Gümüşhane University Journal of Science and Technology

GUFBD / *GUJS* (2026) 16(3): 878-898 doi: 10.17714/gumusfenbil.1888092 Research Article

A hybrid SuperPoint-based feature extraction and classification approach for audio copy-move forgery detection

Şeyma YÜCEL ALTAY*

Atatürk University, Faculty of Engineering, Department of Software Engineering, 25240, Erzurum

• Received: 12.02.2026 • Accepted: 18.08.2026

Abstract

Copy-move forgery, a widely used and content-based manipulation method in audio forensics, is performed by copying a portion of an audio file and pasting it into a different time period. The development of digital audio editing software has led to an increase in this type of audio forgery, consequently jeopardizing the reliability of digital evidence and posing threats to legal processes, journalistic activities, and other areas. This study presents a hybrid method for detecting audio copy-move forgery. Using the SuperPoint architecture, which extracts both keypoints and descriptor vectors simultaneously from images using a deep convolutional neural network, keypoints and their descriptors were obtained from Mel spectrogram images of audio recordings, and the statistical features obtained from descriptor matching were fed into various machine learning algorithms for classification. To ensure optimal performance, hyperparameters were tuned using Random Search, which revealed Random Forest as the most effective classifier for this task. Experimental results on a Turkish audio dataset show that the proposed method significantly outperforms traditional handcrafted feature-based techniques, achieving superior accuracy and F1-score values.

Keywords: Audio copy-move forgery detection, Audio forensics, Machine learning, Random forest, SuperPoint

1. Introduction

The digitalization and increased technological accessibility of today have made forgery in digital audio recordings quite common. Open-source tools like Audacity, professional tools like Adobe Audition, and even mobile applications have made it possible to alter sound waves without requiring advanced technical knowledge or skills. The ease with which audio can be manipulated has made it necessary to determine whether a recording is original. In this context, audio authentication methods discussed in the literature to verify the reliability of digital audio are basically divided into two main categories: active and passive methods (Ustubioglu et al., 2023a). Active authentication methods aim to protect the audio recording from the outset by placing a unique mark on the audio before it is transmitted on the network. These methods are generally designed to verify the integrity of the recording or to protect copyrights. Digital watermarking and digital signatures fall into the category of active authentication techniques.

Passive methods assume that there is no pre-existing mark (watermark, etc.) on the original recording. They search for signs of forgery by analyzing only the available audio data. Passive audio authentication methods are divided into two main approaches depending on which layer of the data being analyzed is focused on: container-based and content-based methods. Container-based methods detect forgery by examining inconsistencies in the digital structure of the file. This group includes integrity-checking methods that check the digital records assigned by the system from the moment the file was created (creation, modification, and access dates) and generate a unique mathematical hash of the file (such as MD5, SHA-256), and file format-based methods that examine the digital code structure and format standards of the file to determine whether the file has been manipulated. Content-based passive authentication methods focus on the physical and statistical characteristics of the audio signal, regardless of file format. These methods primarily focus on detecting splicing and copy-move attacks. Audio splicing refers to combining two or more audio tracks recorded at different times or in different environments. Each recording medium has its own ambient noise. Therefore, audio splicing forgery detection methods attempt to determine whether the audio is forged by tracking acoustic inconsistencies.

*Şeyma YÜCEL ALTAY; seyma.yucel@atauni.edu.tr

Audio copy-move forgery (ACMF), which is one of the types of passive audio forgery and is discussed in this study, is a type of forgery that aims to change the context of speech by copying a segment from an audio recording and placing it at a different position in the same recording. For example, in the Turkish sentence "Olay gecesi orada olduğumu kabul etmiyorum. Evde olduğumu kabul ediyorum." ("I do not admit that I was there on the night of the incident. I admit that I was at home."), copying the word "etmiyorum" ("do not admit") and pasting it over the word "ediyorum" ("admit") changes the context of the sentence. The resulting forged sentence becomes: "Olay gecesi orada olduğumu kabul etmiyorum. Evde olduğumu kabul etmiyorum." ("I do not admit that I was there on the night of the incident. I do not admit that I was at home").

This is an example showing that even a short-copied section can significantly alter the semantic meaning of a speech recording. Since the copied and pasted segments originate from the same recording, they usually retain similar acoustic characteristics, such as background noise, tone, frequency, amplitude, and speed. This makes detecting ACMF more difficult compared to splicing detection, where inconsistencies between different recording environment are more apparent (Su et al. 2023; Zhao et al. 2024; Ustubioglu et al. 2025). Figure 1 shows the manipulation in the Turkish sentence given as an example in the previous paragraph at the waveform level. Figure 1 shows the waveform representation of a forged audio recording involving copy-move forgery. The region marked as (a) highlights the source segment copied from one part of the forged recording, whereas the region marked as (b) shows the corresponding pasted segment inserted at another time position within the same recording.

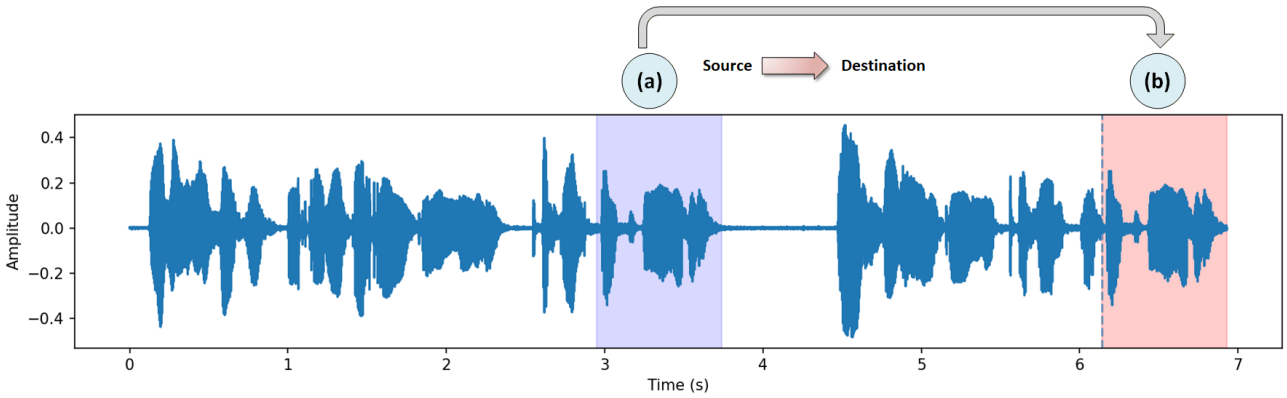


Figure 1. Waveform representation of a forged audio recording containing copy-move manipulation. (a) The copied source segment highlighted in light purple. (b) The pasted segment highlighted in pink.

In this study, we propose a machine learning-driven keypoint-based framework for audio copy-move forgery detection. The methodology is built on the premise that utilizing visual representations of audio, rather than raw audio signals, provides a balanced and effective solution in terms of computational cost and detection performance. Time-frequency domain representations are compatible with human auditory perception and facilitate the extraction of distinctive features. Therefore, this work transforms audio recordings using the Short-Term Fourier Transform (STFT) to obtain Mel spectrograms, which are time-frequency representations. On these visual representations, local features reflecting structural similarities are extracted using the SuperPoint architecture, a pre-trained feature extractor capable of simultaneously performing keypoint detection and descriptor extraction. In the final stage, the summary feature vectors derived from the feature matching process are fed as input to machine learning-based classifiers, and the forgery detection process was carried out. To the best of our knowledge, this study is among the first to investigate the use of a SuperPoint architecture, originally designed for computer vision, in combination with classical machine learning classifiers for audio copy-move forgery detection.

The main contributions of our work can be summarized as follows:

- This study introduces a SuperPoint-based audio copy-move forgery detection framework that integrates Mel spectrogram-based audio representation with deep local keypoint features, providing an alternative to conventional handcrafted keypoint-based approaches.
- It proposes a matching-based statistical representation in which SuperPoint keypoint, descriptor matching, RANSAC inlier, and confidence score statistics are encoded into a compact 8-dimensional feature vector.

- Unlike fixed-threshold-based copy-move forgery detection methods that rely only on the number of matched or inlier keypoints, the proposed framework learns the final authentic/forged decision using a machine learning classifier trained on matching-based statistical features.
- The proposed framework is systematically evaluated under multiple post-processing attacks, including compression, Gaussian noise, and median filtering, demonstrating its robustness under degraded audio conditions.

The remainder of the paper is organized as: Section 2 provides related studies on audio copy-move forgery detection. Section 3 introduces the general methods used in the study. Section 4 presents the obtained experimental results and discussion, whereas Section 5 concludes the paper.

2. Related work

Audio copy-move forgery detection has been addressed using different signal processing, deep learning, and time-frequency-based analysis strategies. Since the copied and pasted segments originate from the same audio recording, they usually share similar acoustic and environmental characteristics. Therefore, existing studies generally attempt to identify duplicated regions by analyzing similarities between audio segments or their transformed representations. Based on the feature representation and analysis strategy, existing ACMF detection studies can be broadly grouped into raw audio-based methods, window-based methods, and time-frequency representation-based methods.

Early ACMF detection studies mainly relied on handcrafted features from raw audio signals to identify traces of forgery. Within this group, a subset of studies first segments the audio into speech and silence regions using techniques such as Voice Activity Detection (VAD) or pitch-tracking algorithms like Yet Another Algorithm for Pitch Tracking (YAAPT). These methods then analyze only the speech segments, aiming to identify regions with similar feature characteristics. For instance, Yan et al. (2019) investigated similarity detection by integrating pitch and formant sequences and evaluating them using Dynamic Time Warping (DTW). In another approach, Imran et al. (2017) utilized Local Binary Pattern (LBP) histograms to Arabic speech recordings to identify forged audio segments. Wang et al. (2017) adopted an approach in which Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) features were extracted after segmenting the signal with VAD. Xie et al. (2018) introduced a multi-feature strategy. Their work combined gammatone features, pitch information, Mel-Frequency Cepstral Coefficients (MFCC), and Discrete Fourier Transform (DFT). Then, these features fed into decision tree-based classification. More recent studies by Akdeniz & Becerikli (2024a, 2024b) employed YAAPT-based segmentation together with deep learning models, including Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNNs), and Artificial Neural Networks (ANNs). Ustubioglu et al. (2022) attempted to detect forgery by extracting the audio portions of the speech recording using pitch tracking and then using the modified DCT coefficients obtained from these portions. The methods in this group focus on the parts where speech is active, reducing unnecessary comparisons and increasing detection efficiency. On the other hand, their performance largely depends on the accuracy of the VAD process; errors in distinguishing between voiced and silent frames can propagate throughout the analysis and impair overall reliability, especially in noisy environments.

