Research Article GU J Sci 36(2): 624-643 (2023) DOI: 10.35378/gujs.1000594

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Effect of Spectrogram Parameters and Noise Types on The Performance of Spectro-temporal Peaks Based Audio Search Method

Murat KOSEOGLU[1, *], Hakan UYANIK[2]

[1]*Inonu University, Faculty of Engineering, 44280 Malatya, Turkey*
[2]*Munzur University, Faculty of Engineering, 62000, Tunceli, Turkey*

Highlights

- This paper focuses on the effects of some spectrogram parameters on audio recognition performance.
- Feature extraction algorithm was tested for different noise levels and types.
- Optimal spectrogram parameters for the studied method were obtained by considering the noise effect.

Article Info

Received: 25 Sep 2021
Accepted: 27 May 2022

Keywords

Audio recognition
Reliability
Noise effect
Spectrogram parameters

Abstract

Audio search algorithms are used to detect the queried file in large databases, especially in multimedia applications. These algorithms are expected to perform the detection in a reliable and robust way within the shortest time. In this study, based on spectral peaks method, an audio fingerprint algorithm with a few minor modifications was developed to detect the matching audio file in target database. This method has two stages as the audio fingerprint extraction and matching. In the first stage, fingerprint features are extracted from spectral peaks on the spectrograms of audio files by hash functions. This state-of-art technique reduces the processing load and time considerably compared to traditional methods. In the second stage, fingerprint data of the queried file are compared with the data created in the first stage in the database. The algorithm was demonstrated, and the effect of spectrogram parameters (window size, overlap, number of FFT) was investigated by considering reliability and robustness under different noise sources. Also, it was aimed to contribute to new audio retrieval studies based on spectral peaks method. It was observed that the variation in the spectrogram parameters significantly affected the number of matchings, reliability and robustness. Under high noise conditions, the optimal spectrogram parameters were determined as 512 (window size), 50% (overlap), 512 (number of FFT). It was seen in general that the algorithm successfully detected the queried file in the database even in high noise conditions for these parameters. No significant effect of music genre was observed.

1. INTRODUCTION

The use of audio files in our daily lives has increased radically over the past few decades due to the great advances in the storage techniques, their accessibility, distribution methods and application fields of these files. Depending on these progresses, one can easily reach large collections of audio files from anywhere at any time in a few seconds. The development of different systems and new methods that will enable users to access required data file easily and quickly from large data stacks has a great importance in the manner of supplying the demands in different application fields [1].

Many audio file search engines perform a search process based on the meta data of the file, mostly text information [2]. Audio fingerprint technology is a technique that is frequently used in the field of Automatic Content Recognition (ACR) that creates the unique digital identity of an audio file. The basic features which an audio fingerprint must preserve, have been extensively analyzed in literature [3]. When we look at real-life applications, the important features that these fingerprints should possess are briefly stated as follows [1]: First of all, these fingerprints must have a highly specific characteristic structure. Thus, with the help of a very short piece of audio file, it can be easily and quickly detected from the database among millions

*Corresponding author, e-mail: murat.koseoglu@inonu.edu.tr

of audio files. However, in real-life applications, audio signals are exposed to various distortions. In particular, the signal is likely to be affected by noise, lossy audio compression, pitch shifting, time scaling, time equalization, or dynamic compression. Fingerprints must be fairly robust against such distortions for reliable identification. On the other hand, it must full-fill the requirements resulting from the changing scale by the way of calculation of minimum storage needs and transmission delays in a database growing in size every day [3]. Audio search algorithms, which are created by considering these basic features, are applied with high efficiency for large scale data process applications.

It can be seen from the literature that two main categorial methods are used in the analysis of audio files. The first method employs short sequences of frame-based feature vectors such as Mel-Frequency Cepstral Coefficients (MFCC) [4], Bark-scale spectrograms [5,6] or a set of low-level descriptors [7]. The second method employs a sparse set of characteristic points such as spectral peaks [8,9] or characteristic wavelet coefficients [10].

There are various algorithms developed for constructing audio fingerprint. Among these algorithms, the most popular two algorithms, which were applied in real life applications and earned great success in the commercial field, are Philips audio fingerprint algorithm [5] and Shazam algorithm [9]. Philips algorithm introduced by Haitsma and Kalker was created on principle of comparing neighboring energy bands. In this method, the audio fingerprints are created by using the energy differences between the contiguous bands. The effect of distortion and noise on the performance of the method is relatively low. The required time for data retrieval is quite short. The linear changes in the speed of the signal may affect the algorithm negatively [11]. Shazam algorithm is first proposed by Wang. It works by handling the relationship between the sequential spectral peaks. It utilizes the obtained frequency and time information. Its performance is quite satisfactory even if the process is realized in a noisy environment [9,11].

In some studies [12–16], the wavelet transform has been applied in the audio fingerprint techniques and successful results have been obtained. Although this method showed a good performance in terms of operation time, it is very sensitive to the distortions in the signal. To reduce the disturbing effects of noise, Park et al. applied several pre-filtering processes to the Philips method and obtained satisfactory results in [17]. In another study [18], modulated complex lapped transform (MCLT), which is a spectro-temporal peak-based audio fingerprinting method, was used for data extraction, and significant improvements have been obtained in the detection of audio file searched. Also, the computational load was decreased in this method.

In 2012, Anguera and colleagues have proposed a new fingerprinting technique called Masked Audio Spectral Keypoints (MASK) by combining the two methods mentioned above [19]. This algorithm creates dual fingerprints by creating regions around selected specific spectral points and comparing the energy of the spectral band in these regions. Although it gives very good results in sound file detection, its inefficiency in storage space and long processing times are the biggest disadvantages of this method [11].

Audio fingerprint technique has found its place among many important studies in parallel with the developments in the field of machine learning in recent years. Especially in applications made with ANN (Artificial Neural Network), very high success rates are achieved. Studies numbered [20–23] are some important studies carried out in this field.

In this study, the effects of several spectrogram parameters on the reliability and robustness of the audio fingerprint method based on the spectral peaks method are investigated for different noise levels and types. More clearly, the effects of the variation of window length, overlap ratio and applied FFT number, on the success rate are evaluated in the aspect of the number of matchings (NM), reliability and robustness at different noise levels for the noise types such as additive white Gaussian noise (AWGN), restaurant ambiance and traffic ambiance. Also, the effects of some different AWGN levels and four different windowing functions have been analyzed before with a simple and crude version of the program [24, 25].

In similar studies on audio fingerprinting, databases have been created in various ways. For example, in study [26], the used database consists of 500 pop music at home and abroad. All songs in this database have

a sampling rate of 8 kHz. Experiments were carried out by adding 5 dB, 0 dB and 3 dB white noise to the songs in the database. In another study [27], speech signals were processed with the audio fingerprint method. The database used in this study was created from the speech signal library named THCHS-30 in the source [28]. The database created consists of 20-second speech signals in single-channel WAV format and 16 kHz sampling rate. Similarly, in this study, the database was reconstructed with noise at various ratios (30 dB, 20 dB, 10 dB, 5 dB, 0dB Gaussian noise addition).

2. AUDIO FINGERPRINT METHODOLOGY AND IMPLEMENTATION

Audio fingerprint method consists of two successive stages as the audio fingerprint extraction and audio fingerprint matching. The audio fingerprint extraction stage consists of three steps as (i) Preprocessing and Creating Spectrogram, (ii) Peak Selection, (iii) Generating Fingerprint Data, respectively. Then, the audio fingerprint matching stage is carried out as the fourth step. The comprehensive explanations are given in this section.

The audio files in the database used in the study are MP3 files with 48 kHz sampling value in stereo format. Similar to the above studies, some preprocessing has been performed on the database, and new databases have been derived with noise additions at various rates (-6 dB, -3 dB, 0 dB, 3 dB, 6 dB, 9 dB, 12 dB) and different types. The database included 5 genres (each with ten songs) of music files such as R&B, alternative rock, blues, rock and dance genres in order to observe whether the method depends on the genre of audio files used. All audio files have been randomly selected from billboard magazine's list of the "The 500 Greatest Songs of All Time" [29], considering their genre. The genre information for the audio files used in the test procedure is given in Table 1.

In the datasets used in previous works, in the stage of the extraction of the fingerprint of each audio file, constant spectrogram parameters (window length, overlap, FFT number) are used in general while these files are recorded in the dataset. In the current work, in the stage of the fingerprint extraction of each audio file, the spectrogram parameters are changed before the record of each audio file version. Therefore, each audio file may have different versions in the database depending on its spectrogram parameters. In the study, 50 audio files are used, but the database includes different versions of these audio files resulting from the records of different spectrogram parameters. The total audio files collected in the database is about 400.

Table 1. *Genre information for tested audio files*

Music Genre	**Song ID in Database**
R&B	4 and 9
Alternative Rock	15 and 18
Blues	21 and 26
Rock	33 and 37
Dance	40 and 42

In the mentioned studies, the details of the utilized algorithms and methods had not been explained explicitly, step by step. Also, we have noticed that the studies analyzing the effect of spectrogram parameters on the reliability and robustness under different noise conditions available in the literature are limited. Therefore, in this study, an audio fingerprint method, which was proposed by Wang [9], was developed with a few minor modifications, and the reliability and robustness of the method was analyzed in terms of different spectrogram parameters and the effect of different levels of three types of noise. The audio files used in this study were selected from a data set including five different music genres. Schematic representation of the method conducted here is shown in Figure 1.

