Protecting Copyright Ownership via Identification of Remastered Music in Radio Broadcasts

Pasindu Marasinghe[1], Lakshman Jayaratne[1], Manjusri Wickramasinghe[1], and Shakya Abeytunge[1]

[1]Affiliation not available

January 8, 2024

Protecting Copyright Ownership via Identification of Remastered Music in Radio Broadcasts

Pasindu Marasinghe[1], Lakshman Jayaratne[2], Manjusri Wickramasinghe[3], Shakya Abeytunge[4]

***Abstract*—In radio broadcasting, the crucial task of monitoring becomes evident for protecting musical work copyrights and ensuring the fair distribution of royalties. Manual monitoring, due to its time-consuming and unreliable nature, necessitates an automated approach. The challenges in automated monitoring arise mainly from the practice of broadcast stations remastering songs before airing. These alterations introduce complexities that complicate the identification process for existing music identification techniques. This paper tackles this challenge by exploring the feasibility of employing computer vision techniques on STFT spectrograms from ongoing audio streams. The objective is to identify similar spectrogram representations by comparing them with previously registered key features extracted from STFT spectrograms generated for original song tracks. This aims to unveil the identity of content being broadcast on radio and, consequently, safeguard the rightful ownership of copyrighted songs. The proposed approach achieved an accuracy of over 97% for tempo alterations up to 20% and over 95% accuracy for pitch alterations up to 20%.**

***Index Terms*—broadcast monitoring, music identification, audio fingerprinting**

I. Introduction

ONE of the primary objectives of radio stations, aimed at increasing revenue, is the expansion of their daily listener base. In pursuit of enhanced listener engagement, radio stations consistently broadcast music playlists that resonate with their audience. Broadcasting a song entails the acquisition of rights, requiring the radio station to remit a predetermined amount to the song's rightful owner. In adherence to the Intellectual Property Act of Sri Lanka [1], royalties must be duly paid to the original artists whenever a song is aired on a radio channel. Therefore, it is essential to keep track of the songs that are played daily by each and every radio station.

Although the daily playlists, curated by individual radio stations, serve as potential sources for future royalty distributions, a systematic and standardized approach to the royalty payment process necessitates the oversight of all radio stations in the nation by a third-party organization. This responsibility is commonly delegated to Performing Rights Organizations (PRO) [2]. Given the extensive number of radio channels and the vast repertoire of songs broadcasted daily, these organizations recognize that manual monitoring is neither feasible nor conducive to maintaining high quality. Consequently, they are considering the adoption of automated audio monitoring systems.

Automated radio monitoring faces significant challenges for various reasons. Radio stations frequently modify the original song compositions by incorporating additional audio elements like commercials and dialogues. Consequently, end-users do not receive the original songs in their unaltered form. Moreover, certain segments of radio broadcasts may suffer damage or introduce noise during transmission, intensifying the complexity of music identification in radio broadcasts [3]. Additionally, radio channels often engage in remastering the original musical content, adjusting the tempo to align with their schedules. This practice results in a loss of synchronization between the time and frequency domains, further complicating content identification [4]. Specifically, these time-frequency distortions can be modelled through time scaling (linear speed change), altering both duration and pitch through resampling, time stretching that modifies only duration, and pitch shifting for changing pitch.

To identify remastered music in radio broadcasts, one can leverage the existing body of literature on cover song identification and music similarity measures. These methods segment songs into smaller chunks and convert them into compact audio clip embeddings, often referred to as fingerprints, instead of using high-dimensional raw audio data to enhance the efficiency of the identification process. However, directly applying these methods to recognize remastered music in radio broadcasts encounters challenges. Time constraints emerge from the necessity to compare an ongoing radio broadcast with a database containing more than twenty thousand songs, prompting the need for modifications to existing methods to effectively handle continuous radio streams.

In 2018, Senevirathna and Jayaratne [5] introduced a technique for identifying content in radio broadcasts. This method involves parallel monitoring of the radio stream and generating fingerprints through the hashing of spectral peaks in Short Time Fourier Transform (STFT) spectrograms. Despite its advantages, such as enhanced retrieval speed using look-up tables and high accuracy compared to existing methods, this approach is limited in its ability to address the changes in time and frequency domains often present in remastered music broadcasts on radio. While effective in handling noise introduced during radio transmissions and mixed commercials and dialogues, the method exhibits vulnerability to alterations in tempo and pitch.

In this study, we utilize computer vision techniques to determine the identity of remastered music broadcasted by radio stations. The process involves segmenting the ongoing audio stream into fixed-length chunks, generating a sequence of STFT spectrogram images for each chunk. Subsequently, Scale Invarient Feature Transform (SIFT) descriptors are computed from each spectrogram image to uniquely identify the frame corresponding to the STFT spectrogram. The generated SIFT

descriptors for the considered audio chunk are then compared with a pre-registered database containing SIFT descriptors of known original songs. This facilitates the identification of the musical work in the corresponding audio chunk. Leveraging the stability of SIFT features, the proposed algorithm exhibits high discrimination and robustness. Notably, this algorithm represents the first attempt to concurrently address changes in the time and frequency domain in the field of identifying remastered music in radio broadcasts.

The structure of this paper is as follows. Section II reviews related work in this research area and outlines various approaches to achieve different levels of robustness. Section III describes the proposed music identification algorithm, and the experiments conducted to verify its performance are presented in Section IV. Finally, Section VI concludes the paper.

II. Related Work

The initial segment of this section delves into the existing literature on music audio identification techniques that can be recognized as potential solutions for identifying remastered songs. The latter part outlines approaches specifically tailored to safeguard copyright ownership in radio broadcasts.

Music audio identification systems have evolved significantly, progressing from direct hash comparisons to enhanced robustness against various audio distortions [6]. Originally leveraging domain knowledge in music and signal processing concepts for audio fingerprinting algorithms [7], [8], contemporary approaches integrate advanced machine learning [9]–[13] and computer vision techniques [4], [14]–[18], yielding highly robust music identification systems.

In 2002, J. Haitsma *et al.* [7] employed a comprehensive set of features, including Fourier coefficients, Mel-frequency Cepstral Coefficients (MFCC), and others, with dimensionality reduction techniques. Although the model demonstrated resilience against noise additions, re-sampling, band-pass filtering, and up to 4% time stretching, it was vulnerable to time scaling, pitch shifting, and various encoding schemes.

C. Wang *et al.* introduced an industrial-strength audio search algorithm, laying the groundwork for Shazam [8]. The algorithm identifies spectral peaks of STFT spectrogram generated for the audio signal and generates fingerprints by considering pairs of these peaks. Notably, this algorithm exhibits robustness against quantization effects, filtering, noise, and significant compression. However, akin to the approach in [7], this method is susceptible to temporal and pitch changes. J. Six *et al.* [19] later enhanced this method by utilizing a Constant-Q spectrogram to achieve invariance to pitch shifting and employing triplets of spectral peaks to achieve invariance to time-scale modulation.