An alternative approach to VAD-based methods focuses on window-based (or frame-based) analysis. Some researchers have focused on splitting the audio signal into uniform temporal segments to identify duplicate audio. For instance, Xiao et al. (2014) presented a technique that splits the audio into constant frames of time T. This approach uses fast convolutional algorithms to measure inter-frame similarity, and segments exceeding a predefined threshold are flagged as forgery traces. In another study, Su et al. (2023) used a sliding window (SW) strategy to facilitate copy-move forgery detection. In this context, they extracted Constant Q Cepstral Coefficients (CQCC) from each equally long frame and used the Pearson correlation coefficient to measure the degree of correlation between these segments. In another study by Su et al. (2022), they improved the detection of duplicate frames using Constant Q Spectral Sketches (CQSS) in conjunction with a hybrid machine learning model. This process involves taking the square mean of the logarithm of the Constant Q Transform to obtain the CQSS features. To maximize detection accuracy, a Genetic Algorithm (GA) is used to optimize the feature set before the Support Vector Machine (SVM) distinguishes between real and spurious frames and performs the final classification. When window-based studies in the literature are examined, it can be concluded that even small copy-move segments can be detected. But the correct selection of the window

size greatly affects the performance of the method. A very small window size increases the amount of data to be processed and, naturally, the computational cost. On the other hand, a very large frame size increases the probability of missing erroneous segments. In conclusion, the success of window-based methods depends on striking a critical balance between computational cost and detection sensitivity.

In recent research, studies focusing on forgery detection using time-frequency representations of audio signals, such as Mel spectrograms and cochleagrams, instead of raw audio signals, have become widespread. Some of these studies focus on extracting features and matching them using keypoint detectors and descriptors used in computer vision, while others use these images as input for deep learning methods. For example, a recent study (Ozgen & Altay, 2025) attempted to detect forgery by searching for regions on the Mel spectrogram where the feature vectors of keypoints extracted using Scale-Invariant Feature Transform (SIFT) and Features from Accelerated Segment Test (FAST) were similar. In another study (Özgen & Altay, 2025), interesting points extracted from Mel spectrogram representations of audio recordings using AGAST and HARRIS algorithms were identified with the FREAK descriptor, and similar regions were attempted to be determined using Hamming distance. In a study (Ustubioglu, 2024), after the voiced and silence parts of the audio recording were separated by VAD, the similarity between the cochleagram representations of the voiced parts was calculated using the Structural Similarity Index (SSIM). To detect forgery, the presence of pairs whose correlation value exceeds a dynamically determined threshold has been decisive. Ustubioglu et al. (2023b) presented an ACMF detection framework that combines time-frequency representations with deep learning. By converting raw audio signals into Mel spectrogram images, they fed these images as input to a Convolutional Neural Network (CNN) architecture and thus determined whether the audio was original or forged based on the output of this network. In a new approach (Arslan & Sadık, 2025) where Mel spectrogram representations are integrated with machine learning, audio recordings were first visualized by converting into Mel spectrograms, then subjected to classification tests using methods such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Random Forest (RF). In performance evaluations, the Extreme Gradient Boosting (XGBoost) algorithm surpassed other techniques.

Overall, existing ACMF detection methods have progressed from raw audio-based handcrafted features to window-based similarity analysis and, more recently, time-frequency representation-based approaches. Although visual representations such as Mel spectrograms and cochleagrams have shown promising results, existing studies generally employ either handcrafted keypoint descriptors, direct image-level classification, or conventional similarity analysis. However, models that perform deep learning directly on time-frequency representations often require relatively large and diverse training datasets, can focus on global image-level patterns, and generally provide limited interpretability regarding the decision-making process. The integration of learned keypoint extraction, descriptor matching, and machine learning-based classification remains limited in ACMF detection. This work addresses this gap by adapting the SuperPoint architecture, originally developed for computer vision, to Mel spectrogram representations of speech signals and combining it with a machine learning classifier that enables authentic/forged classification of audio recordings. In the proposed hybrid pipeline, SuperPoint is used to extract learned keypoints and descriptors, the resulting local similarities are analyzed through matching and geometric verification, and summary statistics derived from these matching patterns are used as features for machine learning classifiers. Therefore, the originality of this study lies in combining learned local visual feature extraction with geometrically verified copy-move similarity analysis and statistical feature-based classification within a unified ACMF detection framework.

3. Material and method

This section introduces the methodological framework used in the study. First, the transformation of raw audio signals into Mel spectrogram representations and preprocessing phase are discussed in detail, followed by an examination of the SuperPoint architecture. After introducing how the similarity between the extracted descriptors is calculated using SuperPoint, the feature vector generation process is discussed. Finally, the stage of classifying the audio as fake or authentic is presented.

3.1. Visualization of audio recordings

To implement the feature extraction process through the SuperPoint architecture, Short-Time Fourier Transform (STFT) based Mel spectrograms were generated to simultaneously represent the temporal and frequency components of the audio signal in the proposed method. In practice, the STFT is computed using

the Librosa library (McFee et al., 2015), which internally applies a Hann window and an efficient FFT-based implementation. Accordingly, the audio signal x[n] is framed by multiplying it with the Hann window function $w[n]$ and the STFT amplitude is obtained as:

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mH] \cdot w[n] \cdot e^{-j\frac{2\pi}{N}kn} \quad (1)$$

Here, N represents the window length and H represents the hop length. The Hann window is defined as:

$$w[n] = 0.5(1 - cos(\frac{2\pi n}{N-1})) \quad (2)$$

The obtained power spectrum ($|X(m,k)|^2$) was multiplied by the Mel filter bank ($Hp[k]$) consisting of P triangular filters suitable for the frequency perception of the human ear, and the Mel coefficients were calculated:

$$S_{mel}(m,p) = \sum_{k=0}^{N/2} |X(m,k)|^2 \cdot Hp[k] \quad (3)$$

In the final stage, the power spectrum was converted to a logarithmic unit to compress the dynamic range of the sound and increase the prominence of low-frequency components. In the experiments, a Hann window with a length of 2048 samples and a hop size of 512 samples was used, while the Mel filter bank consisted of 128 filters. The resulting Mel power spectrograms were then converted to the decibel scale using peak normalization.

3.2. Preprocessing

Following the generation of Mel-spectrograms, an approach mimicking the biological characteristics of the human visual system was implemented for the analysis of this visual data. The human eye has a high sensitivity to the luminance of light rather than its chrominance (Altay & Ulutas, 2024). Based on this biological principle, the generated spectral images were converted to the YCbCr color space.

The YCbCr color space allows for more efficient feature analysis by separating luminance information from color information. The Y channel obtained after the YCbCr color space transformation contains the luminance information for the image, whereas the Cb and Cr channels represent color differences. In SuperPoint-based keypoint and descriptor extraction, only the Y component was used, considering that geometric and structural patterns are more descriptive than color information. Figure 2 shows the Mel spectrogram image created from an audio recording and the Y component obtained after its conversion to the YCbCr space.

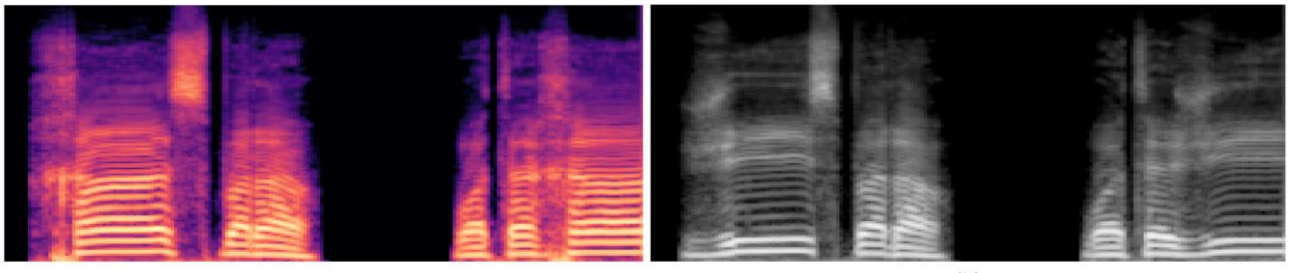


Figure 2. (a) Mel spectrogram representation of an audio (b) its derived luminance (Y) channel in the YCbCr color space

3.3. Deep feature extraction using SuperPoint

In this study, the SuperPoint architecture (Diwan et al., 2023), trained with its own self-supervised learning strategy, was used to perform feature extraction from audio spectrograms. Unlike traditional methods, SuperPoint offers an end-to-end learnable fully Convolutional Neural Network (FCNN) structure, simultaneously calculating both keypoint locations and their descriptors with a single feedforward. In this study, SuperPoint is considered not as a direct decision-making mechanism, but as a feature extraction method

aimed at revealing structural similarities in Mel spectrograms. The technical operation of the architecture is divided into two main components: Keypoint detection and description.

Keypoint detection: The SuperPoint detector consists of an FCNN that operates on a full-size input image and outputs keypoint locations. The process is formulated as follows: When an input image of dimensions $H \times W$ is defined as I, the output of the shared encoder network is $f(I)$. Then, the detection head takes this $f(I)$ data as input and produces a probability map (heat map) of dimensions $H \times W$ (DeTone et al., 2018):

$$M = Softmax(g(f(I))) \tag{4}$$

Here, g represents the function that maps the encoder output to the probability map space. By applying the softmax function, a probability distribution is obtained for each pixel value in the M map. In the final stage, the process is completed by selecting N pixels with the highest values in this probability map to determine the keypoints that will form the basis of the forgery analysis. In this study, keypoint confidence threshold (τ_{conf}) was used to enable the detection of low-energy but structurally significant regions. Additionally, a Non-Maximum Suppression (NMS) process with a radius of 4 pixels was applied to prevent unnecessary concentrations that might occurring in very close proximity.

Feature description: SuperPoint generates fixed-length descriptors that encode the local visual and spectral characteristics of each detected keypoint. In this process, the input image I (with dimensions of $H \times W$) is processed through a convolutional neural network f_θ to produce a semi-dense feature map F with dimensions of $\frac{H}{8} \times \frac{W}{8} \times D$, where $D = 256$ represents the descriptor dimensionality. The descriptor d_i for each keypoint k_i is sampled from the corresponding location via F using a two-dimensional linear interpolation (bilinear sampling) method. The resulting descriptors are normalized via L2-norm to make distance-based matching more stable in comparison processes. These descriptors represent the visual patterns corresponding to local energy distributions on the Mel spectrogram in a numerical form, enabling the detection of similar regions indicating copy-move forgery.

SuperPoint, used as the keypoint detector and descriptor extractor in this framework, was initialized with pretrained weights and operated in evaluation mode. The descriptor dimension was set to 256, the non-maximum suppression radius to 4 pixels, the keypoint confidence threshold to 0.005, and the maximum number of keypoints to 2048.

