Radio Broadcast Monitoring to Ensure Copyright Ownership

E.D.N.W. Senevirathna[#1], K.L. Jayaratne[#2]

***Abstract*— In this research, we provide a way to protect copyright ownership of multimedia objects like songs. Protecting ownership is very important when we are working in a corporative environment. If someone tries to misuse others' property it is an illegal action. But capturing theses kind of actions especially on electronic objects are very difficult. There are several ways to share copyright ownership with others. We can purchase others' property and use them under conditions provided by the owner.**

In this research, we are focusing on protecting ownership of audio songs broadcasted by radio channels. There are so many radio channels in a country, as well as a huge number of songs are broadcasted per day. In order to broadcast particular song in a radio channel, they should purchase the right to do so. In order to make sure that this process is functioning correctly, we have to monitor radio channel and extract the broadcasted songs. Currently, this is happened manually. Manual process is no longer able to continue because the number of songs is increasing day by day. We provide an automated solution for this real-world problem.

There are so many researches on this area done by various researches. They used different approaches to achieve the final goal. Most of them have used content base audio identification base approaches. In this context, an unknown audio file is identified by analysing the content of it. But, it is very hard to find researches that use this concept to monitor radio channel since there are additional complexities when we come to monitor radio channel. In this research, we extend the content-based audio identification approach to monitor radio channel automatically.

First, we perform a pre-process on raw audio objects that broadcasted by a radio channel. After extracting interest areas, we generate set of hash values. Then, we perform powerful approximate matching against pre-stored hashes. Ultimately, we provide a detailed report which contains all the information of broadcasted audio songs.

***Keywords*— Audio fingerprint; features extraction; playlist generation; wavelets; broadcast monitoring, silence detection**

I. INTRODUCTION

Ownership is the key building block in the development of any human society. Over the millennia and across cultures, notions regarding what constitutes "property" and how it is treated culturally have varied widely. The definition is not the case, but we should protect, respect the ownership of others' property.

Manuscript received on 16th Jan 2018. Recommended by Dr. M. G. N. A. S. Fernando on 02nd July 2018.
This paper is an extended version of the paper "Automated Audio Monitoring Approach for Radio Broadcasting Channels in Sri Lanka" presented at the ICTer 2017.

E.D.N.W. Senevirathna is from University Colombo School of Computing. (*nishan.senevirathna@gmail.com*).
Dr. K.L. Jayaratne is a Senior Lecturer at the University Colombo School of Computing. (*klj@ucsc.cmb.ac.lk*).

The electronic song repository is increasing day by day as well as radio stations try to broadcast songs as much as possible within a day. To broadcast a song, radio station must purchase the right. That means radio station should pay some pre-agreed amount to the owner of the song. Usually, song repositories are managed by an association in such a case radio station should settle the final bill to that association. This is a legislation rule imposed by the government of most of the countries. Just to make sure that this process is working smoothly we need a monitoring process. This is still a manual process and will not work in future with the increasing song repository. This is the real-world issue which we are going to address in this research. We provide an automated solution for this.

There are so many representations of electronic objects like audio songs. If we take two different file types of audio songs like Mp3 and wav, those can also be considered as different digital representations. In this research, we store songs as fingerprints. The audio fingerprint is a compressed unique representation of an audio object. There are multiple advantages of storing audio objects as finger prints. First, this is a compact representation. Hence it consumes a very less storage space. We can represent an audio song by countable fingerprints. Thus, we can easily develop very powerful searching algorithms.

Automated radio monitoring is not a very easy task compared to automated audio song identification. Most of the time, radio channels alter the original song blue prints by adding some other audio objects such as commercials, talks and so on. Because of that, end users do not receive the original songs as it is. In this research, we handle this kind of situations. Apart from that, there might be implicit distortions as well. Radio signals might be destroyed due to environmental effects. It will also cause to change the uniqueness of audio songs.

