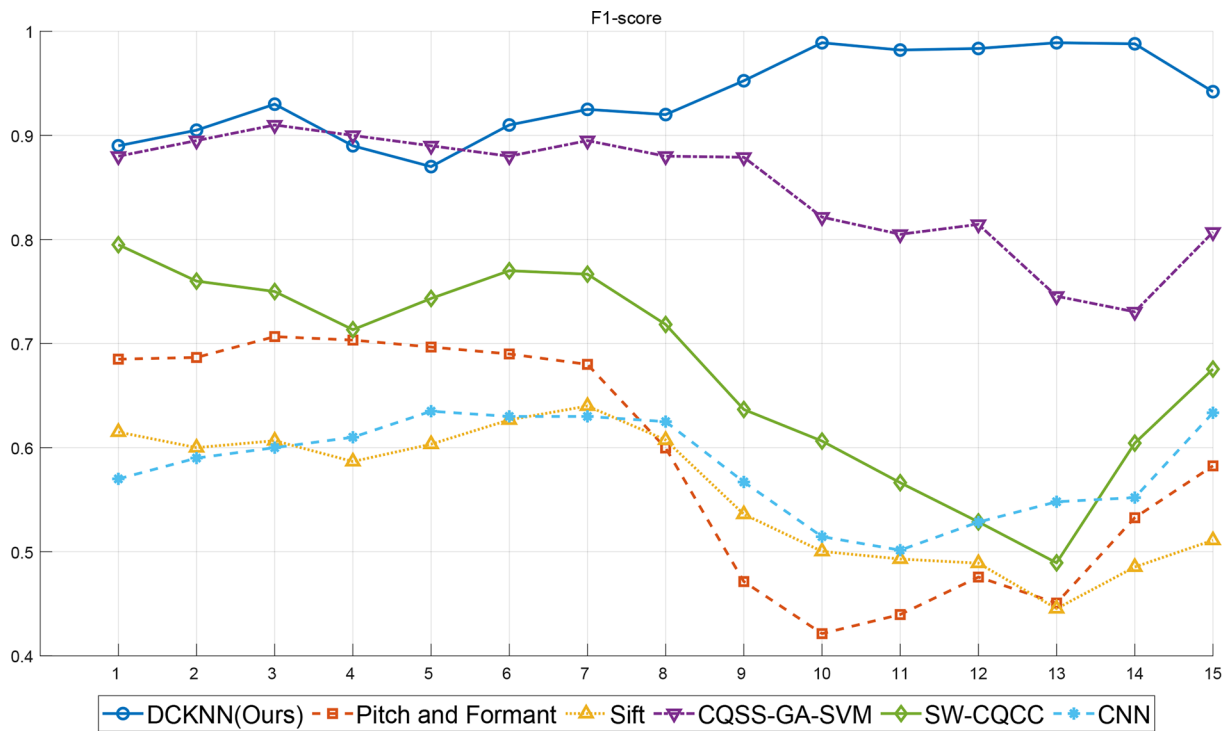


Fig. 16 Comparison of F1 scores of the proposed method and other methods on the TIMIT dataset with post-processing