Machine learning-based techniques for music identification have shown improved accuracy. B. Arcas *et al.* introduced a low-powered music identification system [9] integrated into the Google Pixel phone lineup[1], utilizing a Convolutional Neural Networks (CNN) to create fingerprints. In contrast, B. Suárez *et al.* [20] employed an LSTM-based sequence-to-sequence autoencoder architecture to generate audio fingerprints from MFCC spectrograms. Despite demonstrating some robustness to temporal and pitch changes, these neural network-based approaches are not sample-efficient and require substantial training data.

Recent studies have leveraged contrastive learning to improve the efficacy of audio fingerprinting. In 2020, Z. Yu *et al.* introduced a self-supervised approach for audio fingerprinting based on contrastive learning [10]. Nevertheless, the performance of this proposed algorithm has not been benchmarked against state-of-the-art music identification algorithms. Furthermore, recent investigations [11]–[13] employing contrastive learning have evaluated their approaches under a restricted range of deformations, omitting assessments for pitch and temporal changes.

Utilizing computer vision techniques for music identification entails treating the spectrogram derived from a raw audio document as an image and applying object recognition methods to analyze that image. The music identification system presented by Y. Ke *et al.* [14] employs Haar wavelet-like filters to extract local descriptors from the spectrogram through pairwise boosting. While this method demonstrated robustness against physically introduced noise, it is susceptible to time-varying distortions, and there are no reported experimental results related to this aspect.

In contrast, Baluja *et al.* initially partitioned the spectrogram into smaller spectral images and subsequently decomposed these images using Haar wavelet. Audio fingerprints were then acquired through binary quantization of the retained significant wavelet components. However, this method demonstrated only limited robustness against 10% time-scale modulations and exhibited slight resistance under 2% speed changes.

In 2010, Zhu *et al.* introduced a novel audio fingerprinting method [16] utilizing computer vision techniques applied to STFT spectrograms. They introduced the application of SIFT descriptors in the music identification domain, and their method demonstrated resilience to both time-stretching and pitch shifting. In 2015, they extended their work [4] by combining their method with Locality Sensitive Hashing (LSH) to enhance the efficiency of the fingerprint matching process. However, they do not present a mechanism to verify the validity of the matches they produce. Rather, the method they propose designates the best-matched song in the database as the valid match, even if the query song is not registered in the database beforehand.

Recently, Williams et al. proposed a method utilizing Oriented FAST and rotated BRIEF (ORB) features due to their known robustness and efficiency in extracting local image features. The music signal is initially transformed into a STFT spectrogram image, and subsequently, ORB is employed to compute the image and extract features using ORB descriptors. This method demonstrates robustness against distortions such as pitch and speed changes, which can pose challenges for other existing algorithms. However, it's important to note that the use of a Brute-Force matcher for matching descriptors between the music query sample and the database of spectrogram images may lead to slower retrieval performance, despite the speed gain achieved by ORB feature extraction.

Some approaches have been specifically developed to ad-

[1] https://pixel.google/business/products

dress copyright ownership protection in the context of monitoring radio broadcasts. In 2018, Senevirathna *et al.* introduced a music identification algorithm focused on identifying Sinhala songs in radio broadcasts [5]. The algorithm utilised STFT representations and extracted local spectral peak values from five frequency bins selected in the mid-range of the frequency axis. It demonstrated a 97% accuracy in common scenarios encountered in radio broadcasts, such as the mixing of the audio stream with commercials and dialogues, as well as the addition of noise to the signal due to external factors. Nevertheless, the algorithm showed sensitivity to common challenges in remastering, particularly tempo and pitch alterations.

Recently, Htun and Oo introduced an audio fingerprinting method [21] that employs a space-saving approach by using a binary representation of MFCC features extracted from original songs for efficient retrieval. The method exhibited robust performance under various signal distortions, including Hard Clip, Hard Overdrive, Medium Overdrive, Soft Clip, and Soft Overdrive, as well as white noise addition. However, its performance surpassed the thresholding error rate when subjected to more than 4% pitch shifting and linear speed changes.

In brief, the methods used in music identification present viable solutions for identifying remastered audio within radio broadcasts. These methods fall into different categories, including those utilizing domain knowledge, machine learning-based techniques, and computer vision-based methods. Notably, computer vision-based methods that focus on identifying local features within spectrograms have shown efficacy in addressing challenges related to temporal and pitch changes, which are complex for other existing approaches to resolve. On the other hand, the existing methods to protect copyright ownership by automatically monitoring audio streams for remastered songs show limitations in robustness to time and frequency domain alterations. The proposed method, which utilises computer vision techniques, showcases higher accuracy in identifying remastered music while maintaining robustness against tempo and pitch alterations.

III. Proposed Method

In the proposed method, we frame the remastered music identification problem as an object recognition task and address it using computer vision techniques. Various algorithms and techniques are employed to extract audio features, create audio descriptors, and match them against stored descriptors. The entire process consists of the following two stages.

1) Song Registration
2) Song Identification

In order to identify remastered songs broadcast on radio, the system needs to have previously registered all possible songs played on the radio station island-wide. This process is referred to as song registration. In this phase, a set of distinguished features is extracted from the high-dimensional audio file of the original version of each song and stored in a database to facilitate song identification in the later stage. As shown in Figure 1, each original song undergoes three main steps in the song registration stage.

The pre-processing step transforms the audio file into an image representation, and in the feature extraction step, computer vision techniques are applied to extract unique features specific to each audio file. The descriptor storing step converts the extracted features into a format suitable for easy storage and retrieval. These stored features are then utilized later in the matching step, which is part of the song identification stage.

In the second stage of song identification, the ongoing radio stream is partitioned into fixed-size segments (hereafter referred to as query audio frames). Each query audio frame undergoes two steps: pre-processing and feature extraction, mirroring the steps employed in the song registration stage. Following this, the extracted features of each query audio frame are compared with the registered features of the original songs. In the post-processing step, the algorithm identifies the best-matching original song for each query audio frame. The remastered music identification pipeline is depicted in Figure 1, and each step will be thoroughly discussed in the subsequent subsections.

A. Splitting

This step only occurs in the song identification stage. In contrast to the song registration stage, which concentrates on processing a single audio file of an original song at a time to extract its features, the song identification stage automates the process by partitioning the ongoing radio stream into 40-second intervals, known as query audio frames. Given the absence of information about the timestamps for the start and end of each song in the broadcast list, this segmentation approach is employed. Each query audio frame undergoes pre-processing and feature extraction steps before advancing to the matching and post-processing steps, ultimately determining the content included in the query audio frame.