In this study, the keypoint and descriptor outputs provided by the SuperPoint architecture were not used directly for classification; instead, summary statistical features were derived by analyzing the geometric consistency of similarities between keypoints. These features were then fed as input to the machine learning-based classifier used in the next stage, forming the basis of the forgery detection process.

3.4. Similarity analysis

A similarity search was performed using the k-nearest neighbor (k-NN) algorithm to identify potential copying relationships between the extracted keypoints. To improve match quality and remove outliers unrelated to forgery, some procedures are applied including Lowe's ratio test, spatial distance constraint, and outlier elimination. SuperPoint architecture extracts keypoints, their descriptors, and confidence scores from each Mel spectrogram image, denoted as i.

$$K_i = \{k_1, k_2, k_3, \cdots, k_{N_i}\} \tag{5}$$

$$D_i = \{d_1, d_2, d_3, \cdots, d_{N_i}\} \tag{6}$$

$$C_i = \{c_1, c_2, c_3, \cdots, c_{N_i}\} \tag{7}$$

where $k_j = (x_j, y_j)$ represents the spatial coordinate of the j-th keypoint for i-th image, d_j denotes the corresponding descriptor vector, c_j is the confidence score generated by the SuperPoint detector. The total number of detected keypoints for i-th Mel spectrogram image is represented by N_i.

Lowe's ratio test: To detect candidate duplicated regions within the same Mel spectrogram image, descriptor self-matching is performed using a k-NN ratio test with $k = 3$. For a given descriptor d_p, the two nearest neighboring descriptors are examined. Let $dist_1$, $dist_2$ denote the Euclidean distances from d_p to its nearest and second-nearest neighboring descriptors, respectively. Lowe's ratio test is calculated as the ratio of $dist_1$ and $dist_2$. Matches with low discrimination and ambiguity were eliminated from the system by applying a threshold value t. The threshold t was set to 0.75, consistent with the literature, as follows:

$$dist_1/dist_2 < t. \tag{8}$$

Spatial distance constraint: In both spectrogram and natural images, pixels that are spatially very close to each other are similar due to their inherent texture. Therefore, considering every similarity as forgery would increase false positives. To prevent this, a minimum offset threshold δ is used. A match (p, q) is retained only if:

$$\|k_p - k_q\| > \delta. \tag{9}$$

In this case, δ was set to 10 pixels. Matches below this threshold were filtered out, and the remaining spatially filtered match set was denoted as M.

Outlier elimination: When performing keypoint matching on spectrograms, the algorithm sometimes matches similar but actually unrelated points due to repeating textures; these are called outliers. If a region is copied and pasted elsewhere, all points in that region are expected to be geometrically consistent. The Random Sample Consensus (RANSAC) algorithm was used to filter out these geometrically inconsistent points. For this purpose, data pairs with a re-projection error exceeding a defined threshold value (3 pixels in this study) were identified as outliers and excluded from the analysis. The remaining spatially filtered match set is denoted as M. After this step, RANSAC-based affine verification is applied to obtain the geometrically consistent inlier set I, where $I \subseteq M$.

3.5. Generation of feature vector

In the process following the descriptor similarity analysis, the keypoint matches detected inlier were examined in detail to serve as an input to classification algorithms. Directly using individual descriptor matches encourages the use of fixed decision thresholds, as in studies (Ozgen & Altay, 2025, Özgen & Altay, 2025). Instead, to establish a more dynamic decision strategy, the structural and geometric properties of the matched regions were analyzed in this study, and a fixed-length feature vector was created for each Mel-spectrogram image. The generated feature vector of i-th Mel spectrogram image, denoted as F_i, is an 8-dimensional numerical representation.

$$F_i = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8]^T \tag{10}$$

$$\begin{aligned}
f_1 &= N_i = |K_i| \\
f_2 &= |M_i| \\
f_3 &= |I_i| \\
f_4 &= |I_i|/(|M_i| + \varepsilon) \\
f_5 &= |I_i|/(|K_i| + \varepsilon) \\
f_6 &= (1/N_i) \sum_{j=1}^{N_i} c_j \\
f_7 &= \sqrt{\left(1/N_i\right) \sum_{j=1}^{N_i} (c_j - f_6)^2} \\
f_8 &= 1, if\ |I_i| > \eta; otherwise\ f_8 = 0
\end{aligned} \tag{11}$$

The first feature (f_1) corresponds to the total number of detected keypoints (N_i). The second feature represents the number of initial descriptor matches obtained after applying a self K-nearest neighbor ratio test with a threshold of 0.75, followed by excluding spatially close regions (after the spatial distance constraint step). This feature indicates candidate duplicate regions within the same spectrogram. The third feature represents the number of geometrically consistent matches (inliers) remaining after RANSAC-based affine verification. The fourth feature, calculated based on the third feature, is defined as the inlier ratio, calculated as the ratio of verified inliers to the number of spatially filtered candidate matches. ε in Equation (11) was chosen to avoid a

possible division by zero error. The fifth feature measures the density of geometrically verified matches relative to the total number of detected keypoints. The sixth and seventh features correspond to the mean and standard deviation of the keypoint confidence scores produced by the SuperPoint detector, respectively, as given in Equation (7). Finally, a weak binary indicator is included as the eighth feature. This indicator provides a coarse prior to assist the classification stage. If the number of verified inliers exceeds five ($\eta = 5$), f_8 is set to one; otherwise, it is set to zero. The vector derived from these eight features is then given as input to the classifier.

$$\hat{y}_i = g(F_i) \tag{12}$$

where $g(.)$ denotes a machine learning classifier and F_i is the extracted feature vector. The predicted class label is represented by $\hat{y}_i \in \{0, 1\}$. We use "0" to represent an authentic audio recording. "1" is used as a forged audio recording.

Because the resulting 8-dimensional feature vector is structured as an input set for machine learning algorithms rather than raw descriptive similarity, this approach avoids being bound to a fixed decision threshold. This allows the model to develop a flexible decision-making mechanism that can learn complex patterns in the dataset and differentiate between original and manipulated audio samples.

3.6. Classification

The statistical data obtained from the feature extraction and filtering stages (total number of keypoints, inlier ratio, match density, etc.) were converted into an 8-dimensional feature vector. This vector was then fed as input to Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), k-NN, Gradient Boosting Classifier (GBC), and Extreme Gradient Boosting (XGB) classifiers to make forgery decisions. To avoid diverting the focus of the study, a detailed breakdown of the mathematical models based on these classification models is not included in this text. Detailed technical references regarding the implementation procedures of the mentioned algorithms are available in the scikit-learn documentation (Pedregosa et al., 2011). To measure the accuracy and resistance of the method to post-processing attacks, a Stratified Group K-Fold cross-validation (K=5) strategy was followed. The model was trained only on data without attacks. Then, it was subjected to robustness tests under attack scenarios such as lossy compression (32/64 kbps), noise insertion (20/30 dB SNR), and median filtering. This approach scientifically proves the system's ability to distinguish traces of forgery, even in noisy environments. The process, from the generation of Mel spectrograms and SuperPoint feature extraction to the classification of 8-dimensional feature vectors, is shown in Figure 3.

4. Results and discussion

To evaluate the success of the method, commonly used performance metrics from the literature were employed: Accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC). Here, AUC was adopted as a threshold-independent evaluation metric to assess the discriminability between forged and original audio samples. It is derived from the ROC curve, which shows the relationship between the true positive and false positive rates at different decision thresholds, and an AUC value close to 1 indicates discriminatory performance. In this study, AUC values were calculated using the predicted class probabilities obtained from the classifier outputs. The calculation of precision, recall, F1-score, and accuracy was based on four basic quantities: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). While TP accounts for the forged audio segments accurately captured by the system, FP signifies the original samples that were incorrectly labeled as fraudulent. TN represents the correct classification of original spectrogram representations as authentic, while FN corresponds to forged spectrograms that are misclassified as original.

The mathematical definitions of these evaluation metrics are provided in Equations (13-16) (Ozgen & Altay, 2025).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{13}$$

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

$$Recall = \frac{TP}{TP+FN} \tag{15}$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

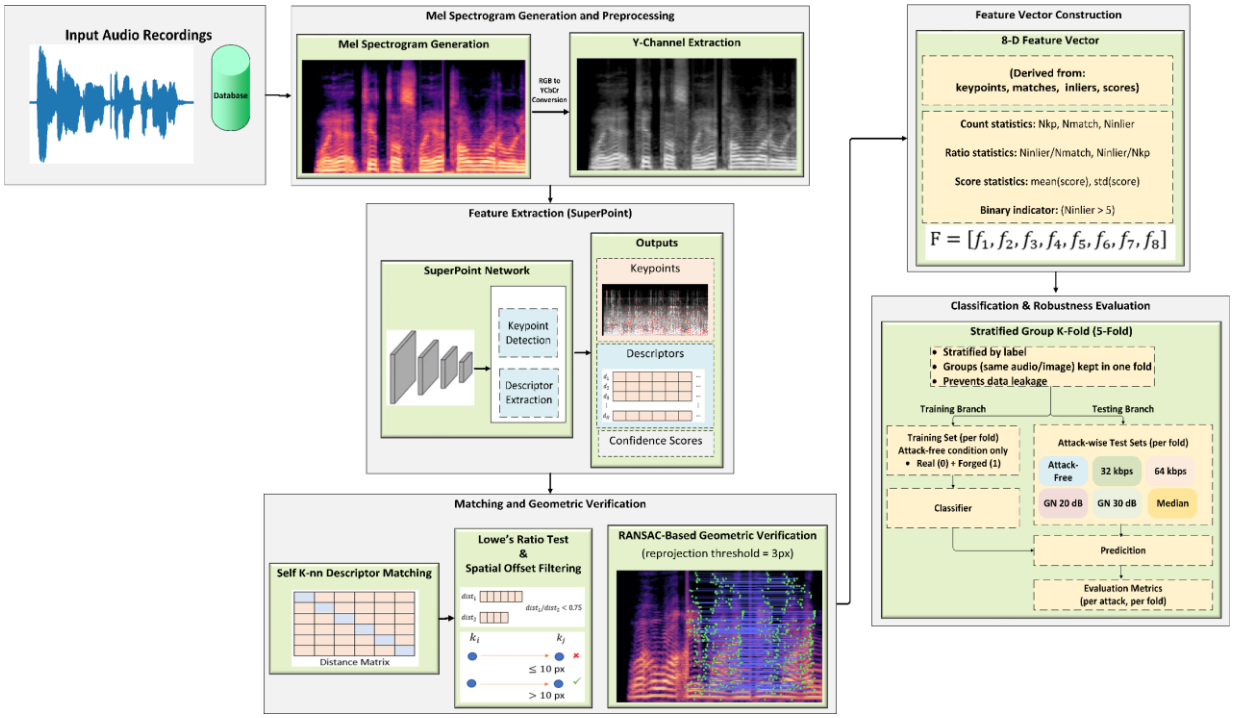


Figure 3. Block diagram of the proposed method

The experiments were conducted using the dataset (Ustubioglu et al., 2024), accessible at https://ceng2.ktu.edu.tr/~csrg. This new corpus consists of Turkish speech recordings captured across three diverse acoustic environments: an office, a cafeteria, and a quiet room, with 200 samples per environment. The dataset includes 1046 semantically coherent forgeries derived from the original recordings. To evaluate robustness, each forged sample was subjected to five distinct post-processing attacks (compression (32/64kbps), noise insertion (20/30dB SNR), and median filtering), resulting in a total of 6276 forged files (comprising 1046 initial forgeries plus five attacked versions per file).