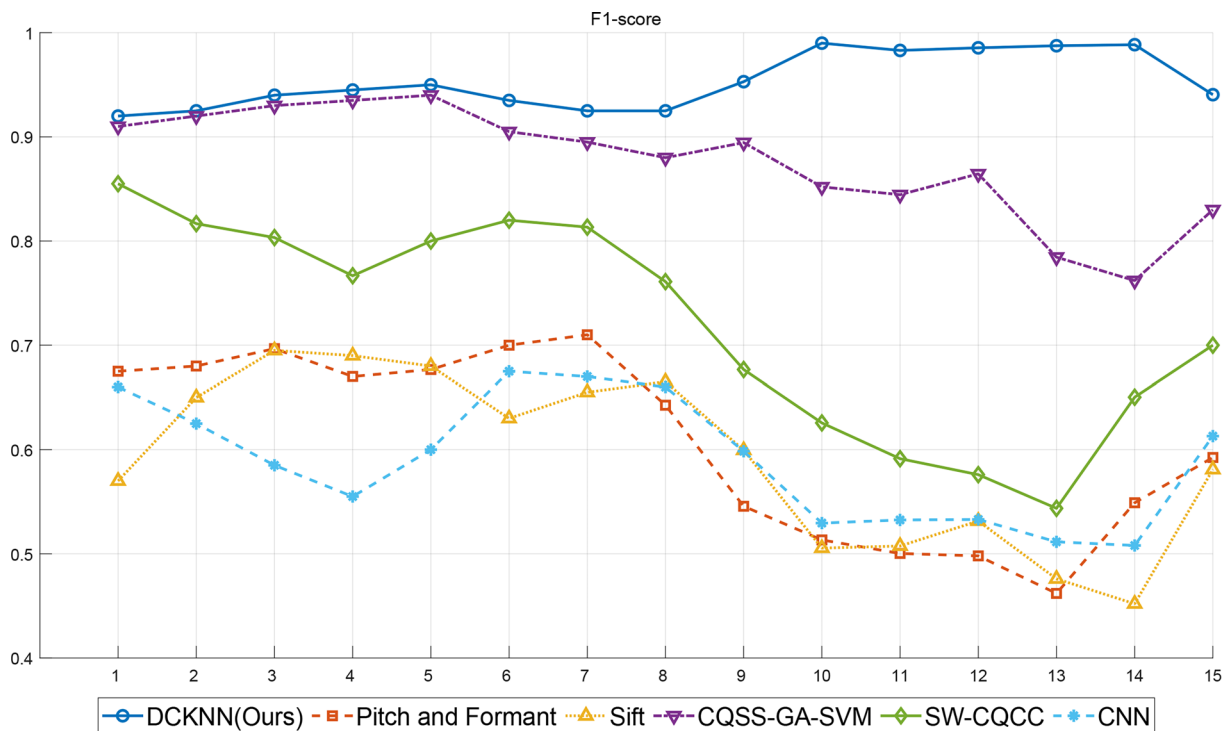


Fig. 17 Comparison of F1 scores of the proposed method and other methods on the arabic speech corpus with post-processing

Discussion

In this subsection, we provide an in-depth analysis of the specific role of the model in coping with the audio tampering detection task. It is clear from the experimental results that the model outperforms the unattacked audio when dealing with a single attack and further outperforms the single attack in the case of multiple attacks. This phenomenon suggests that the proposed model is extremely robust in the face of different levels of attacks and is able to effectively detect complex tampering operations.

Specifically, Fig. 11b demonstrates the effect of different convolutional kernel sizes on the model performance. The change from the second to the third column shows that the tampered segments in the spectrogram are captured more accurately when the convolutional layers with larger convolutional kernels are used. This means that larger convolutional kernels are better able to adapt to the subtle changes brought about by audio post-processing, enhancing the network's ability to capture tampered audio features. Then by reducing the convolutional kernel, the network can more effectively extract the tampering traces hidden in the multi-layer attacks, thus achieving higher classification accuracy.

In addition, sequentially decreasing convolutional kernels not only improve the detection ability of the network, but also enable it to ignore, to some extent, unwanted noises and variations brought about by audio post-processing. This is because the image features extracted by the larger convolutional

kernel can effectively reduce the noise interference of the smaller convolutional kernel when extracting local details, thus improving the generalisation ability of the model. At the same time, the use of larger convolutional kernels can also reduce the number of parameters in the network and optimise the computational efficiency of the model. This feature makes the proposed method not only excellent in terms of accuracy, but also has advantages in terms of resource consumption and execution efficiency.

Collectively, by using sequentially decreasing convolutional kernels, the model demonstrates greater robustness and adaptability to multiple attacks, audio tampering, and post-processing changes. This design enables the network to better capture audio features after attacks, ensuring efficient and accurate classification under complex audio attacks.