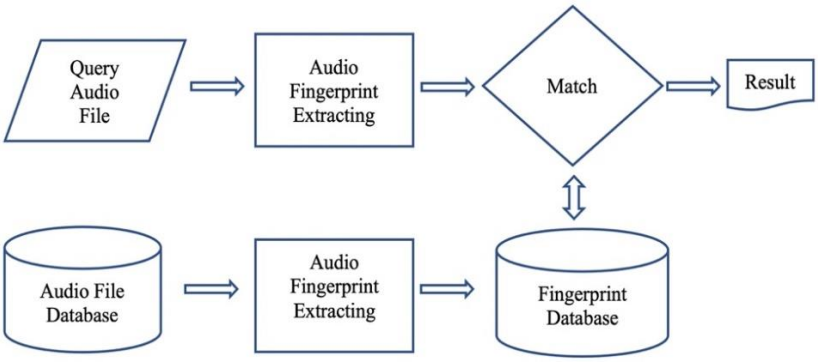


Figure 1. *Diagram of the audio fingerprint technique used in the study*

From the literature we noticed that the methodology used in the algorithms defining fingerprint had been presented in a vague framework, the details of algorithm have been left to the reader to understand. Considering the methodology proposed in Wang's study [9], the systematic steps given in Figure 2 were followed in building the algorithm used here. The algorithm uses spectro-temporal peaks of the signal for achieving a successful detection of audio fingerprint.

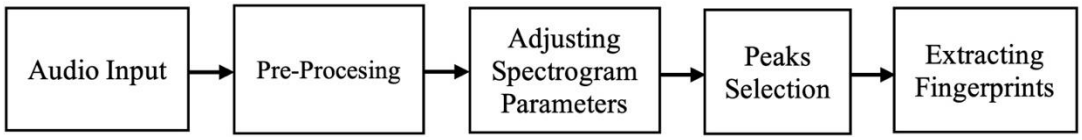


Figure 2. *Block diagram for audio fingerprint extracting process*

2.1. Step I: Preprocessing and Creating Spectrogram

Time-frequency signal processing is one of the best- and well-known method used in audio signal processing. Time frequency methods present “when” and “how” the spectral content of a time-varying signal contribute to the audio signal, which is not possible by LTI processes such as Fourier Transform. Spectrogram is one of the basic methods used in forming time-frequency structure of the signal. Spectrogram, in this sense, is a powerful analysis tool in observing the spectro-temporal characteristics and/or behavior of the non-stationary signals, such as speech [30], biomedical [31], music [32], seismic signals [33].

Before the spectrogram of the audio file is created, some pre-processing should be applied. First of all, it is ensured a single channel structure for the audio file to be handled. Because the input signal x[n] in the equation (1) is a one-dimensional array. A stereo sound signal is converted to mono form by obtaining the arithmetic average of its two channels. After this process, since the main information is generally in low frequency regions in multimedia audio files as seen in Figure 3, the down-sampling process is applied to the audio signal to reduce the workload by filtering the high frequency components in the audio signal. This low frequency region is the case for most multimedia audio files. Because the sound sources that create such sound files are more intense in this band range. Percussion and bass instruments are examples of these sources [34]. At this point, the audio file is recreated with the new sampling frequency of 8 kHz. After this step, a proper combination of spectrogram parameters is selected.

The frequency information of a signal can be obtained with Fourier transform [35], but the time information of the signal is lost [36–38]. In this study, the Short Time Fourier Transform (STFT) was used to obtain spectrogram of the signal [38]. STFT, in discrete form, is given by (1) [39]

$$X_m(k) = \sum_{n=0}^{N-1} x[n+mR]\ \omega[n]\, e^{-jk\omega_0 n} \quad (1)$$

where $X_m(k)$ is STFT value of $x[n]$ input signal, $\omega[n]$ is the window function gliding over the signal, and Hamming window function is used in this study since it preserves the characteristic of signal well. $x[n+mR]$ is the shifted input signal with $(n+mR)$ index, N is the length of the DFT, and m is the window shift control parameter. The spectrogram of the signal is mapped as the power distribution of STFT [40] and given by

$$S_{TF}(t,f) = X(t,f)^2\ . \quad (2)$$

In Equation (2), $X(t,f)$ designates the Fourier transform of the weighted signal by an analysis time window $\omega[t]$ that is moved in time:

$$X(t,f) = \int x(u)w(u-t)\exp(-j2\pi fu)\, du\ . \quad (3)$$

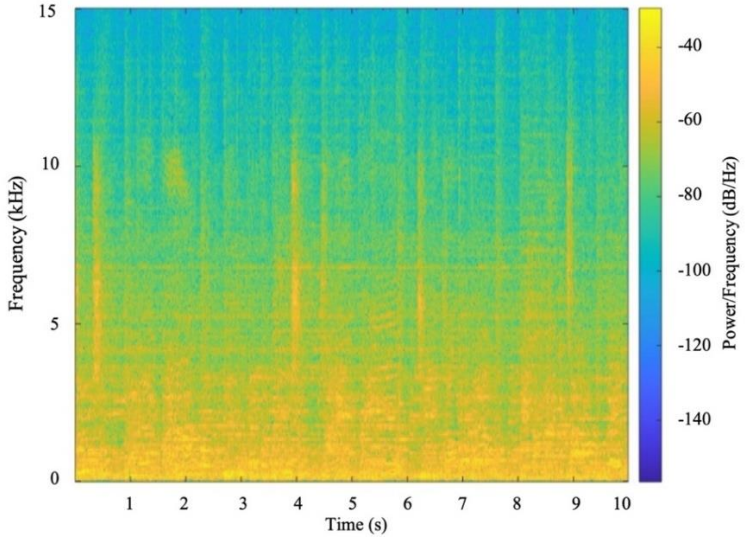


Figure 3. *The spectrogram sample of a 10 s audio file (Song ID=40)*

In this process, the issue is to determine the appropriate window function for a good time-frequency resolution [41]. Due to the well-known natural uncertainty principle, while a long-term window results in a good spectral but poor temporal resolution, a short-term window comes up with a good time but poor spectral resolution [42, 43]. In both cases the spectrogram becomes very sensitive to noise. This situation is defined as the "uncertainty principle" in the literature [44]. This fact is visualized in Figure 4 [45].

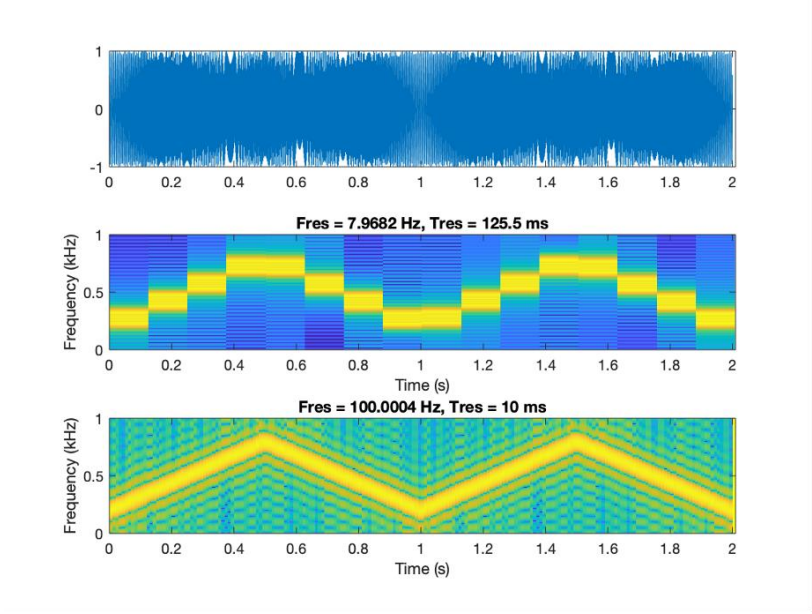


Figure 4. *Time-frequency resolution relationship*

In general, it was experienced that for speech recognition a window length of 20-40 ms offers a good time-frequency resolution [44], while for musical signals a 64 ms window length offers a good time-frequency resolution [32].

2.2. Step II: Peak Selection

After achieving spectrogram of the audio signal, the spectro-temporal peaks which exceeds a certain threshold level and has a resistance to various disturbances [9] are selected on the spectrogram. The distribution of these spectro-temporal peak points is termed as the base of fingerprint of the audio signal, and hence from this data "constellation map" is constructed as explained in Figure 5.