To provide a fair and leakage-free evaluation, the dataset was partitioned using a stratified group-based cross-validation strategy. In the training stage, the classifier used only under the attack-free case, which includes real recordings and clean forged recordings. In each fold, the feature vectors extracted from the attack-free samples were utilized as the training data. A five-fold Stratified Group K-Fold strategy was employed to preserve the class distribution while preventing samples originating from the same base recording from appearing simultaneously in both the training and test sets. The group labels were assigned according to the original recording identity. Therefore, the attack-free and attacked versions of the same sample were kept within the same group, avoiding data leakage between training and testing. After training the classifier on the attack-free training subset, testing was performed separately for each scenario, including the attack-free case, 32 kbps compression, 64 kbps compression, 20 dB Gaussian noise, 30 dB Gaussian noise, and median filtering. For each test scenario, the same test indices determined by the corresponding fold were used. Thus, the robustness of the proposed method was evaluated under post-processing attacks without exposing the classifier to attacked samples during training.

4.1. Hyperparameter optimization protocol

To ensure a fair comparison among different classifiers, hyperparameter optimization was conducted using the Random Search strategy within the training folds. For each model, the search space was defined according to commonly used parameter ranges in the literature. Random Search was performed with 20 iterations, where each iteration evaluated a randomly selected combination of hyperparameters from the predefined search space. In this process, a 3-fold internal cross-validation was applied within each training fold of the main 5-fold stratified group cross-validation procedure with shuffling enabled and a fixed random state of 42. The optimal hyperparameter configurations obtained for each classifier are summarized in Table 1. Using these optimized configurations, performance evaluation was conducted under a stratified group k-fold validation scheme. The resulting performance metrics, including accuracy, precision, recall, F1-score, and AUC, are reported in Figure 4.

Table 1. Hyperparameters obtained via Random Search

Classifier	Optimized Hyperparameters	Search Space	Definition
SVM	C = 1; kernel =linear; γ =scale	C ∈ {0.1, 1, 10, 100}; kernel ∈ {linear, RBF}; γ ∈ {scale, auto}	C controls regularization; kernel defines the decision boundary; γ controls the influence range of samples.
LR	C = 1	C ∈ {0.01, 0.1, 1, 10}	**C** controls the trade-off between model complexity and regularization.

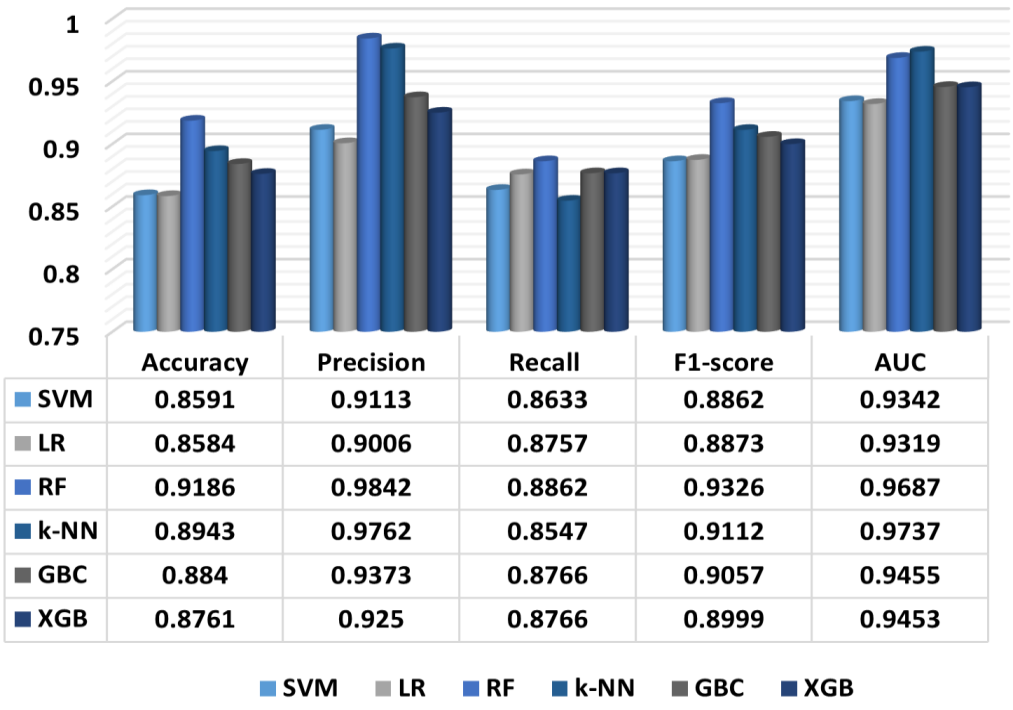
RF	n_estimators =200;
	max_depth = None;
	min_samples_split = 2
		n_estimators ∈ {100, 200, 400};
		max_depth ∈ {None, 10, 20, 40};
		min_samples_split ∈ {2, 5, 10}
			n_estimators sets the number of trees; max_depth limits tree depth; min_samples_split controls node splitting.

k-NN	k = 5;
	weights = distance;
	metric = manhattan
		k ∈ {3, 5, 7, 9, 11};
		weights ∈ {uniform, distance};
		metric ∈ {euclidean, manhattan}
			n_neighbors defines neighborhood size; weights set neighbor contribution; metric defines distance measure.

GBC	n_estimators = 200;
	learning_rate = 0.05;
	max_depth = 3
		n_estimators ∈ {100, 200, 300};
		learning_rate ∈ {0.01, 0.05, 0.1};
		max_depth ∈ {3, 5}
			n_estimators sets boosting stages; learning_rate scales updates; max_depth controls learner complexity.

XGB	n_estimators = 200;
	max_depth = 3;
	learning_rate = 0.1;
	subsample = 1.0
		n_estimators ∈ {200, 400};
		max_depth ∈ {3, 6};
		learning_rate ∈ {0.05, 0.1};
		subsample ∈ {0.8, 1.0}
			n_estimators defines boosting rounds; max_depth controls tree complexity; learning_rate scales updates; subsample improves robustness.

As seen in Figure 4, the RF model achieved the highest overall accuracy (0.9186) among the compared classifiers. It achieved a high recall value (0.9842), exhibiting a low false alarm rate, while also maintaining a competitive recall value (0.8862), which is crucial for effectively detecting forged samples. Therefore, the F1 score (0.9326) obtained by RF demonstrates a strong balance between sensitivity and recall. Although the k-NN classifier yielded the highest AUC value (0.9737), it achieved a lower recall value compared to RF. Overall, considering all evaluation criteria together, RF exhibited the most balanced and robust performance and was therefore selected as the most suitable classifier for the proposed forgery detection framework.



	Accuracy	Precision	Recall	F1-score	AUC
SVM	0.8591	0.9113	0.8633	0.8862	0.9342
LR	0.8584	0.9006	0.8757	0.8873	0.9319
RF	0.9186	0.9842	0.8862	0.9326	0.9687
k-NN	0.8943	0.9762	0.8547	0.9112	0.9737
GBC	0.884	0.9373	0.8766	0.9057	0.9455
XGB	0.8761	0.925	0.8766	0.8999	0.9453

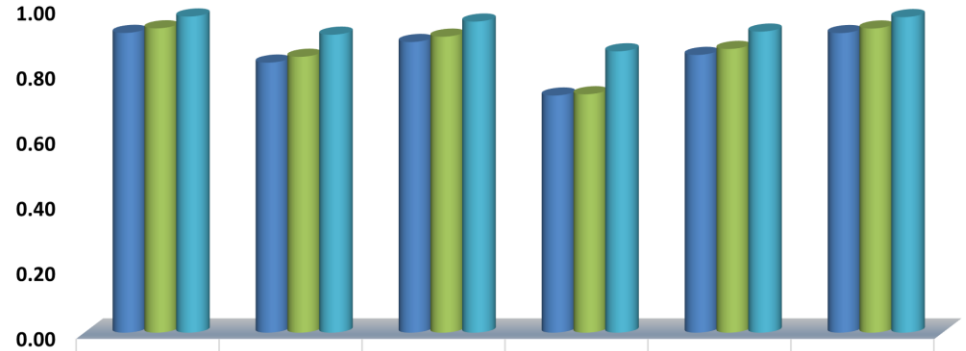
Figure 4. Classifier performances with optimized hyperparameters for attack-free scenario.

The fact that RF achieves better performance than other classifiers can be explained by the structure of the proposed feature set. Features extracted from Mel spectrogram images using SuperPoint are statistical features consisting of key points, descriptor matching, and geometrically consistent matches. These features may have nonlinear relationships with forgery labels. For example, a forgery sample is characterized not by a single feature, but by the combined behavior of various indicators such as the number of matches, the number of inliers, and the consistency ratios obtained after RANSAC. RF is a classifier that can model nonlinear decision boundaries. Its ability to capture interactions between features without requiring strong parametric assumptions also makes it suitable for such feature representations. Its ensemble structure also provides robustness to noisy or weakly informative features, which may explain its superior performance compared with the other tested classifiers.

4.2. Performance of the selected SuperPoint-RF model

After comparing the optimized classifiers, RF classifier was selected as the final classification model due to its superior overall performance. Therefore, the final performance of the proposed SuperPoint-RF framework was evaluated in detail using standard classification metrics, including accuracy, precision, recall, F1-score, and AUC. Figure 5 includes these results. When the figure is examined in general terms, it reveals that the proposed method works quite stably under attack-free and some mild disruptive operations, but performance drops significantly under low-SNR noise. The model exhibits the highest performance in the attack-free condition (Accuracy≈0.92, F1-score≈0.93, AUC≈0.97). This shows that the extracted features can distinguish copy-move forgery with high accuracy under attack-free conditions. When compression attacks are examined, a relative decrease in performance is observed when the loss is higher at 32 kbps (Accuracy≈0.83), while the metrics improve significantly at 64 kbps (Accuracy≈0.89, AUC≈0.95). Gaussian noise attack with 20 dB SNR constitutes the most challenging scenario for the model. While the accuracy and F1-score values dropped to the 0.73 range, the AUC value was obtained at approximately 0.86. Since the noise distorts the fine structural patterns on the Mel spectrogram, this leads to a weakening of keypoint matching and geometric consistency. In contrast, when the SNR was increased to 30 dB, performance improved significantly. Accuracy increased to 0.8511, F1-score to 0.8694, and AUC to 0.9225. This result indicates that copy-move traces are better preserved on the spectrogram because the noise effect is more limited at higher SNR levels. The results obtained under median filtering exhibit performance quite close to the attack-free case (Accuracy≈0.92, AUC≈0.97).