The decision to segment into 40-second intervals strikes a balance, ensuring that the duration is not excessively lengthy, which could result in multiple songs being encompassed within a single frame, potentially complicating the identification process (given that typical song lengths range from 2 to 5 minutes). Simultaneously, the 40-second duration prevents the query audio frame from being too short, a scenario that might prompt the system to frequently segment the stream and execute all the steps to identify its content.

B. Pre-processing

The pre-processing step is a shared component in both the song registration and song identification stages. The key distinction lies in the nature of the audio input processed. During song registration, the entire audio file of a song serves as the input, while in the song identification stage, a segmented portion of the radio stream (referred to as the query audio frame) is utilized.

Pre-processing is a crucial preliminary step before the audio input enters the feature extraction phase. Its primary goals are to reduce the high dimensionality of the audio input and obtain a robust representation that facilitates subsequent phases. The pre-processing steps encompass converting the time-domain waveform of the audio input into a frequency

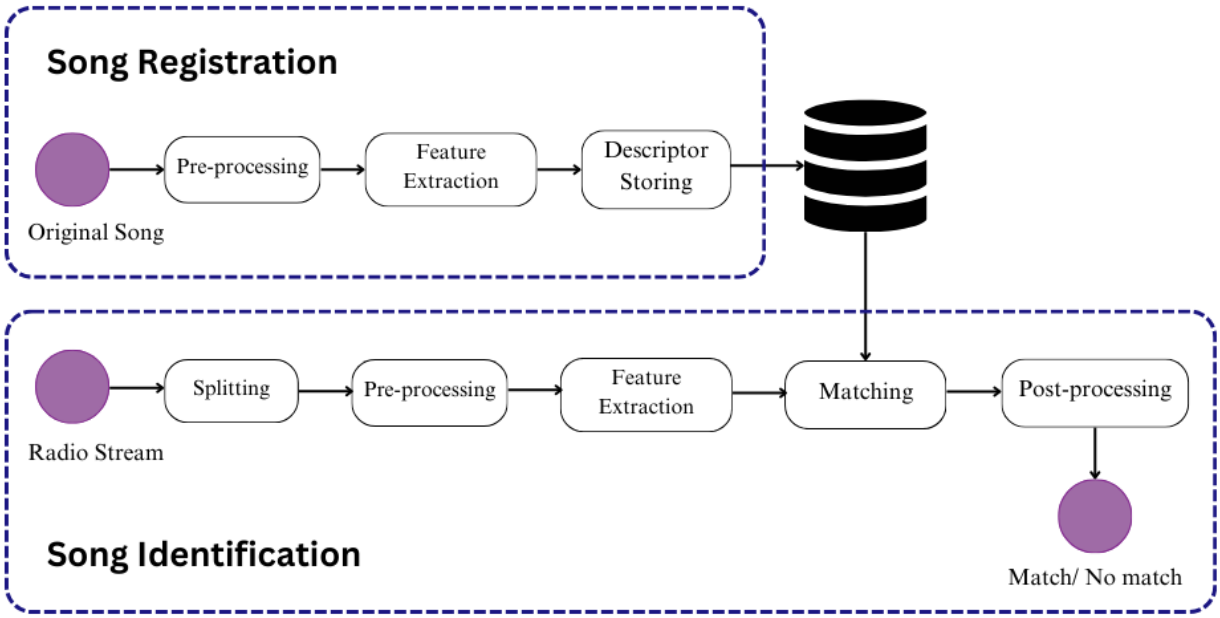

Fig. 1. The process of remastered song identification

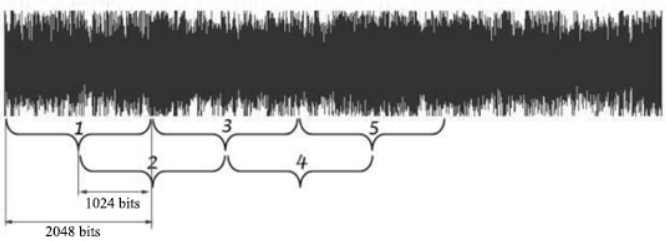

Fig. 2. Key parameters for generating STFT: 2048-bit long window with 50% overlapping

domain representation, generating spectrogram images, and preparing these images for the application of computer vision techniques to extract key features relevant to the audio input.

Audio data is commonly represented in the waveform, depicting sound pressure variations over time. However, this representation is highly susceptible to even minor changes in the audio signal, leading to substantial alterations in the waveform. For example, the addition of external noise can significantly distort the original waveform. Consequently, analyzing the waveform representations of the same song played in different radio stations may not reveal similarity, as various noises and environmental effects in different stations can cause substantial changes in the time domain representations.

To address this challenge, we convert the time domain signal into the frequency domain. The frequency domain representation is directly related to the energy of the audio signal. Notably, a significant impact is required to alter the energy of a frame, indicating that the frequency domain signal representation is highly stable and robust [6]. The transformation from the original time domain waveform to the frequency domain is achieved through the STFT method. This method involves segmenting the audio signal into fixed-sized frames, known as windows, and applying the Fourier Transform (FT) to each window. The sequence of FTs constructs the corresponding STFT representation of the waveform for an audio signal [22].

To mitigate spectral leakage, which occurs when applying the FT directly on the window, the Hann window function is applied on the windows before the transformation. However, this application of window functions can result in some loss of information from the audio signal. To address this, a certain degree of overlap is introduced in the series of windows used to generate the STFT of the audio segment, aiming to minimize information loss [22]. This study utilizes a 2048-bit-long window with 50% overlapping, a common overlapping proportion in the audio domain [5]. Figure 2 illustrates this.

The STFT is commonly visualised using its spectrogram, representing an intensity plot of the STFT magnitude over time [22]. The resulting spectrogram is then transformed into a colour image, as shown in Figure 3. To prevent potential disturbances during the feature extraction step, axis labels and ticks are removed from the image. In this process, a three-channel RGB image can be converted to a grayscale representation, enhancing the performance of the feature extraction step. Despite the loss of colour information, it is not essential for identifying the key features of the spectrogram image.