Conclusion

Currently, audio copy-move forgery is relatively easy to perform, and attackers often employ single or multiple post-processing techniques to obscure traces of their tampering. Therefore, there is an urgent need for highly robust detection methods. In this paper, we propose a convolutional network with progressively decreasing convolutional kernels for audio copy-move forgery detection. The method begins by fusing the Mel spectrogram and Hilbert–Huang spectrogram of the audio, which are then input into a pre-

Table 5 Performance comparison under different post-processing on an Arabic speech corpus

Post-processing	DCKNN (ours)			Pitch and Formant [10]			SIFT [11]			CQSS-GA-SVM [15]			SW-CQCC [13]			CNN [18]		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Noise addition attack-20 dB	0.93	0.92	0.92	0.92	0.92	0.92	0.58	0.56	0.57	0.92	0.90	0.91	0.85	0.81	0.83	0.63	0.69	0.66
Noise addition attack-30 dB	0.90	0.95	0.93	0.93	0.60	0.87	0.71	0.69	0.77	0.91	0.95	0.93	0.89	0.87	0.88	0.49	0.74	0.59
32 kbps MP3 compressing	0.91	0.98	0.95	0.95	0.55	0.93	0.69	0.66	0.66	0.90	0.96	0.93	0.77	0.72	0.74	0.46	0.78	0.58
64 kbps MP3 compressing	0.95	0.94	0.94	0.94	0.59	0.83	0.69	0.69	0.75	0.92	0.96	0.94	0.82	0.76	0.79	0.48	0.59	0.53
Median filtering	0.99	0.93	0.96	0.96	0.61	0.65	0.63	0.66	0.62	0.98	0.90	0.94	0.80	0.74	0.77	0.54	0.88	0.67
Resampling	0.90	0.92	0.91	0.69	0.73	0.71	0.66	0.66	0.59	0.90	0.84	0.94	0.91	0.82	0.77	0.68	0.68	0.68
Re-quantization	0.93	0.95	0.94	0.76	0.76	0.76	0.71	0.68	0.69	0.91	0.93	0.92	0.88	0.82	0.85	0.69	0.63	0.66
Low-pass filtering	0.89	0.93	0.91	0.67	0.65	0.67	0.70	0.62	0.62	0.88	0.80	0.87	0.91	0.75	0.76	0.65	0.67	0.66
No additional attack	0.88	0.91	0.89	0.64	0.85	0.73	0.68	0.75	0.71	0.90	0.92	0.91	0.91	0.89	0.90	0.58	0.91	0.71
Average	0.920	0.937	0.928	0.630	0.780	0.691	0.667	0.667	0.664	0.913	0.907	0.910	0.843	0.798	0.817	0.578	0.730	0.638

Table 6 Performance comparison under different post-processing on the TIMIT dataset

Post-processing	DCKNN (ours)			Pitch and Formant [10]			SIFT [11]			CQSS-GA-SVM [15]			SW-CQCC [13]			CNN [18]		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Noise addition attack-20 dB	1	0.80	0.89	0.66	0.68	0.67	0.58	0.57	0.57	0.97	0.80	0.88	0.82	0.74	0.77	0.63	0.53	0.57
Noise addition attack-30 dB	0.99	0.86	0.92	0.68	0.73	0.70	0.68	0.63	0.66	0.98	0.85	0.91	0.85	0.79	0.82	0.65	0.58	0.61
32 kbps MP3 compressing	1	0.89	0.94	0.67	0.72	0.69	0.59	0.54	0.57	0.95	0.87	0.91	0.74	0.64	0.69	0.64	0.55	0.59
64 kbps MP3 compressing	0.78	0.91	0.84	0.68	0.78	0.73	0.67	0.53	0.59	0.97	0.82	0.89	0.80	0.68	0.74	0.66	0.61	0.63
Median filtering	1	0.82	0.90	0.69	0.69	0.69	0.70	0.70	0.60	0.94	0.85	0.89	0.77	0.66	0.71	0.67	0.62	0.64
Resampling	0.89	0.95	0.92	0.68	0.66	0.67	0.66	0.59	0.62	0.87	0.87	0.87	0.82	0.73	0.78	0.66	0.59	0.62
Re-quantization	0.93	0.93	0.93	0.71	0.71	0.71	0.70	0.62	0.66	0.92	0.92	0.92	0.78	0.85	0.82	0.69	0.60	0.64
Low-pass filtering	0.89	0.95	0.91	0.68	0.64	0.66	0.67	0.62	0.64	0.85	0.83	0.84	0.69	0.69	0.70	0.65	0.61	0.61
No additional attack	1	0.81	0.90	0.69	0.76	0.73	0.65	0.60	0.62	0.96	0.85	0.90	0.91	0.81	0.86	0.68	0.72	0.70
Average	0.940	0.880	0.906	0.682	0.708	0.694	0.656	0.600	0.614	0.934	0.851	0.890	0.806	0.724	0.766	0.659	0.601	0.623