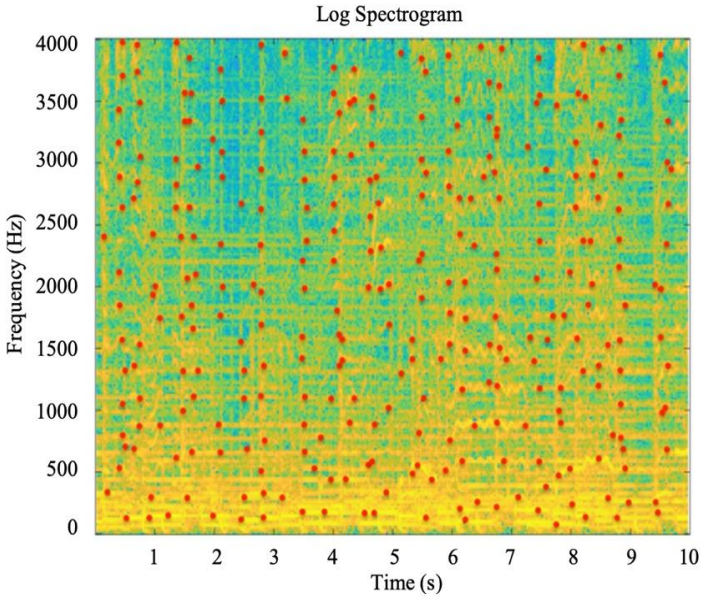


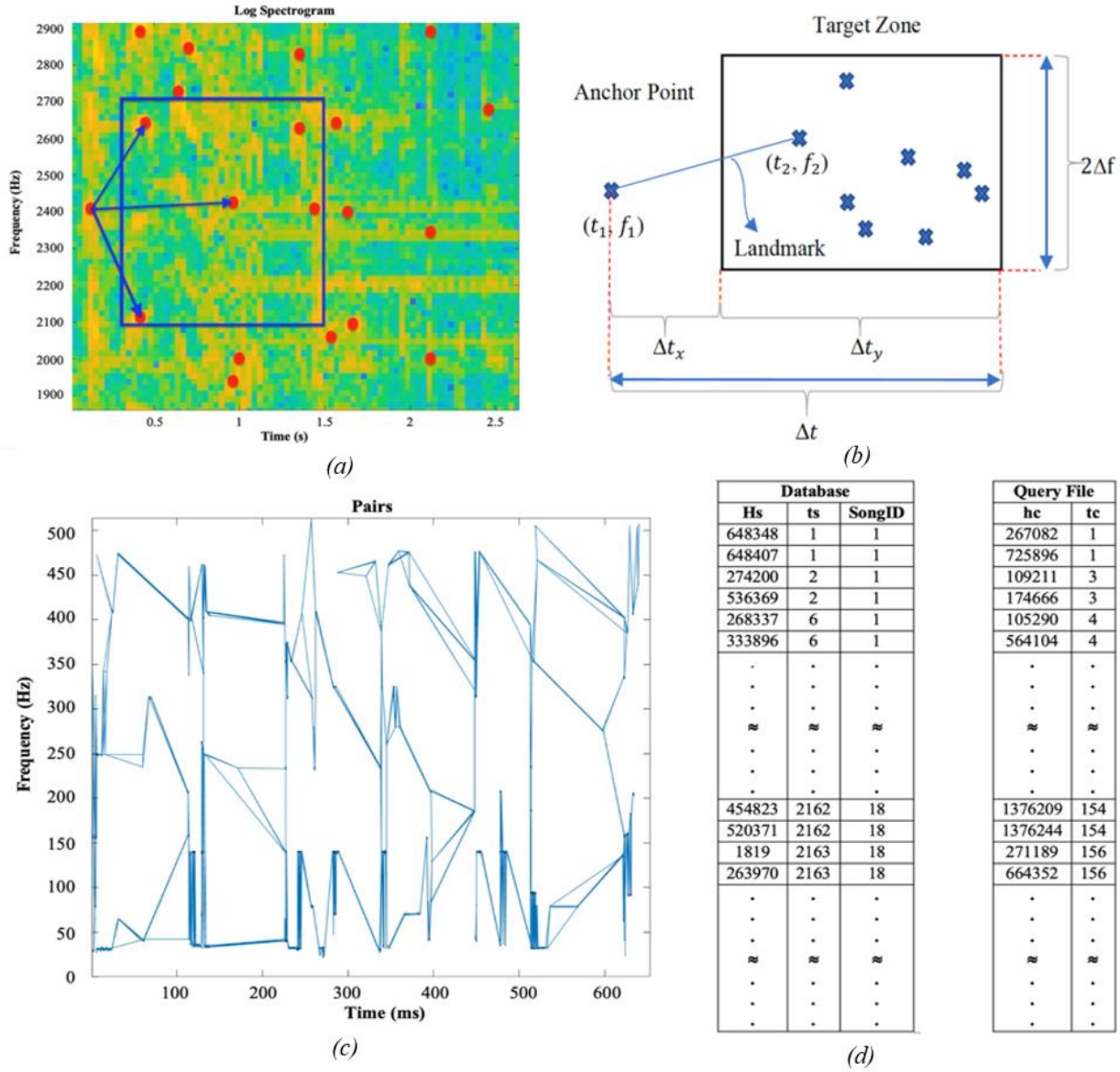
Figure 5. *Display of 30 maximum peak points (red dots) selected per second on the spectrogram created for a 10-second portion of an audio file (SongID=18) at noiseless condition*

2.3. Step III: Generating Fingerprint Data

In this stage, fingerprint of the audio is calculated over signal's constellation map. For this, at first, an anchor point and a target region are determined on the constellation map obtained in the previous step. Then, the fingerprint of audio signal is calculated in association (usually Euclidean distance) with the anchor point to each of the points in the defined target region. The obtained results are tabulated as the fingerprint of the audio signal. Then, with the help of these tables, distinctive features are derived. This technique is termed as hashing, which is frequently used in the field of cryptology [46].

The Hs and hc parameters are the hash value of the song in the song database and the hash value of the queried song, respectively. The ts and tc values are the time value of the anchor point of the file in the database and the time value of the anchor point of the queried file, respectively.

Hashing, by definition, is an irreversible process in which a function or an algorithm is used to map object data (e.g., image data) onto some representative integer value. Hashing is usually used to narrow down our search when looking for the item in the map. The most popular and common example of using hash functions is using for the security of the sharing files over a network. During this process, the hash value of the file to be shared is calculated by the sender. If there is no problem during the transmission, the hash value to be produced on the receiver side overlaps with this value. A problem that may arise at this point is that different files may generate the same hash. In the literature this state is called "collision". To overcome this situation, new techniques are being developed [47].



Database		
Hs	ts	SongID
648348	1	1
648407	1	1
274200	2	1
536369	2	1
268337	6	1
333896	6	1
454823	2162	18
520371	2162	18
1819	2163	18
263970	2163	18

Query File	
hc	tc
267082	1
725896	1
109211	3
174666	3
105290	4
564104	4
1376209	154
1376244	154
271189	156
664352	156

Figure 6. *(a) Target zone scheme and landmark structure (b) Connections of points (landmarks) (c) Tables of fingerprint information generated (d) The table on the left belongs to the database and the table on the right belongs to the audio file to be queried. Querying is performed between these two tables* [48]

In this study, frequency and time values of both peaks are transformed into hash values using a hash function given in [49]:

$$h(f_1, f_2, t_2 - t_1) = (t_2 - t_1) \cdot 2^{16} + f_1 \cdot 2^8 + f_2 \,. \tag{4}$$

In Equation (4), f_1, f_2, t_1 and t_2 are the frequency and time values of the anchor peak point and the querry peak point, respectively. Hashing process is performed for all possible pairs of frequency and time that could be set between query (anchor) and those in the identified region (Figures 6(a), (b) and (c)). The obtained hash values are recorded into a table, as in Figure 6(d) (on the left), while the hash values of the audio files in database are obtained and recorded in a similar table as in Figure 6(d) (on the right). Then searching operations are performed between these two tables.

2.4. Step IV: Matching Fingerprints

In conventional comparison methods, matching process is generally realized as shown in Figure 7 by the way of unit length shift of the audio files constellation points in the form of two-dimensional matrices until

whole constellation points in the database are completed. Then the correct matching is determined according to the maximum number of matching points.

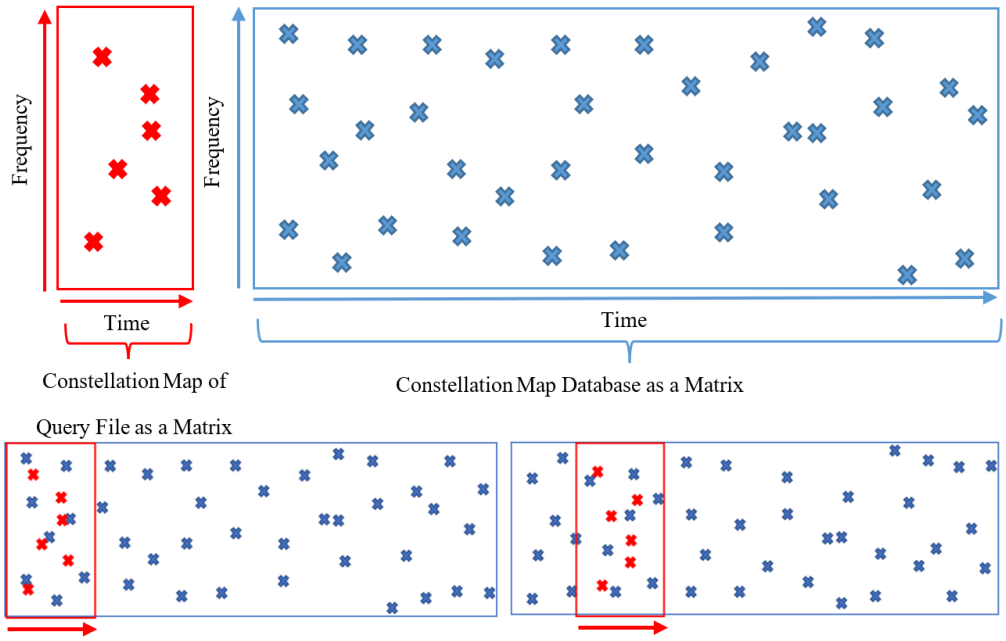


Figure 7. *Conventional comparison method* [1]