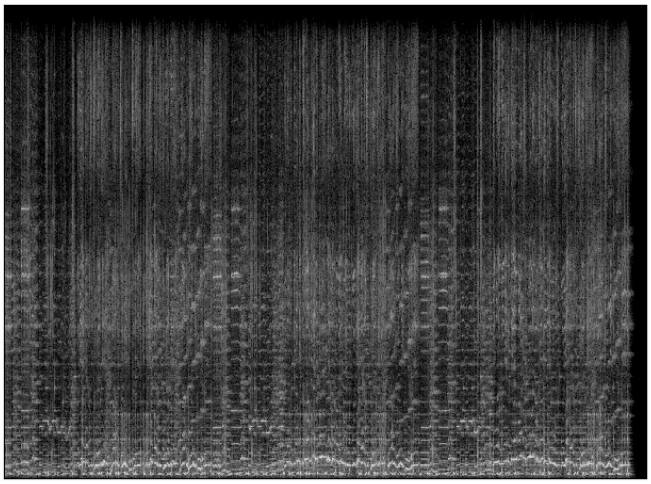

Fig. 3. Generated STFT spectrogram after preprocessing for a 40-second long audio segment.

C. Feature Extraction

The feature extraction step is a pivotal step occurring in both the song registration and identification stages. Its primary objective is to derive representative features from the audio input, ensuring suitability for effective similarity comparisons. Additionally, these features must demonstrate robustness against challenges such as added noise, variations stemming from environmental conditions, and the typical alterations in the time and frequency domains encountered in remastered radio broadcasts.

The impact of noise is alleviated by transforming the audio waveform from the time domain into STFT representation in the frequency domain. Even though STFT spectrogram itself can be considered as an audio descriptor [14], it remains susceptible to alterations in the time and frequency domains, such as time stretching, pitch shifting, and time scaling (these terms are defined in the Introduction section). Since time scaling can be roughly deemed as the combination of time stretching and pitch shifting, in this subsection, we only take time stretching and pitch shifting into consideration.

Figure 4 visually demonstrates the alterations evident in the STFT spectrogram during adjustments to the tempo and pitch. Notably, variations in tempo (time stretching) result in the spectrogram either expanding when the tempo is reduced or compressing when the tempo is increased along the time axis. Similarly, changes in pitch (pitch shifting) lead to the spectrogram shifting either upwards when the pitch is increased or downwards when the pitch is decreased along the frequency axis.

Thus, the transition between the STFT spectrogram of the original song and that of its remastered version can be likened to image scaling. Consequently, applying a scale-invariant feature extraction method to the STFT spectrogram emerges as a robust strategy for preserving essential audio features while withstanding the alterations inherent in remastered music.

Methods from computer vision that entail extracting local image features prove applicable in this context, as the transformations affecting STFT spectrograms primarily influence global features without substantially altering their local characteristics. Among the available local image features, SIFT-based features [23] exhibit remarkable invariance to image rotation and resilience to changes in scale, illumination, and other image deformations. Thus, we choose SIFT descriptors as audio descriptors in this study.

A typical SIFT feature extractor consists of four major stages.

1) Scale-space Extrema Detection: The initial stage of computation involves searching over all scales and image locations, which is efficiently implemented using a Difference of Gaussians (DoG) function. This process includes convolving the image with Gaussian filters at various scales, followed by the computation of differences between sequentially blurred Gaussian images. These differences, known as DoG images at multiple scales, help identify candidate keypoints that exhibit scale and orientation invariance.

Candidate keypoints are selected from local minima/maxima of these DoG images. Figure 5 demonstrates this process, wherein each sample point undergoes comparison with eight neighbours in the current image and nine neighbours in the scale above and below. A sample point qualifies as a keypoint only if it surpasses or falls below all of these neighbouring points.

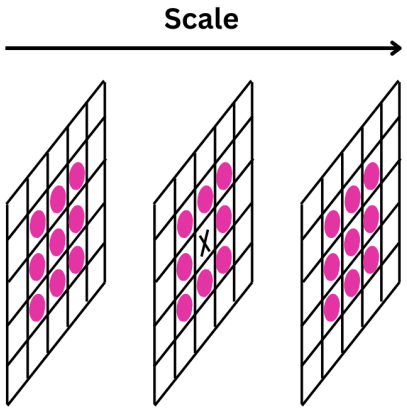

Fig. 5. Maxima and minima in the difference-of-Gaussian images are identified through a comparison of a pixel (indicated with X) with its 26 neighbours in 3×3 regions at the current and adjacent scales (marked with circles).

2) Keypoint Localization: At each candidate location, a detailed model is fitted to determine location and scale, and keypoints are selected based on measures of their stability. However, the initial keypoint detection generates an excessive number of candidates, some of which lack stability. To rectify this, the subsequent phase involves the elimination of keypoints with low contrast. This is achieved by applying the Taylor expansion to each point, reducing sensitivity to noise. Furthermore, poorly positioned keypoints along edges are filtered out by computing principal curvatures using the Hessian matrix. This refinement process enhances the stability of the selected keypoints.