As shown in the Figure 5, a relatively higher performance degradation was observed under low-SNR noise attack. The proposed method is based on keypoint matches obtained via time-frequency representation, and the addition of noise can distort the local spectral structures used to identify repeated regions. Specifically, it can alter local texture patterns and introduce misleading spectral details. As a result, the number of reliable keypoint matches may decrease, while the number of false or unstable matches may increase. Therefore, compared to some other post-processing operations, low-SNR noise attacks may have a stronger negative impact on the proposed detection pipeline.



	Attack-free	Compression 32 kbps	Compression 64 kbps	Gaussian N. SNR=20 dB	Gaussian N. SNR=30 dB	Median Filtering
Accuracy	0.9186	0.8275	0.8907	0.7266	0.8511	0.9180
F1-score	0.9326	0.8453	0.9070	0.7305	0.8694	0.9320
AUC	0.9687	0.9129	0.9532	0.8621	0.9225	0.9669

Figure 5. Performance metrics obtained using the Random Forest classifier

To assess the reliability of the reported results, 95% confidence intervals were computed from the fold-wise scores obtained using five-fold stratified group cross-validation. A 95% confidence interval quantifies the uncertainty around the estimated mean performance and, under repeated sampling, is expected to contain the true mean performance in 95% of such intervals (Hazra, 2017). Hence, narrow confidence intervals indicate more accurate and stable performance predictions across validation folds. To provide additional insight into the consistency of the proposed framework, Table 2 presents the fold-wise F1-score results. The relatively small variation observed across folds under different attack scenarios indicates stable behavior across different train–test partitions. Building upon these fold-wise results, Table 3 reports the mean, standard deviation, and 95% confidence interval for Accuracy, F1-score, and AUC. The mean values are already visualized in Figure 5. This table complements the results in the mentioned figure by showing the variability and statistical stability of the proposed method across validation folds.

Table 2. Fold-wise F1-score results obtained from five-fold stratified group cross-validation

Attacks	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Attack-free	0.9330	0.9343	0.9286	0.9270	0.9400
Compression 32 kbps	0.8283	0.8310	0.8740	0.8361	0.8571
Compression 64 kbps	0.8860	0.9152	0.9340	0.8730	0.9266
Gaussian N. SNR=20 dB	0.7278	0.7095	0.7574	0.7012	0.7566
Gaussian N. SNR=30 dB	0.8860	0.8618	0.8813	0.8571	0.8610
Median filtering	0.9169	0.9397	0.9447	0.9215	0.9373

Table 3. Reliability analysis of the proposed method under different attack scenarios

Attacks	Metric	Mean	Std.	95% CI
Attack-free	Accuracy	0.9186	0.0058	[0.9113, 0.9258]
Attack-free	F1-score	0.9326	0.0051	[0.9262, 0.9390]
Attack-free	AUC	0.9687	0.0027	[0.9654, 0.9721]
Compression 32 kbps	Accuracy	0.8275	0.0201	[0.8025, 0.8524]
Compression 32 kbps	F1-score	0.8453	0.0196	[0.8209, 0.8697]
Compression 32 kbps	AUC	0.9129	0.0142	[0.8952, 0.9306]
Compression 64 kbps	Accuracy	0.8907	0.0290	[0.8547, 0.9267]
Compression 64 kbps	F1-score	0.9070	0.0263	[0.8743, 0.9397]
Compression 64 kbps	AUC	0.9532	0.0179	[0.9309, 0.9754]
Gaussian N. SNR=20 dB	Accuracy	0.7266	0.0218	[0.6996, 0.7536]
Gaussian N. SNR=20 dB	F1-score	0.7305	0.0260	[0.6982, 0.7628]
Gaussian N. SNR=20 dB	AUC	0.8621	0.0083	[0.8518, 0.8723]
Gaussian N. SNR=30 dB	Accuracy	0.8511	0.0128	[0.8352, 0.8671]
Gaussian N. SNR=30 dB	F1-score	0.8694	0.0132	[0.8531, 0.8858]
Gaussian N. SNR=30 dB	AUC	0.9225	0.0114	[0.9084, 0.9366]
Median filtering	Accuracy	0.9180	0.0143	[0.9002, 0.9358]
Median filtering	F1-score	0.9320	0.0121	[0.9170, 0.9471]
Median filtering	AUC	0.9669	0.0056	[0.9599, 0.9738]

The low standard deviations and narrow 95% confidence intervals obtained for all three metrics in the attack-free scenario demonstrate that the method exhibits consistent performance under this condition across validation folds. For compression attacks, the confidence intervals remain relatively narrow, indicating that the method maintains stable discriminatory ability despite performance degradation. 32 kbps compression results in greater degradation compared to 64 kbps compression. However, the presented variability indicates that this degradation is consistent across folds. Gaussian noise (specifically SNR = 20 dB) was the most challenging attack scenario, resulting in lower reliability statistics compared to others. This can be attributed to the distortion of local time-frequency structures in the Mel spectrogram, which affects keypoint detection and descriptor matching. In contrast, the results improve under SNR = 30 dB, showing better stability under moderate noise conditions. Median filtering has the least effect on the proposed method. Its low standard deviations and narrow confidence intervals, which are close to the attack-free case, indicate that the matching-based feature representation is robust against smoothing-based post-processing. Overall, Table 3 confirms that the proposed method provides stable and reliable performance across most attack scenarios, although severe Gaussian noise and low-bitrate compression remain more challenging.

Figure 6 presents the feature importance scores of the RF classifier. The bars show the mean importance values over the five stratified group folds, while the error bars indicate the standard deviation. The most influential features are F3, F2, and F5, which correspond to the number of inliers, the number of matches, and the inlier-to-keypoint ratio, respectively. This result suggests that the classifier primarily benefits from matching- and geometric consistency-related information, which is consistent with the proposed ACMF detection strategy.

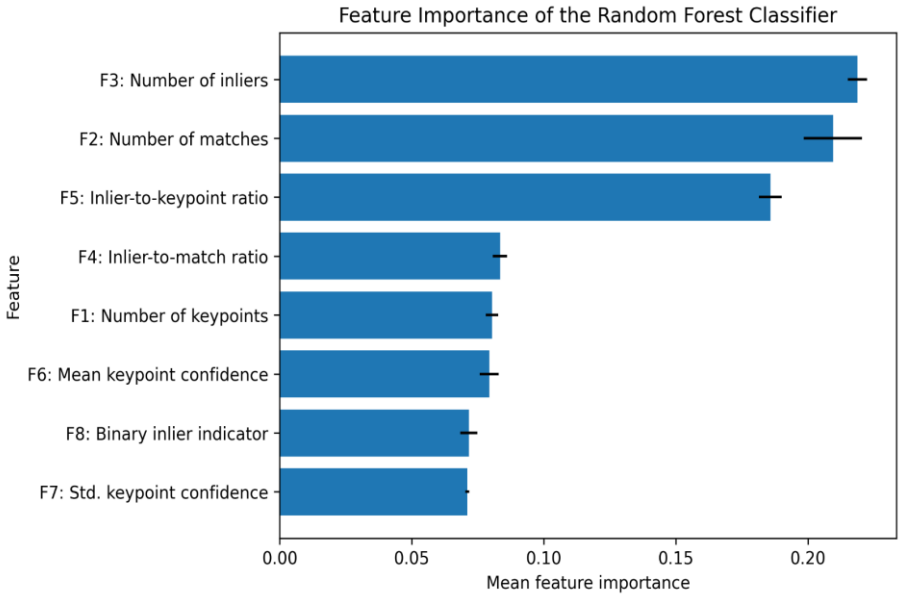


Figure 6. Feature importance of the RF classifier. Bars indicate the mean importance over five folds, and horizontal error bars show the standard deviation.

4.3. Contribution of RANSAC-Derived Features

This sub-section investigates the contribution of the RANSAC verification stage. An ablation experiment was conducted by removing all RANSAC-derived features (f_3, f_4, f_5, f_8 described in Section 3.5) from the feature vector while keeping the remaining pipeline unchanged. The obtained F1-score results are shown in Table 4.

As shown in Table 4, removing RANSAC-derived features led to a consistent decrease in performance in most scenarios. The largest decrease was observed under Gaussian noise attacks. For example, the F1 score under 20 dB Gaussian noise dropped from 0.7305 to 0.6739, while under 30 dB Gaussian noise, the F1 score dropped from 0.8694 to 0.8351. In contrast, only minor performance drops were observed under compression and median filtering attacks. These findings demonstrate that the features derived from RANSAC significantly contribute to the robustness of the proposed framework, especially under noisy conditions.

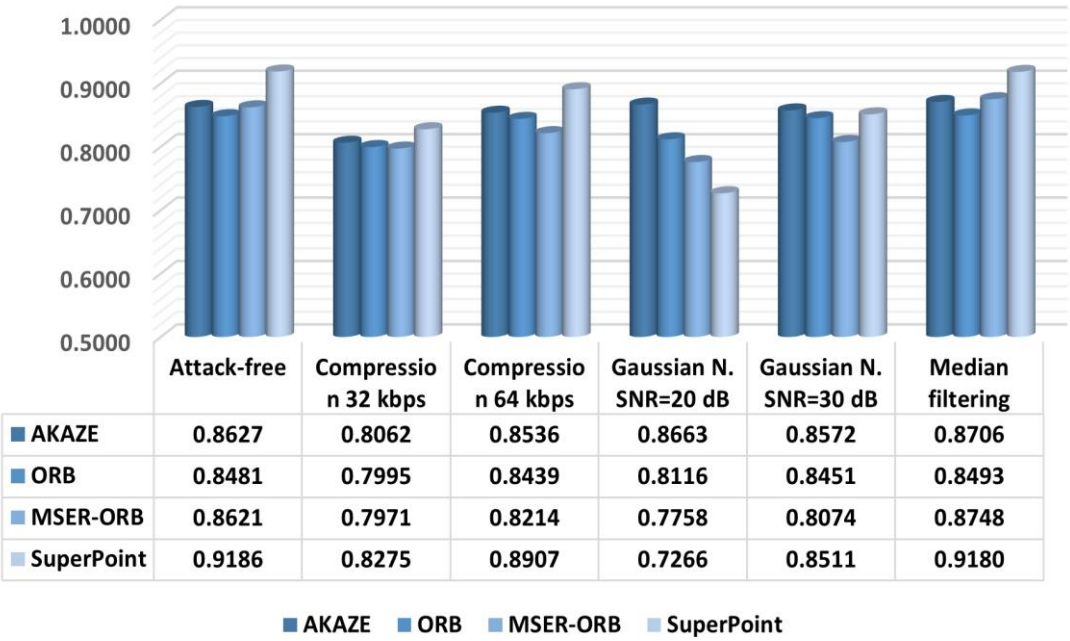
The observed behavior can be explained by the ability of RANSAC-derived features to capture the geometric consistency of matched keypoints. Under Gaussian noise attacks, random distortions create false keypoints and increase the probability of mismatches. In such cases, features derived after the RANSAC process become valuable because they help distinguish genuine copy-move traces from geometrically inconsistent matches. In contrast, compression and median filtering attacks largely preserve the fundamental spectro-temporal structure of the spectrogram. This leads to fewer spurious matches and therefore less reliance on information derived from RANSAC.