However, such methods are not preferred, because these processes require a lot of processing power and time. As suggested in literature [9], querying with hash values to be produced from these two-dimensional matrices has created a radical difference in terms of both time and transaction amounts. With this technique, a strong solution has been created by increasing the entropy of the available data. The data tables (Figure 6(c)) produced with the same procedure are compared quite simply and quickly. At this point, in the first step, equality is sought by comparing the hash values in the table (on the right) of the queried audio file with the hash values in the database (on the left) starting from the top. If the queried audio file is in the database, it is expected that there will be an equality between the hash values in the audio fingerprint data created for this file (hc) and the hash values (Hs) in the audio fingerprint database. Therefore, the algorithm will initially identify rows with the same hash value between these two databases (Figure 6(d)). In the first equation found on the hash value rows, both the time value corresponding to the hash value in the database and the time values of the audio file being queried are recorded. With a second equation that comes after this process, the same process is repeated, and the time values are recorded. After this determination, a connection is created between the time values (ts and tc) between the two files. For this, a variable assignment called “to” (time offset) is made. The main purpose here is to catch the rhythm between the time values of database and the time values of the query files. Because these values are independent of time. It can be from any moment of the audio file. Then, the difference between the time values in the first equation and the time values obtained in the second equation is checked. If there is an equality between these values, the query value (score) of this audio file processed in the database is increased by one unit. In this way, these processes are applied to the entire database in a chain, and the highest scored audio file is determined as the correct file.

An example of this process is given in the Figure 8. In this example, a query is performed from a database obtained from three different audio files. The blue dots on the graphs represent the values of the audio fingerprint database, and the red dots represent the audio fingerprints of the queried audio file. If the audio fingerprint table in the figure is examined, it will be seen that the audio file with the least number of hash value equalities among the three sound files is SongID 3. Hash equality is provided for only one value for this song. Files 1 and 2 have 2 matching hashes. However, a match cannot be achieved in the time offset values of these values. It is observed that these values are equal for the SongID 1. Therefore, in this query process, the highest match value was realized for SongID 1.

Database		
Hs	ts	SongID
2	2	1
4	3	1
2	2	2
4	4	2
6	5	2
2	4	3
4	3	3

Query File	
hc	tc
2	1
4	2

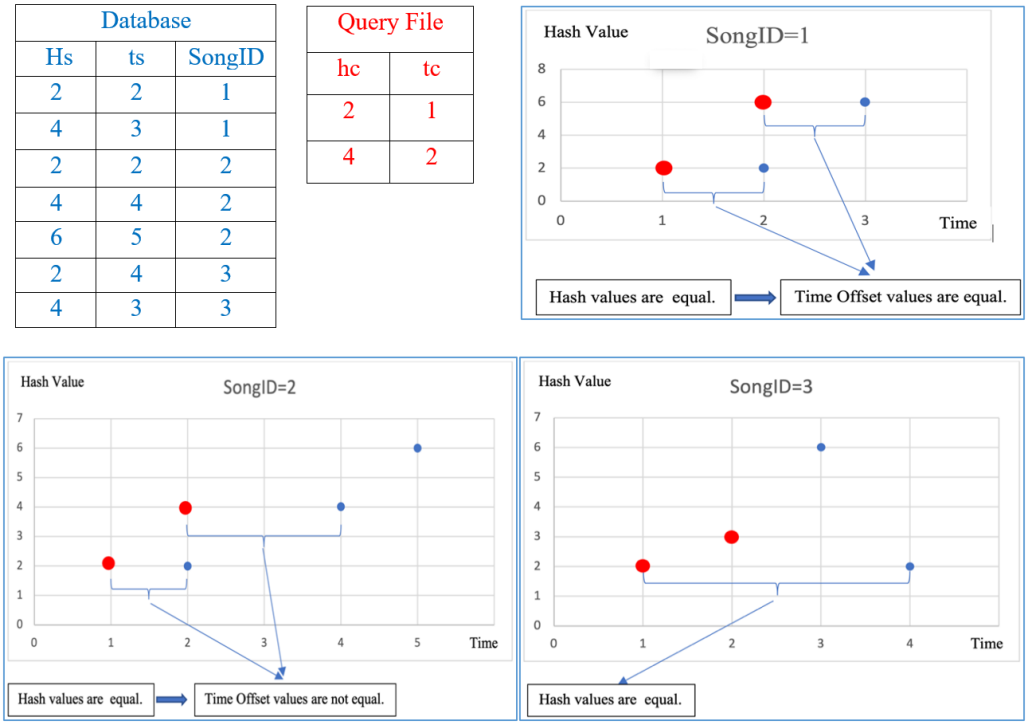


Figure 8. *Template matching*

After hashing procedure was completed in terms of number of matchings (NM) for noiseless case, the reliability and robustness of the method were analyzed for several SNR levels of different noise types such as restaurant, traffic and additive white Gaussian noise (AWGN). Moreover, the effects of variations in spectrogram parameters such as window length, overlap ratio and applied FFT number, on the success rate are evaluated in terms of the number of matchings (NM), reliability and robustness for different noise types and levels. In literature, the effects of some different AWGN levels and four different windowing functions have been analyzed, using similar algorithms [24, 25].

3. RESULTS AND DISCUSSION

In this study, various types of noises (AWGN, restaurant, traffic) were added to a randomly selected 10 sec segments of 10 randomly selected audio files from a database consisting of fifty songs. The spectrograms for a typical audio sample of 10 sec for the noiseless case, with AWGN, restaurant and traffic noises are respectively shown in Figure 9. In the experimental procedure, the different noise types were embedded at different levels onto the defined 10 sec sample parts of each tested song. The noise was added such that to have SNR values between -6 dB and 12 dB, and increased by 3 dB steps for each of these noise types. For testing process, 9 distinct databases were prepared for 9 different triple combinations of spectrogram parameters composed of window size, overlap and FFT number. Afterward the developed program was executed, and the results obtained for different noise types were recorded.

In order to determine the effects of spectrogram parameters, while one of these parameters was varied, the other two parameters were kept constant, and each time the NM values and recognition times were recorded. This process was performed for each parameter to identify its effect on the matching level. For different noise conditions the procedure is repeated, the average rates obtained for NM and corresponding recognition times for ten audio files were presented in Table 2. As seen from the table, the average NM and recognition times vary depending on the spectrogram parameters (window length; percentage of overlap of subsequent windows; number of FFT). While the worst NM values were obtained for the parameters of 512;25%;512, the maximum NM value was obtained for the parameters of 128;50%;512 in the noiseless case. However, without changing the spectrogram parameters where the maximum NM values were obtained, when the same samples were exposed to noise, a quite high rate of decrease in NM compared to other cases was observed. There are some other spectrogram parameter combinations giving high NM rates

in noiseless case, but considering the negative effect of the noise on NM, it is seen that 512;50%;512 and 512; 50%;1024 spectrogram parameters become salient in comparison with the other parameter combinations shown in Table 2.