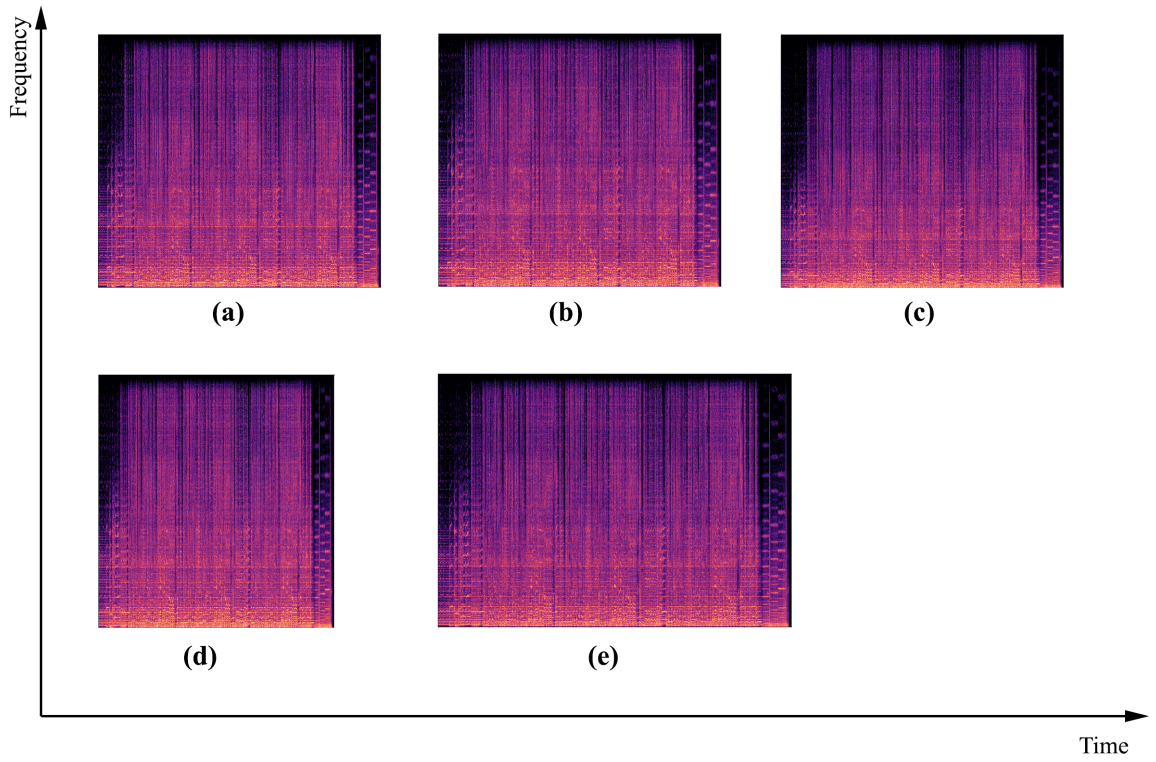

Fig. 4. Spectrogram transformations on audio enhancements. **(a)** is the spectrogram image of an original song. **(b)** 20% pitch increase, **(c)** 20% pitch decrease, **(d)** 20% tempo increase and **(e)** 20% tempo decrease spectrogram images.

3) Orientation Assignment: By consistently assigning an orientation to each keypoint, the keypoint descriptor can be represented relative to this orientation, achieving invariance to image rotation. Each keypoint is assigned one or more orientations based on local image properties, with the keypoint scale determining the Gaussian smoothed image for scale-invariant computations. For each image sample at this scale, gradient magnitude and orientation are precomputed using pixel differences.

An orientation histogram is created from sample points around the keypoint, with 36 bins covering the 360-degree range. Samples in the histogram are weighted by gradient magnitude, and peaks correspond to dominant gradient directions. The highest peak is identified, and any local peak within 80% is used to create a keypoint with that orientation, resulting in multiple keypoints at the same location and scale but with different orientations for locations with similar magnitude peaks.

4) Keypoint Descriptor Generation: Figure 6 depicts the keypoint descriptor computation process. Initially, image gradient magnitudes and orientations are sampled around the keypoint location, with the keypoint scale determining the Gaussian blur level for the image. To ensure orientation invariance, both the descriptor coordinates and gradient orientations are rotated relative to the keypoint orientation, as represented by small arrows at each sample location on the left side of Figure 6. A Gaussian weighting function is applied, assigning weights to the magnitudes of each sample point, as shown by the circular window on the left side of Figure 6.

The keypoint descriptor, as depicted on the right side of Figure 6, accommodates significant shifts in gradient positions through orientation histograms over 4 × 4 sample regions. In the figure, each orientation histogram includes eight directions, and the length of each arrow signifies the magnitude of the corresponding histogram entry. The descriptor is a vector comprising the values of all orientation histogram entries, reflecting the arrow lengths on the right side of Figure 6. While the figure presents a 2 × 2 array of orientation histograms, the actual SIFT implementation utilizes a 4 × 4 array with 8 orientation bins each. Consequently, a descriptor employs a 4 × 4 × 8 = 128-element feature vector for each keypoint.

A collection of 128-dimensional descriptors, each generated for every STFT spectrogram corresponding to an audio input, collaboratively describes that specific audio input (see Figure 7). The extracted SIFT features, as mentioned, exhibit invariance to image stretch and translation. This property enhances their suitability for identifying remastered music, showcasing robustness to tempo alterations and pitch shifting.

D. Descriptor Storing (Registering)

To enable the matching step in the song identification stage, it is imperative to store the SIFT descriptors of the original songs in a database. Typically, a 2-5 minute music clip comprises approximately 2000 keypoints in its STFT spectrogram, with each keypoint corresponding to a SIFT descriptor containing 128 values. Consequently, a 2000×128 matrix is generated for each original song, serving as the registered set of descriptors.

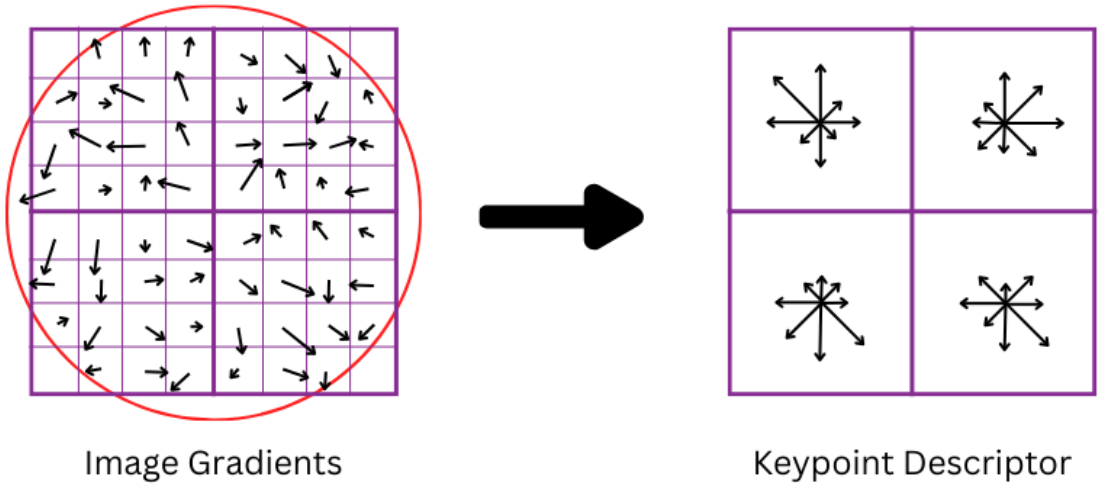

Fig. 6. A keypoint descriptor is created by calculating the gradient magnitude and orientation at each sample point around the keypoint, weighted by a Gaussian window. These values are then aggregated into orientation histograms for 4x4 subregions. The length of each arrow in the histograms represents the sum of gradient magnitudes in that direction within the region. The figure illustrates a 2 × 2 descriptor array computed from an 8 × 8 sample set, while the real SIFT implementation uses 4 × 4 descriptors from a 16 × 16 sample array.

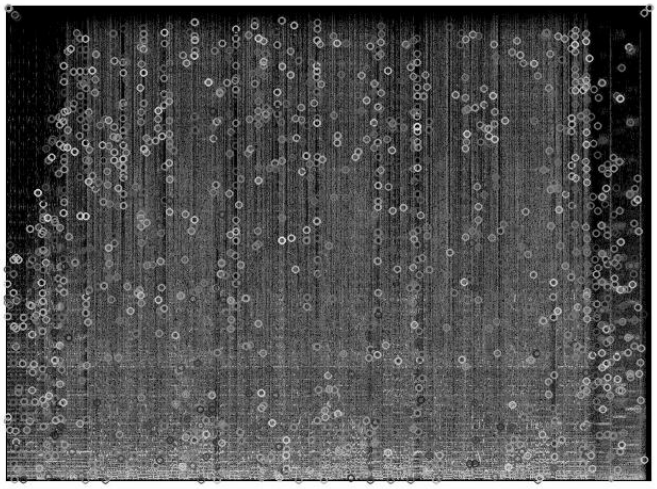

Fig. 7. Visual representation of SIFT local features extracted from the spectrogram image of a 40-second audio stream. Each keypoint, denoted by a circle, is associated with a 128-dimensional descriptor.