Table 7 Performance comparison of the proposed method and other methods after applying multiple attacks to the arabic speech corpus

Post-processing	DCKNN (ours)			Pitch and formant [10]			SIFT [11]			CQSS-GA-SVM [15]			SW-CQCC [13]			CNN [18]		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MP3 and median	1	0.992	0.996	0.586	0.447	0.508	0.613	0.514	0.559	0.949	0.950	0.949	0.718	0.651	0.683	0.621	0.474	0.537
MP3, median and noise	1	0.969	0.984	0.556	0.405	0.469	0.539	0.389	0.452	0.777	0.734	0.755	0.615	0.580	0.597	0.621	0.450	0.522
MP3 and noise	1	0.965	0.982	0.588	0.541	0.563	0.625	0.513	0.563	0.921	0.947	0.934	0.650	0.626	0.597	0.576	0.514	0.543
Resampling and re-quantization	1	0.977	0.989	0.517	0.429	0.469	0.571	0.444	0.500	0.744	0.853	0.795	0.645	0.526	0.580	0.586	0.472	0.523
MP3, noise and resampling	1	0.972	0.986	0.556	0.395	0.462	0.583	0.368	0.452	0.799	0.751	0.774	0.613	0.500	0.551	0.607	0.425	0.500
Median, noise and low-pass filtering	1	0.982	0.991	0.577	0.375	0.455	0.539	0.389	0.452	0.730	0.772	0.750	0.615	0.421	0.500	0.630	0.436	0.516
Average	1	0.976	0.988	0.563	0.432	0.488	0.578	0.436	0.496	0.820	0.835	0.826	0.643	0.551	0.585	0.607	0.462	0.525

Table 8 Performance comparison of the proposed method and other methods after applying multiple attacks to the TIMIT dataset

Post-processing	DCKNN (ours)			Pitch and formant [10]			SIFT [11]			CQSS-GA-SVM [15]			SW-CQCC [13]			CNN [18]		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MP3 and median	1	0.989	0.995	0.436	0.422	0.429	0.527	0.517	0.522	0.921	0.916	0.918	0.649	0.621	0.635	0.531	0.517	0.524
MP3, median and noise	1	0.966	0.983	0.325	0.325	0.325	0.462	0.431	0.446	0.741	0.709	0.725	0.588	0.563	0.575	0.499	0.511	0.505
MP3 and noise	1	0.984	0.981	0.491	0.443	0.510	0.552	0.418	0.533	0.891	0.728	0.885	0.622	0.466	0.609	0.497	0.532	0.498
Resampling and re-quantization	1	0.972	0.986	0.556	0.429	0.484	0.552	0.457	0.500	0.725	0.763	0.744	0.567	0.472	0.515	0.633	0.500	0.559
MP3, noise and resampling	1	0.969	0.992	0.423	0.401	0.433	0.451	0.442	0.434	0.766	0.758	0.747	0.458	0.461	0.462	0.542	0.631	0.537
Median, noise and low-pass filtering	1	0.984	0.984	0.475	0.443	0.435	0.369	0.418	0.402	0.676	0.728	0.714	0.525	0.466	0.491	0.515	0.532	0.567
Average	1	0.974	0.987	0.451	0.425	0.436	0.486	0.463	0.473	0.787	0.792	0.789	0.568	0.530	0.548	0.536	0.532	0.532