Table 4. Ablation study of RANSAC-derived features based on F1-score across different attack scenarios

	Attack-free	Comp. 32 kbps	Comp. 64 kbps	Gaussian N. SNR=20 dB	Gaussian N. SNR=30 dB	Median filtering
Without RANSAC Features	0.9216	0.8394	0.9028	0.6739	0.8351	0.9318
Full Feature Set	0.9326	0.8453	0.9070	0.7305	0.8694	0.9320

4.4. Comparison with alternative keypoint extraction methods

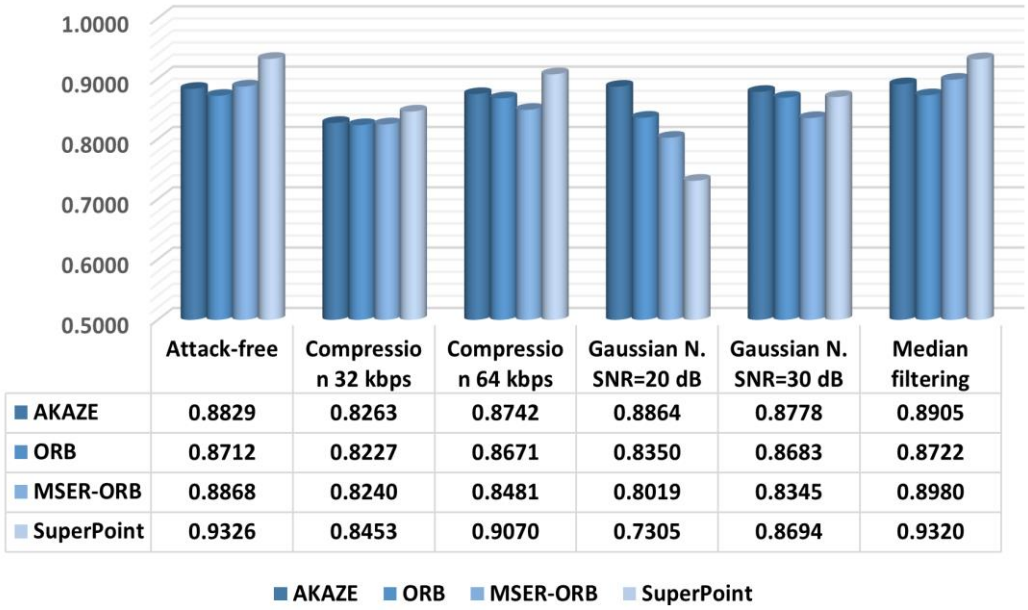
To evaluate the contribution of the SuperPoint-based keypoint extraction strategy, the proposed method was compared with several alternative keypoint extraction approaches, including Accelerated KAZE (AKAZE) (Alcantarilla and Solutions, 2011), Oriented FAST and Rotated BRIEF (ORB) (Rublee et al., 2011), and Maximally Stable Extremal Regions (MSER)-ORB (Matas et al., 2004). In these baseline configurations, AKAZE was used with Modified Local Difference Binary (MLDB) descriptors, while ORB was applied with 2048 maximum features and the Harris score criterion. Since MSER is only a region detector and does not inherently provide descriptors, the detected stable MSER regions were first converted into keypoints using the centroid and bounding-box size of each region, and ORB was then employed to compute the corresponding descriptors.



	Attack-free	Compression 32 kbps	Compression 64 kbps	Gaussian N. SNR=20 dB	Gaussian N. SNR=30 dB	Median filtering
AKAZE	0.8627	0.8062	0.8536	0.8663	0.8572	0.8706
ORB	0.8481	0.7995	0.8439	0.8116	0.8451	0.8493
MSER-ORB	0.8621	0.7971	0.8214	0.7758	0.8074	0.8748
SuperPoint	0.9186	0.8275	0.8907	0.7266	0.8511	0.9180

Figure 7. Comparative robustness evaluation of AKAZE, ORB, MSER-ORB, and the proposed SuperPoint-based method under various post-processing attacks in terms of Accuracy.

For all baseline methods, descriptor matching was performed within the same image using self k-nearest-neighbor matching and Lowe's ratio test. Very close matches were discarded using a minimum spatial offset constraint, and the remaining correspondences were verified using RANSAC-based affine estimation. In this experiment, the classifier was kept fixed as the optimized RF model, and only the keypoint extraction method was changed. The evaluation was conducted under different attack conditions. Figures 7-9 present comparison of these keypoint extraction methods in terms of accuracy, F1-score, AUC, respectively. In the attack-free scenario, SuperPoint achieved the highest performance with an accuracy of 0.9186, F1-score of 0.9326, and AUC of 0.9687, outperforming AKAZE, ORB, and MSER-ORB. This indicates that SuperPoint provides more discriminative keypoint representations when no post-processing attack is applied. Under compression attacks, SuperPoint also provided the best results, particularly under 64 kbps compression, where it achieved F1-score of 0. 9070. This suggests that the learned keypoint representation of SuperPoint is relatively robust against moderate compression artifacts. However, the mentioned performance metrics decreased under 32 kbps compression, which indicates that severe compression may degrade the local structures required for reliable keypoint matching (accuracy=0.8275, F1-score=0.8453, AUC=0.9129). Under median filtering, SuperPoint again achieved the best performance with accuracy of 0.9180, F1-score of 0.9320, AUC of 0.9669, followed by MSER-ORB and AKAZE. This result indicates that SuperPoint is effective in preserving discriminative matching patterns under nonlinear filtering operations. We said that, unlike other scenarios, SuperPoint is challenging, especially at SNR = 20 dB. AKAZE reached the highest F1 score with 0.8864, while SuperPoint achieved an F1-score of 0.7305 against this attack. In terms of AUC values (see Figure 9), SuperPoint experienced a significant decrease for the same condition, unlike in attack-free, compression, and median filtering scenarios, recording an AUC of 0.8621 under Gaussian noise (20 dB SNR). These results show that the pre-trained SuperPoint representation is more susceptible to severe noise distortion in this experimental environment than methods like AKAZE, ORB, and MSER.



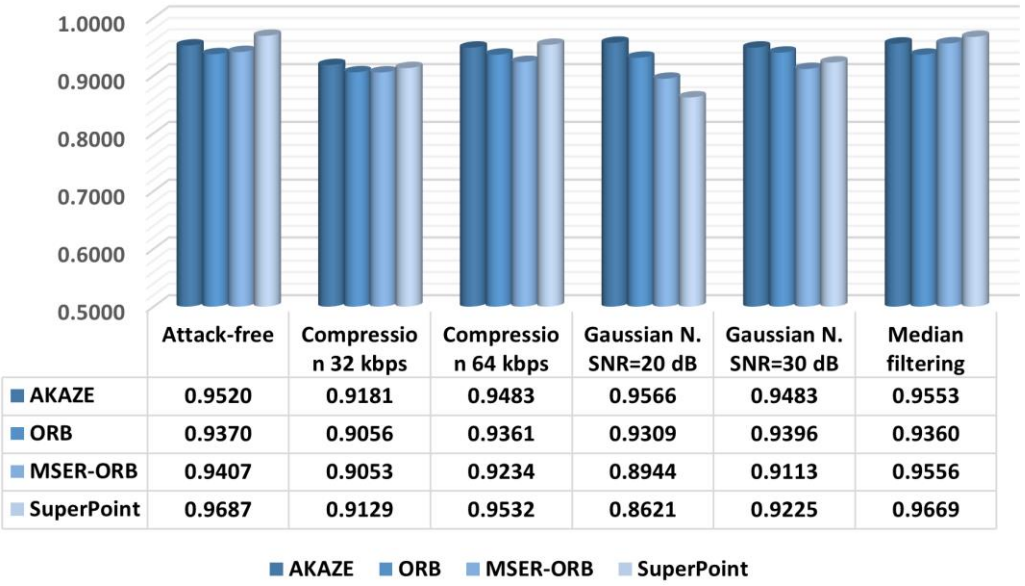
	Attack-free	Compression 32 kbps	Compression 64 kbps	Gaussian N. SNR=20 dB	Gaussian N. SNR=30 dB	Median filtering
AKAZE	0.8829	0.8263	0.8742	0.8864	0.8778	0.8905
ORB	0.8712	0.8227	0.8671	0.8350	0.8683	0.8722
MSER-ORB	0.8868	0.8240	0.8481	0.8019	0.8345	0.8980
SuperPoint	0.9326	0.8453	0.9070	0.7305	0.8694	0.9320

Figure 8. Comparative robustness evaluation of AKAZE, ORB, MSER-ORB, and the proposed SuperPoint-based method under various post-processing attacks in terms of F1-score.

In light of these results, it can be said that pre-trained and handcrafted local feature extraction methods exhibit disparate resilience profiles against post-processing attacks. Although SuperPoint achieves the best detection performance in most scenarios, further improvements can be obtained by developing approaches that combine the discriminative capability of pre-trained descriptors with the noise robustness of handcrafted feature representations.

4.5. Comparison with existing methods

This sub-section compares the performance of our study with approaches found in the literature on the same dataset. Table 5 contains the accuracy results of the proposed Random Forest (RF) classifier in comparison with existing methods reported in the literature. Among these methods are VAD-based methods based on handcrafted features such as Pitch and Formant (Yan et al., 2019), LBP (Imran et al., 2017), DCT-SVD (Wang et al., 2017), DFT (Huang et al., 2020), RASTA-PLP (Hatipoglu et al., 2024), and the SIFT-FAST-FREAK (Ozgen & Altay, 2025) approach, which uses a hybrid of traditional keypoint-based approaches obtained from Mel spectrograms.