Due to the variable lengths of original songs, the descriptor matrix for each song does not have a fixed size. Hence, each original song's descriptor matrix is stored in the database as a Binary Large Object (BLOB) [24], an unstructured data type that converts any data into a binary string for storage. This conversion to a binary string and storage as a BLOB facilitates rapid recreation of the matrix during retrieval.

E. Matching Keypoints

In the song identification stage, the process entails the comparison of the descriptor matrix produced for the STFT spectrogram of the query audio frame with each descriptor matrix associated with an original song registered in the song registration stage, aiming to identify a match. This matching process encompasses comparing all possible pairs of keypoint descriptors, where each pair is composed of one descriptor from each matrix. The count of matching keypoint pairs serves as the basis for determining the overall match between the two descriptor matrices. This count, in turn, dictates whether the query audio frame aligns with the selected original song.

This section delineates the criteria used to determine whether two keypoints are considered a match. The upcoming section (Section III-F) will elaborate on the criteria for considering a query audio frame and a chosen original song as a match, taking into account the number of matched keypoints within those two audio segments.

The optimal candidate match for each keypoint in the query audio frame's spectrogram is determined by locating its nearest neighbour within the keypoints database. The nearest neighbor is identified as the keypoint with the minimum Euclidean distance for the invariant descriptor vector. However, numerous features in a spectrogram image may lack a valid match in the database, either due to background noise or the features of the query audio frame not being registered in the database (in the case of unregistered songs, commercials, and dialogues included in the broadcast stream). Consequently, it is crucial to exclude features that lack a satisfactory match in the database.

We employ the thresholding criteria presented in [23], derived from experiments, to determine the validity of a match for a query keypoint in the database. The criteria involve calculating the distances to the two nearest keypoints from a query keypoint and checking whether the distance to the closest keypoint is less than 0.75 times the distance to the second closest keypoint. If this condition is met, the closest keypoint is considered a match for the query keypoint; otherwise, the query keypoint is discarded and considered as not having a match in the database.

F. Postprocessing

In the preceding phase, the most similar song and the matched keypoint count for a given query audio frame are identified. However, it cannot be guaranteed that the query audio frame contains that song; the matched keypoint count only indicates how much the two spectrograms are similar. Therefore, a threshold keypoint count is needed to determine whether a query audio frame contains the particular best-matching song or not.

We propose that the threshold should be determined for the ratio between the number of matched keypoints and the total number of keypoints in the query audio frame, rather than for the matching keypoint count. This is essential because different query audio frames generate varying numbers of keypoints to match against the database. The keypoint ratio can be calculated using Equation 1.

$$\text{Keypoint ratio} = \frac{\text{Matched keypoint count}}{\text{Keypoints generated for query audio frame}} \tag{1}$$

An experimentally determined threshold (see Section IV for more details regarding this experiment) for the keypoint ratio is utilized to evaluate the validity of matches identified in the previous phase. If the keypoint ratio for a query audio frame exceeds the threshold, it is considered a valid match to the song identified in the matching phase; otherwise, it is deemed an invalid match. This thresholding mechanism helps prevent random matches, particularly in cases where the query audio frame does not contain a previously registered song or includes commercial content or dialogue.

IV. Experiments

In this section, we evaluate the accuracy and performance of the proposed method. Initially, we offer an overview of the song dataset used to construct the descriptor database. Subsequently, we elucidate the process employed to generate query audio streams for various test cases. Following that, we present the experiment conducted to determine the optimal threshold keypoint ratio. The results obtained from the experiment are presented at the end of this section.

A. Song Dataset

A dataset of 2300 Sinhala songs was used in the registration process of the experiment. These songs were obtained from the Outstanding Song Creators Association (OSCA) of Sri Lanka, a governing organization responsible for safeguarding the intellectual property rights of music in Sri Lanka. The songs were contributed by over 200 member artists of the organization.

B. Test Cases

It is necessary to evaluate the performance of the proposed method in terms of its robustness against different audio distortions. In order to evaluate the robustness, 18 test cases that varied in terms of the type, direction, and level of audio distortion were created. The performance of the method was evaluated for three types of distortion: tempo alterations, pitch alterations, and alterations in both tempo and pitch. Both increased and decreased alterations were considered at three different levels of distortion (10%, 20%, and 50%) for each type of distortion, which resulted in the following 18 ($3\times2\times3$) test cases.

1) Tempo Alterations
   - Tempo Increase 10%
   - Tempo Increase 20%
   - Tempo Increase 50%
   - Tempo Decrease 10%
   - Tempo Decrease 20%
   - Tempo Decrease 50%
2) Pitch Alterations
   - Pitch Increase 10%
   - Pitch Increase 20%
   - Pitch Increase 50%
   - Pitch Decrease 10%
   - Pitch Decrease 20%
   - Pitch Decrease 50%
3) Both Tempo & Pitch Alterations
   - Tempo & Pitch Increase 10%
   - Tempo & Pitch Increase 20%
   - Tempo & Pitch Increase 50%
   - Tempo & Pitch Decrease 10%
   - Tempo & Pitch Decrease 20%
   - Tempo & Pitch Decrease 50%

These test cases were considered when creating query audio streams with audio distortions used in the experiment.

C. Query Audio Streams

During the song registration stage (as described in Section III), descriptor matrices were generated and saved for each of the 2300 songs in the dataset. To assess the method's effectiveness in distinguishing between songs present in the database and those that are not, 519 audio streams were generated using songs already registered in the database, while 325 audio streams were generated using songs that had not been previously registered. Altogether, 844 query audio streams were created with a prevalence of 0.61492. The size or duration of the audio streams was intentionally varied to evaluate the appropriateness of the 40-second audio splitting. Finally, all 844 audio streams were altered according to the 18 test cases mentioned above, resulting in 15192 (844×18) query audio streams.

D. Adjusting Threshold

The threshold value for the keypoint ratio determines the criteria for the method to decide whether the best-match song identified by the system for a query audio frame is a valid match or not. As this threshold prevents random matches, finding the optimal value for the threshold is crucial for the model's performance. We conducted an experiment to determine the optimal threshold value using 500 songs, with 250 songs registered in the database and the remaining 250 from outside the database. The selected 500 songs were altered

according to the 18 test cases mentioned above, resulting in 9000 (500×18) songs used as query audio streams input to the experiment.