defined DCKNN (Decreasing Convolution Kernel Neural Network) model for training. The DCKNN model is specifically designed to classify the new spectrogram into categories of original and forged images. Furthermore, we integrate data augmentation techniques to enhance the training process. We evaluated the model on the TIMIT dataset and the Arabic speech corpus, and the experimental results demonstrate that our proposed method outperforms existing techniques. Notably, the proposed approach excels in detecting audio with multiple post-processes, underscoring its strong robustness and applicability in complex forgery scenarios.

Author contributions The authors Xin Qiu and Canghong Shi contributed to the conception of the study, performed the experiment the data analyses and wrote the manuscript. The authors Xiaojie Li, Min Wu, Xianhua Niu and Sani M. Abdullahi contributed significantly to analysis and manuscript preparation, and they helped perform the analysis with constructive discussions. All authors reviewed the manuscript.

Funding The funding provided by Sichuan Province Science and Technology Support Program (Grant number: 2023NSFSC0470, 2024ZDZX0004), National Natural Science Foundation of China (Grant number: 62171387), Xihua University Science and Technology Innovation Competition Project for Postgraduate Students (Grant number: YK20240140).

Data Availability The data that support the findings of this study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethics approval and consent to participate The authors are satisfied that the data we use is publicly available to others and consent to the use of audio data.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Zakariah KM (2018) Digital multimedia audio forensics: past, present and future. *Multimedia Tools Appl* 77(1):1009–1040
- Bevinamarad PR, Shirldonkar MS (2020) Audio forgery detection techniques: present and past review. In: 2020 4th International conference on trends in electronics and informatics (ICOEI) (48184), pp 613–618
- Kraetzer C, Memon ND, Dittmann J, Qian K, Schott M, Alattar AM, Delp Iii EJ, Dittmann J (2011) A context model for microphone forensics and its application in evaluations. *Media Watermark Secur Forens III* 7880:78800
- Xinxin LMYLSN (2022) Semi-fragile audio watermarking algorithm in dwt domain. *China Commun Engl Ed* (4)
- Hua G, Goh J, Thing V (2015) Time-spread echo-based audio watermarking with optimized imperceptibility and robustness. *IEEE-Acm Trans Audio Speech Lang Process* 23(2):227–239
- Ji-Nian X, Yun-Zhe J, Er-Dong FU, Zheng H, Yan LI, Shao-Pei S (2014) Audio authenticity: duplicated audio segment detection in waveform audio file. *J Shanghai Jiaotong Univ (Sci)* (004), 019
- Huang X, Liu Z, Lu W, Liu H, Xiang S (2019) Fast and effective copy-move detection of digital audio based on auto segment. *Int J Digit Crime Forens* 11(2):47–62
- Imran M, Ali Z, Bakhsh ST, Akram S (2017) Blind detection of copy-move forgery in digital audio forensics. *IEEE Access* 5:12843–12855
- Küçükuğurlu B, Ustubioglu B, Ulutas G (2020) Duplicated audio segment detection with local binary pattern. In: 2020 43rd International conference on telecommunications and signal processing (TSP), pp 350–353
- Yan Q, Yang R, Huang J (2019) Robust copy-move detection of speech recording using similarities of pitch and formant. *IEEE Trans Inf Forens Secur* 14(9):2331–2341
- Beste U, Gul T, Guzin U (2023) Detection of audio copy-move-forgery with novel feature matching on mel spectrogram. *Expert Syst Appl* 213:118963
- Ustubioglu B, Tahaoglu G, Uluta G, Ustubioglu A, Kili M (2023) Audio forgery detection and localization with super-resolution spectrogram and keypoint-based clustering approach. *J Supercomput*
- Su Z, Li M, Zhang G, Wu Q, Wang Y (2023) Robust audio copy-move forgery detection on short forged slices using sliding window. *J Inf Secur Appl* 75:103507. <https://doi.org/10.1016/j.jisa.2023.103507>
- Xie Z, Lu W, Liu X, Xue Y, Yeung Y (2018) Copy-move detection of digital audio based on multi-feature decision. *J Inf Secur Appl* 43:37–46
- Su Z, Li M, Zhang G, Wu Q, Li M, Zhang W, Yao X (2023) Robust audio copy-move forgery detection using constant q spectral sketches and ga-svm. *TDSC* 20:4016–4031
- Akdeniz F, Becerikli Y (2024) Recurrent neural network and long short-term memory models for audio copy-move forgery detection: a comprehensive study. *J Supercomput*:1–31
- Dincer S, Ustubioglu B, Ulutas G, Tahaoglu G, Ustubioglu A (2023) Robust audio forgery detection method based on capsule network. In: 2023 International conference on electrical and information technology (IEIT). IEEE, pp 243–247
- Arda U, Beste U, Guzin U (2023) Mel spectrogram-based audio forgery detection using cnn. *SIViP* 17:2211–2219
- Ulutas G, Tahaoglu G, Ustubioglu B (2023) Deepfake audio detection with vision transformer based method. In: 2023 46th International conference on telecommunications and signal processing (TSP), pp 244–247. <https://doi.org/10.1109/TSP59544.2023.10197715>
- Peng K, Shi C, Li X, Qi C, Qiu X, Niu X (2025) Robust audio copy-move detection and localisation based on smoothed pseudo wigner-ville distribution and adaptive processing. *Appl Acoust* 231:110429. <https://doi.org/10.1016/j.apacoust.2024.110429>
- Saxena N, Sharma KK (2017) Image fusion scheme using two dimensional discrete fractional fourier transform. In: 2017 Conference on information and communication technology (CICT), pp 1–6. <https://doi.org/10.1109/INFOCOMTECH.2017.8340631>