	Attack-free	Compression 32 kbps	Compression 64 kbps	Gaussian N. SNR=20 dB	Gaussian N. SNR=30 dB	Median filtering
AKAZE	0.9520	0.9181	0.9483	0.9566	0.9483	0.9553
ORB	0.9370	0.9056	0.9361	0.9309	0.9396	0.9360
MSER-ORB	0.9407	0.9053	0.9234	0.8944	0.9113	0.9556
SuperPoint	0.9687	0.9129	0.9532	0.8621	0.9225	0.9669

Figure 9. Comparative robustness evaluation of AKAZE, ORB, MSER-ORB, and the proposed SuperPoint-based method under various post-processing attacks in terms of AUC.

The table shows the performance results not only for the non-attack case but also for various attacks. Table 5 shows that the proposed method achieves an accuracy of 0.9186 in the attack-free scenario, almost the same as the SIFT-FAST-FREAK method (0.9192). Under compression attacks (32 kbps and 64 kbps), the performance of the proposed method is superior to all reference methods. In particular, under the 32 kbps compression condition, it provides a significant advantage over the SIFT-FAST-FREAK method (0.7187) with an accuracy of 0.8275. Similarly, in the 64 kbps scenario, the proposed method exhibited the highest performance with an accuracy of 0.8907. These results reveal that the method has high robustness against compression. The results obtained under median filtering are quite close to the attack-free scenario, showing that the proposed method largely maintains its performance with 0.9180 accuracy. On the other hand, under the Gaussian noise scenario with 20 dB SNR, the accuracy of the proposed method decreased compared to the attack-free scenario. The accuracy of this attack is 0.7266. According to these results, it is seen that the SIFT-FAST-FREAK method provides higher accuracy under the 20 dB noise condition. Although a noticeable improvement in performance was observed at 30 dB SNR, the SIFT-FAST-FREAK method still showed relatively better performance. Yet the method is still more successful against this challenging attack compared to other reference studies.

Table 5. Comparison of accuracy scores between the proposed method and existing approaches on the KTUCengAudioForgerySet under attack-free and post-processing conditions

	Pitch and formant (Yan et al., 2019)	**LBP** (Imran et al., 2017)	**DCT-SVD** (Wang et al., 2017)	**DFT** (Huang et al., 2020)	**RASTA-PLP** (Hatipoglu et al., 2024)	**SIFT-FAST-FREAK** (Ozgen & Altay, 2025)	**This study**
Attack-free	0.2412	0.5073	0.5188	0.5298	0.5300	0.9192	0.9186
Compression 32 kbps	0.1719	0.4192	0.4635	0.4885	0.4890	0.7187	0.8275
Compression 64 kbps	0.1859	0.3682	0.4617	0.4957	0.4960	0.8834	0.8907
Median filtering	0.2369	0.5067	0.5237	0.5328	0.5320	0.9162	0.9180
Gaussian N. SNR=20 dB	0.2096	0.3779	0.4970	0.5243	0.5240	0.7801	0.7266
Gaussian N. SNR=30 dB	0.2309	0.3779	0.4927	0.5322	0.5320	0.8663	0.8511

Table 6. Comparison of F1-scores between the proposed method and existing approaches on the KTUCengAudioForgerySet under attack-free and post-processing conditions

	Pitch and formant (Yan et al., 2019)	**LBP** (Imran et al., 2017)	**DCT-SVD** (Wang et al., 2017)	**DFT** (Huang et al., 2020)	**RASTA-PLP** (Hatipoglu et al., 2024)	**SIFT-FAST-FREAK** (Ozgen & Altay, 2025)	**This study**
Attack-free	0.2407	0.5647	0.5934	0.5234	0.5230	0.9373	0.9326
Compression 32 kbps	0.1097	0.4435	0.5245	0.4589	0.4590	0.7414	0.8453
Compression 64 kbps	0.1377	0.3635	0.5221	0.4707	0.4710	0.9069	0.9070
Median filtering	0.2332	0.5639	0.5992	0.5279	0.5280	0.9348	0.9320
Gaussian N. SNR=20 dB	0.1833	0.3794	0.5669	0.5152	0.5150	0.8087	0.7305
Gaussian N. SNR=30 dB	0.2224	0.3794	0.5617	0.5270	0.5270	0.8918	0.8694

Table 6 shows a comparison of the F1-scores obtained from the reference methods presented in Table 5 with the F1-scores of the method proposed in this study under different attack conditions. In the attack-free case, this study exhibits high performance with an F1-score of 0.9326. Although the SIFT-FAST-FREAK method gives a very close result (0.9373), the proposed method is significantly more successful than all the classical methods in the literature (Pitch and Formant, LBP, DCT-SVD, DFT, RASTA-PLP). In compression scenarios

(32 kbps and 64 kbps), the performance of this study is maintained at a high level (0.8453 and 0.9070, respectively). Especially at 64 kbps compression, the proposed method gives almost the same results as SIFT-FAST-FREAK, while providing a clear advantage compared to other traditional methods. Under median filtering, this method stands out as one of the most successful, with an F1-score of 0.9320. In scenarios with added Gaussian noise with an SNR of 20 dB and 30 dB, the F1-score of the proposed method decreases to 0.7305 and 0.8694, respectively, compared with the attack-free scenario. Nevertheless, the proposed method still achieves significantly better results than VAD-based approaches such as Pitch and Formant and LBP.

The SIFT-FAST-FREAK-based method and this work are keypoint-based forgery detection methods. However, performance differences have been observed between them in the face of attacks. This difference can be attributed to the characteristics of the feature representations used. Under compression attacks, the dominant spectral-temporal structures of the spectrogram are largely preserved, despite the loss of high-frequency spectral details. Since learned descriptors are designed to capture discriminative structural patterns, reliable matches can be established between duplicated regions. In contrast, additive Gaussian noise introduces random distortions throughout the spectrogram, altering local intensity distributions. Therefore, such distortions can lead to performance degradation by reducing descriptor consistency and matching reliability. On the other hand, the SIFT-FAST-FREAK basic model combines hand-made gradient and intensity-based descriptors, which have been reported to exhibit robustness against certain photometric distortions and noise variations (Lowe, 2004; Alahi et al., 2012). This may explain the superior performance of the reference study under the severe Gaussian noise conditions observed in our experiments.

In summary, this study demonstrated a significant performance improvement compared to VAD-based methods, including Pitch and Formant (Yan et al., 2019), LBP (Imran et al., 2017), DCT-SVD (Wang et al., 2017), DFT (Huang et al., 2020), and RASTA-PLP (Hatipoglu et al., 2024), in the attack-free scenario and under compression, filtering, and noise attacks. It was also found to achieve superior or similar performance to the SIFT-FAST-FREAK (Ozgen & Altay, 2025) method, except for noise attacks, which use traditional computer vision keypoint detectors and descriptors together. The fact that this study can make a forgery decision without being bound to a fixed threshold value, unlike the SIFT-FAST-FREAK-based method, suggests improved robustness and within-dataset generalization capability across the evaluated attack scenarios. It should be noted, however, that the generalization demonstrated in this study is limited to the evaluation protocol adopted on the KTUCengAudioForgerySet dataset. Although the dataset includes recordings collected under three different environmental conditions, providing variability in acoustic characteristics and recording scenarios, all experiments were conducted within a single dataset. Hence, the reported results primarily indicate within-dataset generalization rather than cross-dataset generalization. Therefore, the robustness of the proposed framework under substantially different speakers, languages, recording devices, and acquisition protocols cannot be inferred directly from the present experiments.

A direct numerical comparison with recent deep learning-based methods could not be included because, to the best of our knowledge, these methods were not evaluated on the same dataset, manipulation types, attack scenarios, and validation protocol used in this study. Moreover, the lack of publicly available source code or trained models for some related approaches prevents their reliable reproduction under identical experimental conditions. Therefore, this study avoids direct performance comparison across incompatible settings and instead provides a qualitative positioning of the proposed method. In this regard, unlike fully end-to-end deep learning models, the proposed method offers a more understandable and lower-cost structure that combines local feature matching and statistical classification steps. Future work will include the implementation of CNN- and Transformer-based baselines on the same dataset and evaluation protocol.

4.6. Processing time analysis

The proposed methodology was implemented on a PC equipped with an Intel Core i7 processor and 16 GB of RAM. The software environment utilized Python 3.10, with OpenCV 4.11 for image processing and scikit-learn 1.5.1 for machine learning model development and evaluation. In addition to analyzing the classification performance, the study also examined the computational cost and processing time of the proposed method. The computational cost was evaluated using the same stratified group k-fold validation protocol adopted in the performance evaluation. Therefore, the attack-free case, 32 kbps and 64 kbps compression, 20 dB and 30 dB Gaussian noise, and median filtering scenarios were jointly included in the evaluation. For each scenario, 1046 forged Mel spectrogram images and the same set of 600 real Mel spectrogram images were considered. Since

the real images were processed only once, whereas the forged images were processed separately for each of the six scenarios, a total of 6276 forged images (1046×6) and 600 real images were processed, corresponding to 6876 Mel spectrogram images overall.

Table 7. Computational cost and processing time of the proposed method for the complete KtuCengAudioForgerySet (Attack-free and Attacked samples)

Processing stage	Audio-to-Mel spectrogram conversion	Image loading and preprocessing	Keypoint detection and descriptor extraction	Feature vector generation and classification	Overall processing time
Total time (s)	8832.1068	2461.0992	39,985.9488	1800.3948	53,079.5496
Average time per sample (s)	1.2844	0.3579	5.8152	0.2618	7.7195

Table 7 presents the computational cost and processing time of the proposed method for the complete KtuCeng AudioForgerySet, including both attack-free and attacked samples. The overall processing time was 53,079.5496 s, corresponding to an average of 7.7195 s per sample. Among all stages, keypoint detection and descriptor extraction required the highest computational cost, with a total time of 39,985.9486 s and an average time of 5.8152 s per sample. This indicates that the SuperPoint-based feature extraction stage is the most time-consuming component of the proposed pipeline. This result is expected because SuperPoint is a deep learning-based model that simultaneously performs keypoint detection and descriptor generation on each Mel spectrogram image. In contrast, feature vector generation and classification required only 0.2618 s per sample, showing that the classification stage introduces a relatively low computational burden. These results suggest that the main computational cost of the proposed method arises from deep keypoint and descriptor extraction, while the subsequent feature vector construction and classification stages are computationally efficient.

Table 8 presents the computational efficiency and attack-free detection performance of the keypoint extraction methods evaluated in Section 4.4. The attack-free scenario was considered to assess the intrinsic discriminative capability of the evaluated keypoint extraction methods without the influence of post-processing operations.