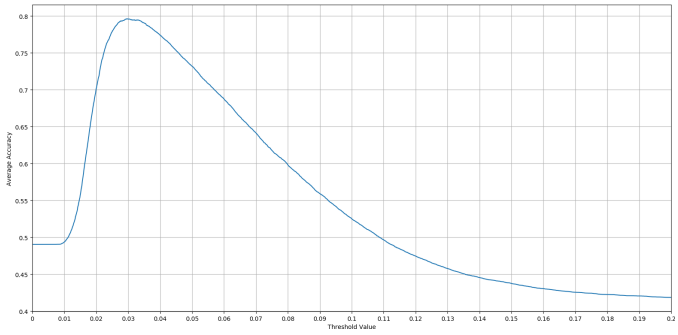

Fig. 8. Variation of the system accuracy with the threshold keypoint ratio

The accuracy was recorded for each threshold value, ranging from 0 to 1 with an increment of 0.0001 in each iteration. The accuracy distribution is depicted in Figure 8. It can be observed that there is a threshold value that maximizes accuracy, which is 0.0298. Therefore, 0.0298 was selected as the optimal threshold value for the experiment.

E. Test Results

The proposed method provides specific criteria to determine whether a query audio clip has a matching registered song or not, thereby functioning as a classifier. The evaluation of a classifier involves analyzing the confusion matrix generated for a given sample. True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) values were calculated for each test case. In the context of our experiment, these elements of the confusion matrix can be mapped as follows.

- TP: The system has successfully identified the song contained in the query audio frame.
- FP: The system has incorrectly mapped the query audio frame to a registered song that is not included in the query audio frame.
- TN: The system has successfully identified that the query audio frame does not contain any of the registered songs.
- FN: The system was not able to find the correct matching song in the registered database even though there is a match to the content of the query audio frame.

The accuracy ($\frac{TP+TN}{P+N}$) and FP rate ($\frac{FP}{FP+TN}$) were calculated for each test case, as shown in tables I to III. Reducing the FP rate is as significant as increasing accuracy because it is crucial for the system to prevent falsely matching songs that were not actually played by the radio station.

Results for the tempo alterations are shown in Table I. The method achieved more than 96% accuracy for all the test cases. It can be observed that test cases with a tempo decrease exhibit higher accuracy than those with a tempo increase, as a tempo increase tends to compress the audio spectrogram in the time axis, leading to the loss of details in the spectrogram.

Table II shows the experiment results for pitch alterations. Results show that performance considerably decreases when pitch alterations reach 50%, and the method performs better for pitch increases as expected. More than 50% alterations tend to push patterns in the spectrogram to the top if there is an increase in pitch and to the bottom if there is a decrease in pitch. Increases in pitch perform better than decreases because pitch decreases compress spectrogram patterns to the bottom, which leads to loss of spectrogram details. In contrast, increased pitch extends the pattern to the top as music remains in the lower part of the frequency band [25].

Test Case	TP	FP	TN	FN	Accuracy	FP Rate
Tempo Increase 10%	512	13	312	7	0.97630	0.04000
Tempo Increase 20%	508	10	315	11	0.97512	0.03077
Tempo Increase 50%	501	9	316	18	0.96801	0.02769
Tempo Decrease 10%	515	11	314	4	0.98223	0.03385
Tempo Decrease 20%	519	10	315	0	0.98815	0.03077
Tempo Decrease 50%	519	9	316	0	0.98934	0.02769

TABLE I
EXPERIMENT RESULTS: TEMPO ALTERATIONS

Test Case	TP	FP	TN	FN	Accuracy	FP Rate
Pitch Increase 10%	519	9	316	0	0.98934	0.02769
Pitch Increase 20%	517	9	316	2	0.98697	0.02769
Pitch Increase 50%	0	4	321	519	0.38033	0.01231
Pitch Decrease 10%	516	16	309	3	0.97749	0.04923
Pitch Decrease 20%	503	25	300	16	0.95142	0.07692
Pitch Decrease 50%	23	160	165	496	0.22275	0.49231

TABLE II
EXPERIMENT RESULTS: PITCH ALTERATIONS

Combined tempo and pitch alterations decrease performance compared to individual alterations, as shown in Table III. However, the method achieves 70% accuracy for combined alterations up to 20%. Expansions and compressions in both axes of a spectrogram tend to alter patterns in the spectrogram, making it hard to compare and find similarities with the original spectrogram.

Test Case	TP	FP	TN	FN	Accuracy	FP Rate
Tempo & Pitch Increase 10%	413	11	314	106	0.86137	0.03385
Tempo & Pitch Increase 20%	287	17	308	232	0.70498	0.05231
Tempo & Pitch Increase 50%	0	7	318	519	0.37678	0.02154
Tempo & Pitch Decrease 10%	432	28	297	87	0.86374	0.08615
Tempo & Pitch Decrease 20%	346	34	291	173	0.75474	0.10462
Tempo & Pitch Decrease 50%	6	154	171	513	0.20972	0.47385

TABLE III
EXPERIMENT RESULTS: BOTH TEMPO & PITCH ALTERATIONS

Overall, the experiment results show that the method is more robust towards tempo alterations than pitch alterations. However, up to 20% of alterations model performs with 95% accuracy for individual alterations. 20% alteration is a considerable alteration where the human ear can quickly

identify the difference, which makes the method ideal for identifying songs that radio stations slightly alter, making it unable to be detected by radio broadcast monitoring systems.

V. Discussions

In general, the current automatic monitoring systems designed to identify remastered music in radio streams exhibit limited effectiveness when radio stations modify the signals in either the time or frequency domain. The approach proposed by Senevirathna *et al.* [5] assessed their model through various scenarios, including mixing non-song elements (such as commercials and dialogues) and introducing non-song elements in the middle of song broadcasts, as well as adding noise to the song broadcasts. However, this method did not evaluate their model's performance against alterations in the time and frequency domains.

In practical scenarios, radio stations frequently adjust the speed of songs to accommodate their schedules, resulting in changes in both the time and frequency domains. The results demonstrate that the proposed method can effectively handle spectrogram transformations arising from these practically occurring tempo and pitch alterations. This is attributed to the utilization of SIFT descriptors, known for their high robustness to common affine transformations in images.

Analysis of FPs is valuable for comprehending the behaviour of the proposed method. It is observed that more FPs occur with songs that exhibit rapid changes compared to an average Sinhala song. These changes include alterations in instrumentalization, dynamics, and melody. The potential reason for this observation is that these types of songs produce complex STFT spectrograms with many keypoints having a range of different feature descriptors. Therefore, due to the generality it creates, the segments in the radio streams, especially the audio segments containing songs that do not have their matching original song in the database and non-song elements, tend to match to these songs and produce FPs.