22. Diao S, Wan Y, Huang S, Ma H (2024) Research on cancer prediction and identification based on multimodal medical image fusion. In: *Proceedings of the 2024 3rd international symposium on robotics, artificial intelligence and information engineering*, pp 120–124
23. Zhou M, Huang J, Yan K, Hong D, Jia X, Chansussot J, Li C (2024) A general spatial-frequency learning framework for multimodal image fusion. *IEEE Trans Pattern Anal Mach Intell*
24. Li Z, Liu F, Yang W, Peng S, Zhou J (2022) A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans Neural Netw Learn Syst* 33(12):6999–7019
25. Liu Y, Guo W, Lu X, Kong L, Yan Z (2025) Class activation map guided backpropagation for discriminative explanations. *Appl Sci* 15(1):379
26. Jung H, Oh Y (2021) Towards better explanations of class activation mapping. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 1336–1344
27. Chefer H, Gur S, Wolf L (2021) Transformer interpretability beyond attention visualization. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 782–791
28. Shi G, Liu B, Walls L (2022) Data augmentation to improve the performance of ensemble learning for system failure prediction with limited observations. In: *2022 13th International conference on reliability, maintainability, and safety (ICRMS)*, pp 296–300
29. Salamon J, Bello JP (2017) Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process Lett* 24(3):279–283
30. Rezaul KM, Jewel M, Islam MS, Siddiquee K, Barua N, Rahman M, Sulaiman R, Shaikh M, Hamim M, Tanmoy F et al (2024) Enhancing audio classification through mfcc feature extraction and data augmentation with cnn and rnn models. *Int J Adv Comput Sci Appl* 15(7):37–53
31. Hizlisoy S, Arslan RS, Çolakoğlu E (2024) Singer identification model using data augmentation and enhanced feature conversion with hybrid feature vector and machine learning. *EURASIP J Audio Speech Music Process* 2024(1):14
32. Alemu AA, Melese MD, Salau AO (2024) Ethio-semitic language identification using convolutional neural networks with data augmentation. *Multimedia Tools Appl* 83(12):34499–34514
33. Halad N Modern standard Arabic phonetics for speech synthesis. <http://en.arabicspeechcorpus.com/>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.