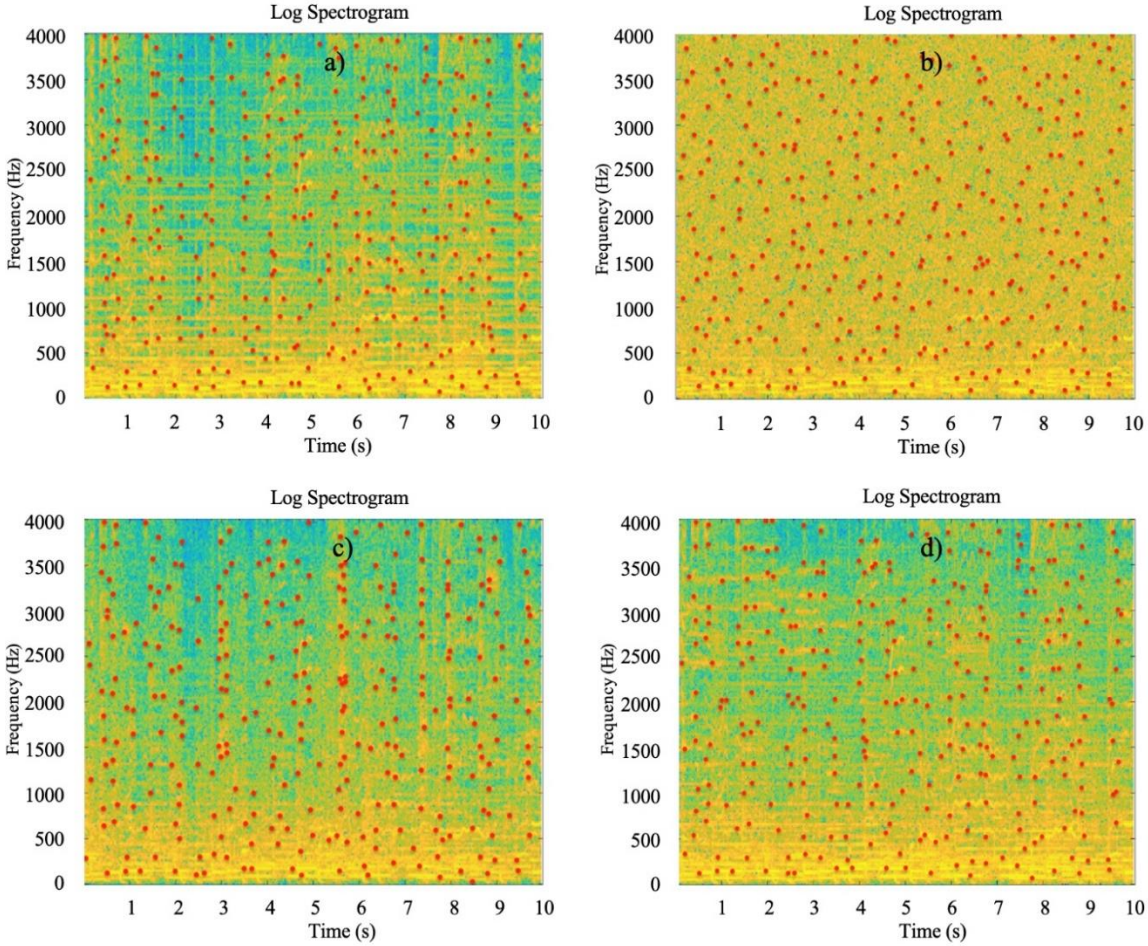


Figure 9. *Spectrogram outputs of a 10-second part of an audio file (SongID=18) including spectral peaks shown with red dots for different SNR values and noise types: (a) Noiseless, (b) 3 dB AWGN noise, (c) 4.8 dB restaurant noise, (d) 13.5 dB traffic noise*

Table 2. *Average number of matching and corresponding average recognition time values for 10 sec parts of 10 audio files according to different spectrogram parameters and noise types*

	Variable Combinations of Spectrogram Parameters			Noiseless		AWGN SNR: 3 dB		Restaurant Noise SNR: 3 dB		Traffic Noise SNR: 3 dB	
#	Window Size	Overlap	Number of FFT	Average of NM	Average of Recog. Times (s)	Average of NM	Average of Recog. Times (s)	Average of NM	Average of Recog. Times (s)	Average of NM	Average of Recog. Times (s)
1	128	50%	512	636	5.565	32.4	4.788	107.9	5.389	55.2	5.264
2	256	75%	256	339.3	5.083	33.3	4.616	86.6	4.694	51.7	4.684
3	256	75%	512	225	4.956	27	4.384	61.4	4.424	47.2	4.479
4	256	50%	256	248.7	7.877	42.3	6.233	86.2	7.081	67.1	6,784
5	256	50%	512	271.5	4.698	42.6	4.163	84.4	4.156	69.6	4.164
6	512	75%	512	256.9	4.385	43.3	4.392	87	4.221	69	4.104
7	512	25%	512	73	5.752	30.4	5.044	33.3	5.468	32	5.283
8	512	50%	512	351.4	5.778	69.8	5.112	113.8	5.304	106.2	5.13
9	512	50%	1024	304.3	4.392	47.8	4.179	91.4	4.253	81.3	4.185

A classical presentation giving the NM values for 18th song compared to the other files in the database was shown in Figure 10. In this figure, the reliability, which is one of the most important indicators in the

detection of correct audio file, was analyzed comparatively according to the NMs by considering noiseless case as well as different noise types. The numbers of matchings were presented for four different cases in the first, second, third and fourth columns, respectively. The numbers of matchings were shown for three different spectrogram parameter combinations. Here, the program was tested for a 10-sec randomly selected segment of song id-18, in the aspect of the success in song detection, for different spectrogram parameters and noise types.

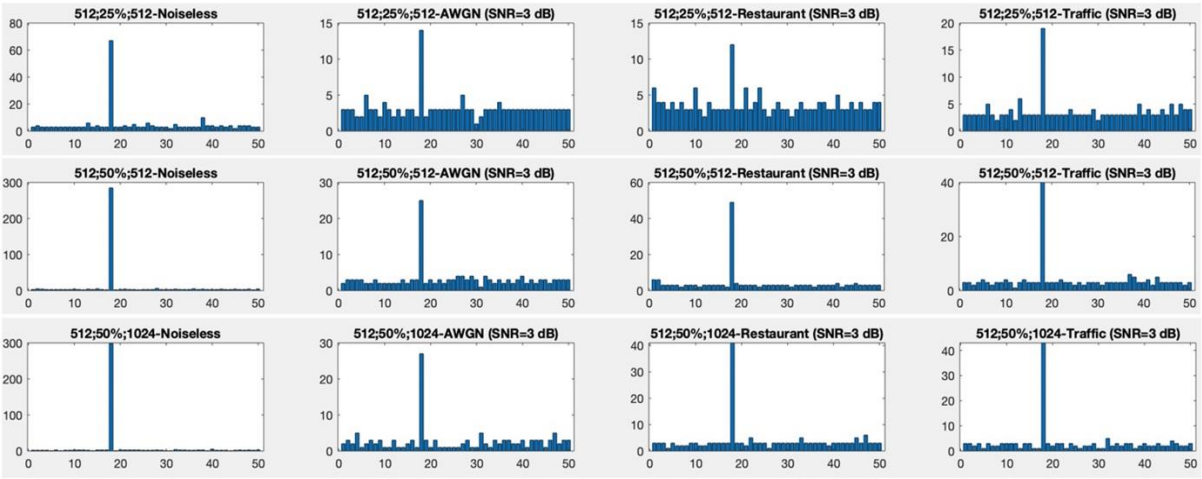


Figure 10. *Simulation results showing song-id (horizontal axis) versus the numbers of matchings (vertical axis) for 512;25%;512, 512;50%;512 and 512;50%;1024 spectrogram parameter combinations in different noise conditions for a randomly selected 10 sec part of 18th song*

As seen from Figure 10, the program has detected the desired song with a great accuracy in all conditions, and the song, which was searched in the database, dominated among other songs. It is seen that the NMs has decreased normally with the increase of noise (SNR=3 dB), and the amount of the decrease showed variations due to the type of the noise. In terms of NMs, it was observed that the most and the least influential noise types are AWGN and restaurant noise, respectively. In all cases, the correct song has been successfully detected with a much higher matching rate in comparison with the other audio files even in noisy cases. Also, it was observed that for 512;50%;512 and 512;50%;1024 spectrogram parameter combinations the algorithm gives a good result, even in noisy cases. This test has been also done for other songs, and similar results have been obtained. Due to the large relative differences in matching numbers detected with the searched song and the other audio files, it can be considered that any extension to be made in the database will not considerably affect the success in the detection of the correct song. As seen from Table 2, NM and recognition time values can give some information about the effects of spectrogram parameters and noise on the reliability of the method. However, this information may not be sufficient to determine the optimal spectrogram parameters. Besides of NM, the decrease in NM due to the different noise types and levels should also be considered. For this reason, it was thought that another criterion, which considers the noise effect on the number of matchings, would be more useful to be used for a better evaluation. We called this criterion as "normalized relative reliability (NRR)" and calculated with the formula given as in Equation (5)

$$NRR\ (\%) = 100*(NMD - NMC)/NMD\ . \tag{5}$$

In Equation (5), NMD is the number of matchings of the song searched in the database, and NMC is the number of matchings of any song which has the closest score to NMD. For a better analysis, NRRs were calculated and presented by considering different noise types and SNR values. For this purpose, the averages of NRR values were calculated for 10 sec parts of ten different songs handled in Table 2. The variations in these values with respect to the increasing SNR levels for different spectrogram parameter combinations were presented in Figure 11. As seen in figure, the NRR values increased with the increase of SNR value as expected.

In Figure 11(a), the effect of window size was analyzed for the window size values of 128, 256, 512, while keeping overlap and FFT number constant as 50% and 512. As seen in Figure 11(a), for SNR values 3, 6, 9 and 12 dB, the average NRR values seemed to be close to each other and did not differ significantly with the change in window size. But, when the noise effect was increased and hence SNR values decreased to the values -6, -3 and 0 dB, it was seen that the window size value of 512 became salient for the level of reliability.