Furthermore, an increase in model accuracy has been observed as the tempo decreases. The most accurate results in tempo alterations are produced when the tempo is decreased to its maximum level, which is 50%. Conversely, the accuracy is reported to be lowest when the tempo is increased to its maximum level, also set at 50%. This observation can be attributed to the fact that as the tempo decreases, audio features are interpolated along the time axis, preserving all information. Consequently, the corresponding STFT spectrogram generated becomes more stretched, making the features clearer, and the keypoints more visible, leading to higher accuracies. On the other hand, when the tempo is increased, features are compressed along the time axis, resulting in a loss of some original audio data. This compression affects the corresponding spectrogram, causing it to lose some unique features in the audio that do not last for an extended duration, such as a triangle hit.

Despite its capabilities, the proposed method has certain limitations. The keypoint descriptor matching involves a brute-force search across the entire keypoint descriptor database of the original songs, impacting the system's efficiency. As the database expands with the registration of newly released songs, the performance of the system is expected to further degrade.

VI. Conclusions and Future Work

In this paper, we have introduced a novel music identification algorithm designed to identify remastered music in radio broadcasts, contributing to the protection of artists' copyright ownership. The proposed algorithm involves segmenting the continuous radio broadcast into fixed-length chunks. Each chunk undergoes an identification process that results in either a match from the database or no match. No match is expected to occur in cases where the audio segment contains non-song elements or a song not registered in the database.

The matching algorithm employed in this study utilizes the STFT spectrogram of the audio snippet. A SIFT descriptor is generated for this spectrogram and compared with pre-registered descriptors in the database. A threshold is applied to discern whether the matched musical content is present in the music database. The utilization of SIFT descriptors has demonstrated notable robustness against tempo and pitch alterations, as well as other changes such as noise that can commonly occur due to environmental factors affecting radio broadcasts.

The algorithm demonstrates impressive accuracy, surpassing 97% in identifying tempo alterations of up to 20% and achieving over 95% accuracy in identifying pitch alterations within the same range.

Future work involves the integration of a search-efficient and scalable storage mechanism to enhance the overall performance of the proposed method. Further studies could focus on improving the speed of both descriptor matching and feature extraction. While the proposed method demonstrated effectiveness when the generated spectrogram is stretched due to alterations in the time and frequency domain, its performance degrades when the spectrogram is compressed. Exploring novel methods to address these phenomena can enhance the system's robustness and accuracy.

Moreover, addressing complex changes introduced during remastering, such as the addition of a drumbeat that alters the audio stream's structure, poses a challenging task for music identification. These aspects represent opportunities for future research inspired by our work.

References

[1] Parliament of the democratic socialist republic of Sri Lanka, "Intellectual Property Act, No.36 of 2003."
[2] "Radio royalties: How do radio stations pay artists?" https://soundcharts.com/blog/radio-royalties, accessed: 2023-11-30.
[3] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015.
[4] X. Zhang, B. Zhu, L. Li, W. Li, X. Li, W. Wang, P. Lu, and W. Zhang, "Sift-based local spectrogram image descriptor: a novel feature for robust music identification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 6, 2015.
[5] E. Senevirathna and L. Jayaratne, "Radio broadcast monitoring to ensure copyright ownership," *ICTer*, vol. 11, no. 1, 2018.
[6] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 41, no. 3 SPEC. ISS., pp. 271–284, 2005.
[7] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system." in *Ismir*, vol. 2002, 2002, pp. 107–115.

[8] A. Wang *et al.*, "An industrial strength audio search algorithm." in *Ismir*, vol. 2003. Washington, DC, 2003, pp. 7–13.
[9] B. A. y Arcas, B. Gfeller, R. Guo, K. Kilgour, S. Kumar, J. Lyon, J. Odell, M. Ritter, D. Roblek, M. Sharifi, and M. Velimirović, "Now Playing: Continuous low-power music recognition," *arXiv:1711.10958 [cs, eess]*, Nov. 2017.
[10] Z. Yu, X. Du, B. Zhu, and Z. Ma, "Contrastive unsupervised learning for audio fingerprinting," *arXiv preprint arXiv:2010.13540*, 2020.
[11] S. Chang, D. Lee, J. Park, H. Lim, K. Lee, K. Ko, and Y. Han, "Neural audio fingerprint for high-specific audio retrieval based on contrastive learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3025–3029.
[12] X. Wu and H. Wang, "Asymmetric contrastive learning for audio fingerprinting," *IEEE Signal Processing Letters*, vol. 29, pp. 1873–1877, 2022.
[13] A. Singh, K. Demuynck, and V. Arora, "Attention-based audio embeddings for query-by-example," 2022.
[14] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, pp. 597–604, 2005.
[15] S. Baluja and M. Covell, "Waveprint: Efficient wavelet-based audio fingerprinting," *Pattern Recognition*, vol. 41, no. 11, pp. 3467–3480, 2008.
[16] B. Zhu, W. Li, Z. Wang, and X. Xue, "A novel audio fingerprinting method robust to time scale modification and pitch shifting," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 987–990.
[17] M. Zanoni, S. Lusardi, P. Bestagini, A. Canclini, A. Sarti, and S. Tubaro, "Efficient music identification approach based on local spectrogram image descriptors," *Journal of The Audio Engineering Society*, 2017.
[18] D. Williams, A. Pooransingh, and J. Saitoo, "Efficient music identification using orb descriptors of the spectrogram image," *EURASIP J. Audio Speech Music Process.*, vol. 2017, no. 1, dec 2017.
[19] J. Six and M. Leman, "Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification," in *ISMIR*, 2014.
[20] A. Báez-Suárez, N. Shah, J. A. Nolazco-Flores, S.-H. S. Huang, O. Gnawali, and W. Shi, "SAMAF: Sequence-to-sequence autoencoder model for audio fingerprinting," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 16, no. 2, May 2020.
[21] M. T. Htun and T. T. Oo, "Broadcast monitoring system using mfcc-based audio fingerprinting," in *2023 IEEE Conference on Computer Applications (ICCA)*, 2023, pp. 243–247.
[22] N. Kehtarnavaz, "Frequency Domain Processing," *Digital Signal Processing System Design*, vol. 1, pp. 175–196, 2008.
[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[24] "Blob data type," https://docs.oracle.com/javadb/10.8.3.0/ref/rrefblob.html, accessed: 2023-11-30.
[25] A. I. Al-Shoshan, "Speech and music classification and separation: A review," *Journal of King Saud University - Engineering Sciences*, vol. 19, no. 1, pp. 95–132, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S101836391830850X