Table 8. Computational comparison of the proposed method and alternative keypoint extraction methods

	AKAZE	ORB	MSER-ORB	SuperPoint
Average time per sample (s)	2.1001	1.9781	2.0211	7.7195
F1-score (attack-free condition)	0.8829	0.8712	0.8868	0.9326

The results show that ORB achieved the lowest processing time (1.9781 s per sample), followed by MSER-ORB (2.0211 s) and AKAZE (2.1001 s). While these conventional keypoint-based methods offer relatively low computational costs, they achieve lower detection performance than the proposed method. In contrast, the proposed SuperPoint-based method achieved the highest F1-score (0.9326), exceeding those of AKAZE (0.8829), ORB (0.8712), and MSER-ORB (0.8868). This performance gain was obtained at the expense of increased computational time, requiring 7.7195 s per sample, which is approximately 3.7–3.9 times higher than the processing times of the conventional methods. These results reveal a clear trade-off between computational efficiency and detection accuracy. While ORB, AKAZE, and MSER-ORB provide faster execution, the proposed SuperPoint-based framework demonstrates superior detection performance under attack-free conditions, making it a suitable choice for forensic applications where detection reliability is prioritized over processing speed.

5. Conclusion

This study presents an audio copy-move forgery detection method based on deep keypoint features obtained from Mel spectrograms. Unlike end-to-end deep classification approaches, the SuperPoint architecture was employed as a feature extraction module rather than as a direct classifier. In this way, structurally meaningful keypoints and descriptors were obtained from Mel spectrogram images, and the final decision was performed using classical machine learning-based classifiers.

Experimental results have shown that the proposed feature set can effectively characterize copy-pasted portions in audio recordings. In the classifier comparison, the RF classifier offered balanced and consistent performance

in terms of accuracy, F1-score, and AUC. These results demonstrate that the matching-based features extracted from SuperPoint keypoints provide distinctive information for distinguishing forged recordings from original recordings. When evaluations under different post-processing attacks were examined, the method achieved stable detection performance under the attack-free case and under relatively mild distortion operations. In particular, under 32 kbps and 64 kbps compression attacks, the proposed method achieved higher accuracy values than the compared methods in the literature. Similarly, the results obtained under median filtering were close to the attack-free results; this indicates that the method is only minimally affected by smoothing-based post-processing operations. On the other hand, the low-SNR Gaussian noise scenario was the most challenging condition for the proposed method. Since the proposed framework relies on keypoint detection, descriptor matching, and geometric consistency analysis, additional noise can reduce the stability of local spectral structures and weaken the reliability of keypoint matching.

The 95% confidence interval analysis confirmed the stability of the selected SuperPoint-RF configuration across the evaluation folds. Furthermore, comparisons using AKAZE, ORB, and MSER-ORB configurations in the keypoint detection and description phases showed that SuperPoint was more discriminative in attack-free, compression, and filtering for detecting copy-move forgeries in Mel spectrogram images.

Although the proposed method performs well in most experimental settings, it has some limitations. Strong noise and severe post-processing can make the extracted keypoints less stable and reduce the reliability of descriptor matching. In addition, very short copied segments may produce fewer distinctive keypoints, making forgery detection more difficult. Moreover, processing very long audio recordings may also increase computational cost and memory requirements, as the proposed approach involves both generating time–frequency representations and performing keypoint detection and matching over the resulting images. Future studies will focus on improving the noise tolerance of the proposed framework. In this context, more robust time-frequency representations may be investigated to reduce the negative effect of additive noise on keypoint detection and descriptor matching. In addition, deep-learning-based feature extraction strategies or hybrid feature learning strategies may be explored to obtain more stable representations under noisy conditions.

Declaration of ethical code

The authors of this article declare that the materials and methods used in this study do not require ethics committee approval and/or legal-special permission.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

Akdeniz, F., & Becerikli, Y. (2024a). Detecting audio copy-move forgery with an artificial neural network. *Signal, Image and Video Processing*, *18*(3), 2117-2133. https://doi.org/10.1007/s11760-023-02856-w

Akdeniz, F., & Becerikli, Y. (2024b). Recurrent neural network and long short-term memory models for audio copy-move forgery detection: a comprehensive study. *The Journal of Supercomputing*, *80*(12), 17575-17605. https://doi.org/10.1007/s11227-024-05960-x

Altay, S. Y., & Ulutas, G. (2024). Biometric watermarking schemes based on QR decomposition and Schur decomposition in the RIDWT domain. *Signal, Image and Video Processing*, *18*(3), 2783-2798. https://doi.org/10.1007/s11760-023-02949-6

Arslan, M., & Sadık, Ş. A. (2025). Audio copy-move forgery detection with machine learning methods. *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*, *26*(2), 132-149. https://doi.org/10.18038/estubtda.1624909

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012, June). Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 510-517). IEEE. https://doi.org/10.1109/CVPR.2012.6247715

Alcantarilla, P. F., & Solutions, T. (2011). Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell, 34(7)*, 1281-1298. https://doi.org/10.5244/C.27.13

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 224-236). https://doi.org/10.48550/arXiv.1712.07629

Diwan, A., Kumar, D., Mahadeva, R., Perera, H. C. S., & Alawatugoda, J. (2023). Unveiling copy-move forgeries: Enhancing detection with SuperPoint keypoint architecture. *IEEE Access*, *11*, 86132-86148. https://doi.org/10.1109/ACCESS.2023.3304728

Hatipoğlu, A., Üstübioğlu, B., Üstübioğlu, A., & Ulutaş, G. (2024). Detecting audio forgery with Rasta-PLP. In *2024 32nd Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE. https://doi.org/10.1109/SIU61531.2024.10601048

Hazra, A. (2013). Using the confidence interval confidently. *Journal of Thoracic Disease, 9(10)*, 4125-4130. https://doi.org/10.21037/jtd.2017.09.14.

Huang, X., Liu, Z., Lu, W., Liu, H., & Xiang, S. (2019). Fast and effective copy-move detection of digital audio based on auto segment. *International Journal of Digital Crime and Forensics (IJDCF)*, *11*(2), 47-62. https://doi.org/10.4018/IJDCF.2019040104

Imran, M., Ali, Z., Bakhsh, S. T., & Akram, S. (2017). Blind detection of copy-move forgery in digital audio forensics. *IEEE Access*, *5*, 12843-12855. https://doi.org/10.1109/ACCESS.2017.2717842

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60(2),* 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing, 22(10)*, 761-767. https://doi.org/10.1016/j.imavis.2004.02.006

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in Python. In *Proceedings of the 14th Python in Science Conference*, (pp. 18–24). https://doi.org/10.25080/Majora-7b98e3ed-003

Ozgen, E., & Altay, S. Y. (2025). Detecting audio copy-move forgeries on mel spectrograms via hybrid keypoint features. *Applied Sciences*, *15*(21), Article 11845. https://doi.org/10.3390/app152111845

Özgen, E., & Altay, Ş. Y. (2025). Audio copy–move forgery detection on mel-spectrograms using AGAST, Harris, and FREAK. In *2025 16th International Conference on Electrical and Electronics Engineering (ELECO)* (pp. 1-5). IEEE. https://doi.org/10.1109/ELECO69582.2025.11329280

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825-2830. https://doi.org/10.48550/arXiv.1201.0490

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision* (pp. 2564-2571). https://doi.org/ 10.1109/ICCV.2011.6126544

Su, Z., Li, M., Zhang, G., Wu, Q., Li, M., Zhang, W., & Yao, X. (2022). Robust audio copy-move forgery detection using constant Q spectral sketches and GA-SVM. *IEEE Transactions on Dependable and Secure Computing*, *20*(5), 4016-4031. https://doi.org/10.1109/TDSC.2022.3215280

Su, Z., Li, M., Zhang, G., Wu, Q., & Wang, Y. (2023). Robust audio copy-move forgery detection on short forged slices using sliding window. *Journal of Information Security and Applications*, *75*, 103507. https://doi.org/10.1016/j.jisa.2023.103507

Ustubioglu, B., Küçükuğurlu, B., & Ulutas, G. (2022). Robust copy-move detection in digital audio forensics based on pitch and modified discrete cosine transform. *Multimedia Tools and Applications, 81(19)*, 27149-27185. https://doi.org/10.1007/s11042-022-13035-3

Ustubioglu, B., Tahaoglu, G., & Ulutas, G. (2023a). Detection of audio copy-move-forgery with novel feature matching on Mel spectrogram. *Expert Systems with Applications*, *213*, 118963. https://doi.org/10.1016/j.eswa.2022.118963

Ustubioglu, A., Ustubioglu, B., & Ulutas, G. (2023b). Mel spectrogram-based audio forgery detection using CNN. *Signal, Image and Video Processing*, *17*(5), 2211-2219. https://doi.org/10.1007/s11760-022-02436-4

Ustubioglu, B. (2024). An attack-independent audio forgery detection technique based on cochleagram images of segments with dynamic threshold. *IEEE Access*, *12*, 82660-82675. https://doi.org/10.1109/ACCESS.2024.3409543

Ustubioglu, B., Tahaoglu, G., Ayaz, G. O., Ustubioglu, A., Ulutas, G., Cosar, M., Karabulut, M., & Kılıc, M. (2024). KTUCengAudioForgerySet: a new audio copy-move forgery dataset. *2024 47th International Conference on Telecommunications and Signal Processing (TSP)* (pp. 123-129). IEEE. https://doi.org/10.1109/TSP63128.2024.10605983

Ustubioglu, B., Tahaoglu, G., Ustubioglu, A., Ulutas, G., & Kilic, M. (2025). A novel audio copy move forgery detection method with classification of graph-based representations. *IEEE Access, 13*, 22029-22054. https://doi.org/10.1109/ACCESS.2025.3535840

Wang, F., Li, C., & Tian, L. (2017). An algorithm of detecting audio copy-move forgery based on DCT and SVD. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)* (pp. 1652-1657). IEEE. https://doi.org/10.1109/ICCT.2017.8359911

Xiao, J. N., Jia, Y. Z., Fu, E. D., Huang, Z., Li, Y., & Shi, S. P. (2014). Audio authenticity: Duplicated audio segment detection in waveform audio file. *Journal of Shanghai Jiaotong University (Science)*, *19*(4), 392-397. https://doi.org/10.1007/s12204-014-1515-5

Xie, Z., Lu, W., Liu, X., Xue, Y., & Yeung, Y. (2018). Copy-move detection of digital audio based on multi-feature decision. *Journal of Information Security and Applications*, *43*, 37-46. https://doi.org/10.1016/j.jisa.2018.10.003

Yan, Q., Yang, R., & Huang, J. (2019). Robust copy–move detection of speech recording using similarities of pitch and formant. *IEEE Transactions on Information Forensics and Security*, *14*(9), 2331-2341. https://doi.org/10.1109/TIFS.2019.2895965

Zhao, W., Zhang, Y., Wang, Y., & Zhang, S. (2024). An Audio Copy-Move Forgery Localization Model by CNN-Based Spectral Analysis. *Applied Sciences, 14(11)*, 4882. https://doi.org/10.3390/app14114882