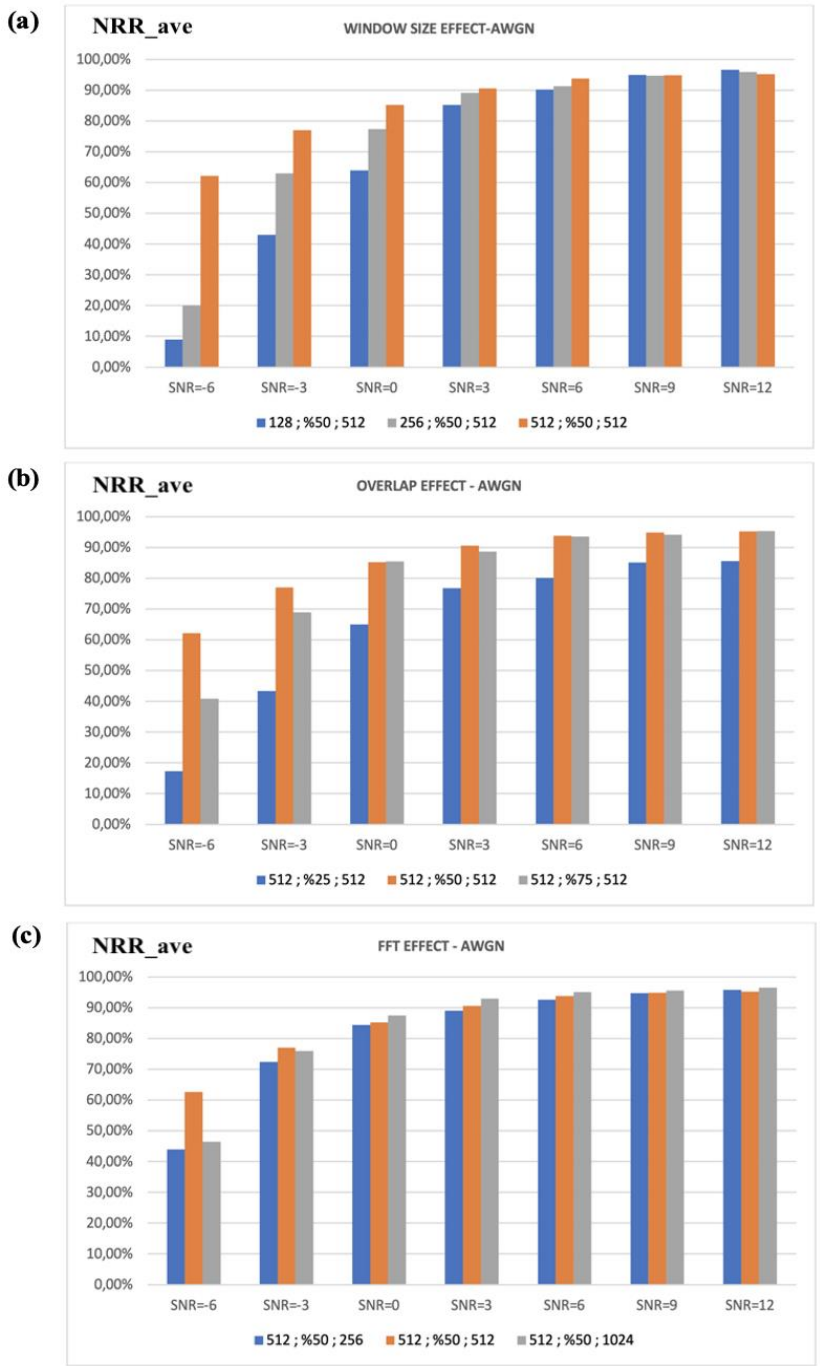


Figure11. *Change of average NRR values for 10-sec parts of ten different songs versus increasing SNR levels for AWGN noise by considering the change in the spectrogram parameters: (a) window size, (b) overlap, (c) FFT number*

In Figure 11(b), the effect of overlap was analyzed for 25%, 50%, 75% and keeping the window size and FFT number constant as 512. As seen in the Figure 10, for the SNR values 0, 3, 6, 9 and 12 dB, average NRR values seemed to be close to each other for 50% and 75% overlap values, while it was relatively low

for 25%. Nonetheless, when SNR values decreased to the values -6 and -3 dB, it was seen that the results obtained for the overlap value of 50% were better.

In Figure 11(c), the effect of FFT number was analyzed for the values of 256, 512, 1024 and keeping window size and overlap constant as 512 and 50%. As seen in Figure 11(c), for the SNR values -3,0,3, 6, 9, 12 dB, average NRR values seemed to be close to each other and did not differ significantly with the rate of change of FFT number. But, when SNR value decreased down to -6 dB, it was seen that the FFT number of 512 came into prominence.

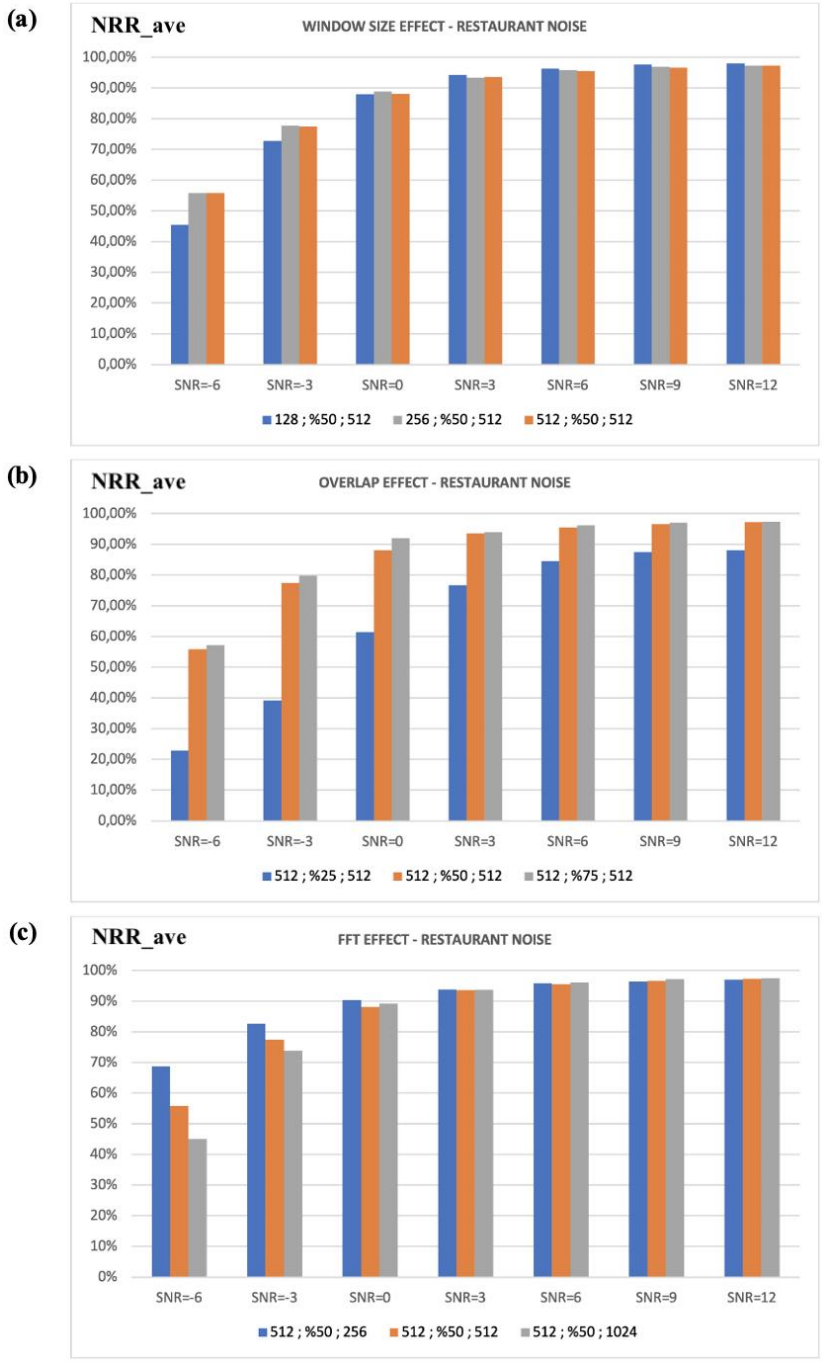


Figure 12. *Change of average NRR values for 10-sec parts of ten different songs versus increasing SNR levels for restaurant noise by considering the change in the spectrogram parameters: (a) window size, (b) overlap, (c) FFT number*

In order to observe the effect of different noise types on the audio recognition process, similar analyzes were conducted with respect to different SNR levels for restaurant and traffic noises respectively. As seen from Figure 12(a), while the window size had no significant effect at low noise conditions, at high noise levels (for SNR values -3 and -6 dB) it became effective. Window size values of 256 and 512 were relatively seemed to be better for a good NRR.

In Figure 12(b), the overlap effect was presented. As it can be seen the overlap values 50% and 75% were suitable, but 25% overlap value was not. Figure 12(c) shows the effect of FFT number for 256 and 512. These two values were found to be suitable even in the case of very high noise contaminations.

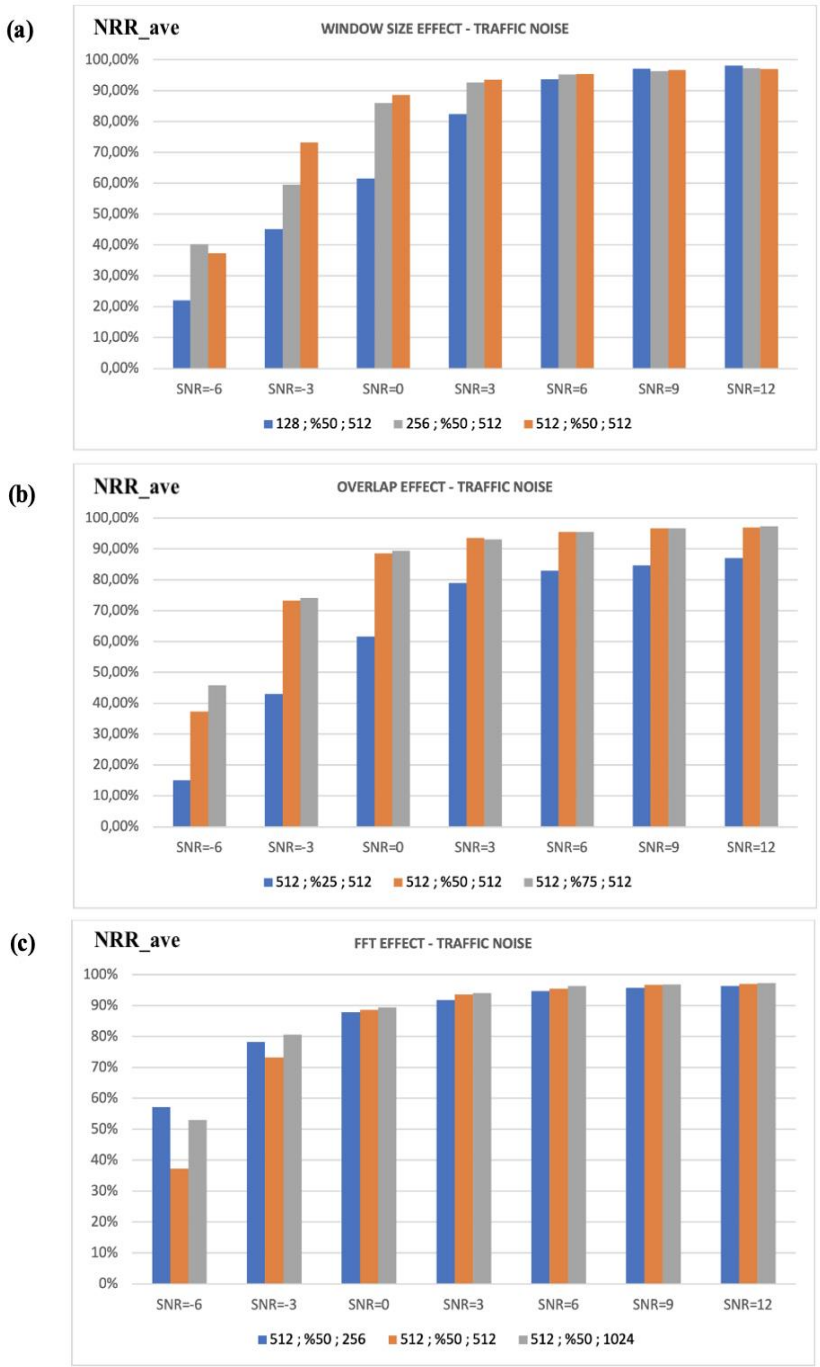


Figure 13. *Change of average NRR values for 10-sec parts of ten different songs versus increasing SNR levels for traffic noise by considering the change in the spectrogram parameters: (a) window size, (b) overlap, (c) FFT number*

Figure 13 shows the effect of different spectrogram parameters on NRR when SNR increases for traffic noise. As seen in Figure 13(a) and 13(b), the results are almost similar to those obtained for AWGN and restaurant noise cases. In this case window size of 512 and 256, and overlaps of 50% and 75% were relatively good choices even at high noise levels, but no significant effect at low noise levels. When the effect of FFT number was considered, according to Figure 13(c), the FFT numbers of 256 and 1024 were significant especially for low SNR levels of -3 dB and -6 dB. Also, it was seen that the change in FFT number did not have a noticeable effect on the NRR at higher SNR values, which is inline with the results obtained in the previous tests for AWGN and restaurant noises. In terms of NM, NRR, and RT, when all conditions considered, it can be concluded that the spectrogram parameter combination 512;50%;512 (window size, overlap and FFT number) would be good choice in audio recognition process.

When overall results are considered, it was observed that the changes in FFT number have affected the number of matches in low level. The effects of the changes in overlap and window size parameters were determined as almost equal, but the overlap effect on the number of matches is relatively a little bit higher in comparison with the window length effect. This result was obtained by considering all noise conditions. It was seen that the additive white Gaussian noise is quite effective on the number of matches while the restaurant noise has relatively no very important effect on the number matches.

Also, a confusion matrix including the number of matches for ten different songs have been presented for the noiseless case, as in Table 3. In the matrix, the spectrogram parameters were taken as 512;50%;512. As seen in the Table 3, the numbers of matchings, so the possibility to find the correct song, are quite high for the mentioned combination of spectrogram parameters. According to these parameters, it is seen that the success rate is almost over 91%. This ratio can be evaluated as satisfactory for these parameters.

Table 3. *The confusion matrix showing the numbers of matchings for 10-sec parts of ten audio files according to the spectrogram parameter combination of 512;50%;512 in the noiseless case*

SongID	4	9	15	18	21	26	33	37	40	42		
4	480	3	3	6	4	6	4	4	5	4	92.5%	7.5%
9	3	439	2	3	2	3	3	3	3	3	94.6%	5.4%
15	3	3	355	3	4	3	3	3	3	5	92.2%	7.8%
18	3	3	5	285	4	3	3	3	3	4	90.2%	9.8%
21	3	3	3	3	327	7	4	3	8	3	89.8%	10.2%
26	3	6	3	3	5	284	5	3	6	2	88.8%	11.3%
33	5	3	3	9	6	5	249	3	5	3	85.6%	14.4%
37	3	3	3	3	3	3	3	320	3	2	92.5%	7.5%
40	3	3	3	3	3	4	3	4	380	2	93.1%	6.9%
42	3	3	3	2	3	3	3	3	3	395	93.8%	6.2%
	94.3%	93.6%	92.7%	89.1%	90.6%	88.5%	88.9%	91.7%	90.7%	93.4%		
	5.7%	6.4%	7.3%	10.9%	9.4%	11.5%	11.1%	8.3%	9.3%	6.6%		

4. CONCLUSIONS

In this study, the effects of spectrogram parameters on the reliability of an audio search program based on the spectral peaks method were investigated by considering different noise levels and types. Firstly, a program, which is inspired from Wang's study and based on the spectral peaks method, was written for this study. After the validation of the method for different songs in regard to recognition times and NM, the algorithm was tested for different tripartite spectrogram parameter combinations such as window size, overlap and FFT number. The used algorithm was basically applied to the 10-sec parts of ten songs under

different noise conditions. In all conditions, the applied algorithm has found the song searched in the database successfully.

When all the results are analyzed together, it was observed that the changes in spectrogram parameters did not affect the reliability and robustness of the algorithm significantly under low noise conditions. However, when the noise level was increased gradually, for the lower SNR rates, the spectrogram parameter combinations 512;50%;512 and 256;75%;256 were significant in regard to NRR. One can choose one of these spectrogram parameter combinations depending on the field of application. However, when the recognition time, number of matches and robustness against noise are taken into account together, it is seen that the spectrogram parameter combination 512;50%;512 shows the best performance. Although, NM and reliability could change with noise, it was observed that this change does not have a noticeable effect on the determination of proper spectrogram parameter combination. Also, the effect of the music genre on the NM was analyzed, but any obvious effect was not observed.

As a future work, effects of time-stretching and pitch shifting on the audio recognition process are planned to be studied with and without these three different noise levels and types. The retrieval performance may be compared after Audio classification is realized by using LSTM method which is presented in previous works [50,51]. Also, time-frequency information of the audio files may be extracted with the help of WSST (Wavelet Synchro-squeezed Transform) method, since this method reduces the spectral leakage that occurs in time-frequency graph. The effect of this method on the audio retrieval is planned to be evaluated. In addition, some artificial intelligence methods such as deep learning and machine learning may be used for audio file classification after the fingerprints obtained from each audio file are converted to constant sized images.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

APPENDIX

A. Algorithm used to extract audio fingerprint

Input: Audio File
Output: Audio Fingerprint Data of The Audio File
Steps
1: Make audio file in mono structure, then downsample at 8 kHz
2: Starting to calculate the spectrogram of the audio file (S)
Set window size, overlap and FFT values
3: Detect the spectral peaks on the spectrogram

```
rowStep=x;
while (lastLineOfRow<=numberOfRowLines)
columnStep=y;
while (lastColumn<=numberOfColumnLines)
find (max(S(|rowStep|,|columnStep|)))
thresholdValue=max(s)*thresholdCoefficient
for r=x:lastRow
for c=y:lastColumn
if (S(r,c)>=thresholdValue)
peakMap(r,c)=1
end
y=y+columnStep
lastColumn=lastColumn+columnStep
```

```
        end
        x=x+rowStep
        lastRow=lastRow+rowStep
        end
        end
        end
```

4: Calculate hash values of spectral peaks

Hs→ h (f1, f2, t2-t1) =(t2-t1) *2^16 + f1*2^8 + f2

5: Create audio fingerprint database table

[Hs: ts: songID]

B. The experimental hardware and software environments

All experiments were performed in MATLAB 2019b on a computer with Intel® Core i5 – 7360U CPU, 2.30 GHz, 8 GB of Memory hardware specifications and operating system with macOS Monterey.

REFERENCES

[1] Grosche, P., Müller, M., Serra, J., "Audio Content-Based Music Retrieval", in: M. Müller, M. Goto, M. Schedl (Eds.), Multimodal Music Processing, Dagstuhl Follow-Ups, 157–174, (2012).

[2] Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M., "Content-Based Music Information Retrieval: Current Directions and Future Challenges", Proceedings of the IEEE, 96 (4): 668–696, (2008).

[3] Cano, P., Batlle, E., Kalker, T., Haitsma, J., "A Review of Audio Fingerprinting", Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, 41(3): 271–284, (2005).

[4] Cano, P., Battle, E., Mayer, H., Neuschmied, H., "Robust Sound Modeling for Song Detection in Broadcast Audio", AES 112th Convention, Munich, 1–7, (2002).

[5] Haitsma, J., Kalker, T., "A Highly Robust Audio Fingerprinting System", International Conference on Music Information Retrieval, Paris, 1–9, (2002).

[6] Haitsma, J., Kalker, T., "Speed-change resistant audio fingerprinting using auto-correlation", IEEE International Conference on Acoustics, Speech, and Signal Processing, IV-728–31, (2003).

[7] Cremer, M., Froba, B., Hellmuth, O., Herre, J., Allamanche, E., "AudioID: Towards Content-Based Identification of Audio Material", AES 110th Convention, Amsterdam, (2001).

[8] Fenet, S., Richard, G., Grenier, Y., "A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting", 12th International Society for Music Information Retrieval Conference, Miami, 121–126, (2011).

[9] Wang, A.L., "An industrial-strength audio search algorithm", International Conference on Music Information Retrieval, Baltimore, Maryland, 7–13, (2003).

[10] Yan K., Hoiem, D., Sukthankar, R., "Computer Vision for Music Identification", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1: 597–604, (2005).

[11] Jia, M., Li, T., Wang, J., "Audio Fingerprint Extraction Based on Locally Linear Embedding for Audio Retrieval System", Electronics, 9 (9): 1483, (2020).

[12] Baluja, S., Covell, M., "Waveprint: Efficient wavelet-based audio fingerprinting", Pattern Recognition, 41(11): 3467–3480, (2008).

[13] Pucciarelli G., "Wavelet Analysis in Volcanology: The Case of Phlegrean Fields", Journal of Environmental Science and Engineering A, 6: 300-307, (2017).

[14] Chen, D., Zhang, W., Zhang, Z., Huang, W., Ao, J., "Audio retrieval based on wavelet transform", IEEE 16th International Conference on Computer and Information Science, 531–534, (2017).

[15] Liu, N., Gao, J., Jiang, X., Zhang, Z., Wang, Q., "Seismic Time–Frequency Analysis via STFT-Based Concentration of Frequency and Time", IEEE Geoscience and Remote Sensing Letters, 14(1): 127–131, (2017).

[16] Nan, C., "Research on Intelligent Vocal Music Training System Based on Wavelet Transform", IEEE 4th International Conference on Information Systems and Computer Aided Education, 278–282, (2021).

[17] Park, M., Kim, H.R., Yang, S.H., "Frequency-Temporal Filtering for a Robust Audio Fingerprinting Scheme in Real-Noise Environments", ETRI Journal, 28(4): 509–512, (2006).

[18] Kim, H.-G., Kim, J.Y., "Robust Audio Fingerprinting Method Using Prominent Peak Pair Based on Modulated Complex Lapped Transform", ETRI Journal, 36(6): 999–1007, (2014).

[19] Anguera, X., Garzon, A., Adamek, T., "MASK: Robust Local Features for Audio Fingerprinting", IEEE International Conference on Multimedia and Expo, 455–460, (2012).

[20] Tao, S., Getachew, Y., High Fidelity Song Identification via Audio Decomposition and Fingerprint Reconstruction by CNN and LSTM Networks, Stanford University Report, http://cs230.stanford.edu/projects_spring_2020/reports/38911459.pdf. Access date: 11.05.2022

[21] Chang, S., Lee, D., Park, J., Lim, H., Lee, K., Ko, K., Han, Y., "Neural Audio Fingerprint for High-Specific Audio Retrieval Based on Contrastive Learning", IEEE International Conference on Acoustics, Speech and Signal Processing, 3025–3029, (2021).

[22] Báez-Suárez, A., Shah, N., Nolazco-Flores, J.A., Huang, S.H.S., Gnawali, O., Shi, W., "SAMAF: Sequence-to-sequence Autoencoder Model for Audio Fingerprinting", ACM Transactions on Multimedia Computing, Communications, and Applications, 16(2): 1–23, (2020).

[23] Altalbe, A., "Audio fingerprint analysis for speech processing using deep learning method", International Journal of Speech Technology, (2021).

[24] Koseoglu, M., Uyanik, H., "The Effect of Different Noise Levels on The Performance of The Audio Search Algorithm", IEEE International Congress on Human-Computer Interaction, Optimization and Robotic Applications, 1–7, (2020).

[25] Uyanik, H., Koseoglu, M., "Performance Evaluation of Different Window Functions for Audio Fingerprint Based Audio Search Algorithm", IEEE 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, 1–4, (2020).

[26] Han, B.B., Hou, Y.H., Zhou, L., Shen, H.Y., "A Filtering Method for Audio Fingerprint Based on Multiple Measurements", Proceedings of the International Conferenc on Information Technology and Computer Application Engineering, Hong-Kong, 377-381, (2014).

[27] Zhang, Q.Y., Xu, F.J., Bai, J., "Audio Fingerprint Retrieval Method Based on Feature Dimension Reduction and Feature Combination", KSII Transactions on Internet and Information Systems, 15(2): 522–539, (2021).

[28] Wang, D., Xuewei, Z., "THCHS-30 : A Free Chinese Speech Corpus", ArXiv, (2015).

[29] The 500 Greatest Songs of All Time, https://www.rollingstone.com/music/music-lists/best-songs-of-all-time-1224767/. Access date: 28.05.2021

[30] Yan, B.C., Liu, S.H., Chen, B., "Modulation spectrum augmentation for robust speech recognition", Proceedings of the International Conference on Advanced Information Science and System, Singapore, 1–6, (2019).

[31] Gupta, V., Mittal, M., "QRS Complex Detection Using STFT, Chaos Analysis, and PCA in Standard

and Real-Time ECG Databases", Journal of The Institution of Engineers (India): Series B, 100(5): 489–497, (2019).

[32] Ellis, D., "Robust Landmark-Based Audio Fingerprinting", https://www.ee.columbia.edu/~dpwe/resources/matlab/fingerprint/. Access date: 28.06.2021

[33] Suriñach, E., Márquez, E.L.F., "A Template To Obtain Information On Gravitational Mass Movements From The Spectrograms Of The Seismic Signals Generated", Earth Surface Dynamics Discussions, 1–34, (2022).

[34] Walker, J.S., Don, G.W., Mathematics and Music, Chapman and Hall/CRC, (2019).

[35] Bracewell, R., The Fourier Transform & Its Applications, McGraw-Hill, (2000).

[36] Cohen, L., Time-Frequency Analysis, Electrical Engineering Signal Processing, Prentice Hall, New Jersey, (1995).

[37] Shie, Q., Dapang, C., "Joint time-frequency analysis", IEEE Signal Processing Magazine, 16(2): 52–67, (1999).

[38] Gabor, D., "Theory of communication. Part 1: The analysis of information", Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering, 93(26): 429–441, (1946).

[39] Hill, P., Audio and Speech Processing with MATLAB, CRC Press, (2018).

[40] Castanié, F., Digital Spectral Analysis, John Wiley & Sons Inc, Hoboken, NJ, USA, (2011).

[41] Lukin, A., "Adaptive Time-Frequency Resolution for Analysis and Processing of Audio", AES 120th Convention, Paris, 1–10, (2006).

[42] Boashash, B., "Heuristic Formulation of Time-Frequency Distributions", in: B. Boashash (Ed.), Time-Frequency Signal Analysis and Processing, Elsevier, 65–102, (2016).

[43] Heisenberg, W., "Uber den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik", Zeitschrift Fur Physik, 43(3–4): 172–198, (1927).

[44] Paliwal, K.K., Lyons, J.G., Wojcicki, K.K., "Preference for 20-40 ms window duration in speech analysis", IEEE 4th International Conference on Signal Processing and Communication Systems, 1–4, (2010).

[45] Practical Introduction to Time-Frequency Analysis, Mathworks, www.mathworks.com/help/signal/ug/practical-introduction-to-time-frequency-analysis. Access date: 10.07.2021

[46] Schneier, B., Applied Cryptography, John Wiley & Sons, Inc, (1996).

[47] Haitsma, J., Kalker, T, Oostveen, J., "Robust Audio Hashing for Content Identification", in: Int. Workshop on Content-Based Multimedia Indexing, Brescia, 4: 117-124, (2001).

[48] Cuff, P., ELE301:Signals and Systems-Labs, Fall Semester 2011-12, Princeton University, https://www.princeton.edu/~cuff/ele301/files/Lab5_2011.pdf. Access date: 22.07.2021

[49] Cuff, P., ELE301:Signals and Systems-Labs, Fall Semester 2011-12, Princeton University, https://www.princeton.edu/~cuff/ele301/files/Lab6_2011.pdf. Access date: 22.07.2021

[50] Tombaloglu, B., Erdem, H., "Turkish Speech Recognition Techniques and Applications of Recurrent Units", Gazi University Journal of Science, 34(4): 1035-1049, (2021).

[51] Banuroopa, K., Priyaa, D.S., "MFCC based hybrid fingerprinting method for audio classification through LSTM", International Journal of Nonlinear Analysis and Applications, 12 (Special Issue), 2125-2136, (2022).