However, our goal is to identify songs even though those are altered by the above cases. In order to reduce false positives and improve the performance, we invoke pre-filtering process before identifying songs. During this pre-filtering process, all the non-song objects will be removed. This will help us to improve the system accuracy and the efficiency as well.

The process can be divided into two major parts. First, we register songs to the system. In this process, we extract a considerable amount of audio fingerprints from a song. Those fingerprints are stored in the database. The second process is the song identification. The input to this process is a raw audio stream of a radio channel. After extracting songs, audio fingerprints are generated following the same process as song registration. After that, by using special matching algorithm we identify broadcasted songs.

Our ultimate goal is to generate a detailed report which includes broadcasted summary of a radio channel. Finally, the radio station should settle the bill according to this report.

Using this automated process, we can ensure that the intellectual property right of artists are protected.

II. RELATED WORKS

Even though there are so many researches on content-based audio identification [1] [2] [3] [4], it is very hard to find researches on automated FM channels monitoring. However, the heart of our research is also a content-based audio identification. Most related researches on content-based audio identifications (CBID) are discussed below.

CBID is a very broad area of research. The audio fingerprint representation is one of implementations of CBID. Audio fingerprinting or Content-based audio identification (CBID) systems extract perceptual digests of a piece of audio content. When presented with unlabelled audio, its fingerprint is calculated and matched against those stored in the database [21]. As we know, fingerprints of human can be used to identify people uniquely. Likewise, audio fingerprints can be used to identify peace of audio object uniquely. There is not a specific implementation of audio fingerprint. Researches implement audio fingerprint in many different ways. Our proposed approach is also based on this. In order to create an audio fingerprint, we need to extract very strong audio features which should be able to survive against external effects such as noise. This means that audio features should be robust in any kind of alterations. An ideal fingerprinting system should satisfy several requirements. It should be able to accurately identify an item, regardless of the level of the compression and distortion or interference in the transmission channel [21].

Most of the researches have tried to find the most stable audio features [5] [6] [7] [8] [23]. Following are the widely used stable audio features.

- Mel-Frequency Cepstrum Coefficients (MFCC)
- Spectral Flatness Measure (SFM)
- Peaks of the spectram
- Zero crossing rate

Mel Frequency Cepstral Coefficients (MFCC), which is a stable audio feature to be used for audio analysing [10]. This feature is widely used for speech recognition. To obtaining MFCC, we have to transform raw audio data into a machine friendly format.

Spectral Flatness Measure (SFM) is widely used to classify audio objects. Sometimes, it is also called "tonality coefficient", it is used to quantify how much tone-like a sound is, as opposed to being noise-like. The meaning of "tonal" in this context is in the sense of the number of peaks or resonant structure in the power spectrum, as opposed to flat spectrum of a white noise [22]. GSFM (Generalize SFM) is applied to the problem of voicing determination in speech signals i.e. it can also be used to recognize and classify speeches. According to the most of the researches, SFM can be considered as a widely used very stable audio feature.

According to most of the researches [9], peaks of an audio spectrum is another stable audio feature. Always noise like unnecessary law energy audio disturbances are spread around the zero axis. Refer the Fig 1, which shows the audio file destroyed by noise. Peaks of the original spectrum were not altered by the noise. According to that key feature, we can use peaks or peaks related some other features to identify an audio object objects uniquely.

There are so many other stable audio features which can be used to manipulate audio files. Spectral similarity can be considered as an audio feature which is widely used for scene classifications. Linear prediction coefficient derived cepstral coefficients (LPCCs) is also used by many researches. 'Zero crossing rate' is another feature which can be used to classify audio object into different groups. For an example, non-song audio objects have high value for zero crossing rate than song objects. MPEG-7 descriptors, entropy and octaves are also used to analyse audio objects, for more details please refer [11] [12] [13] [20].

Some of researches have used neural network based approaches to manipulate audio files like [14] [17] which are also a better way to identify audio patterns and classify audio objects but we cannot directly use it for song recognition.

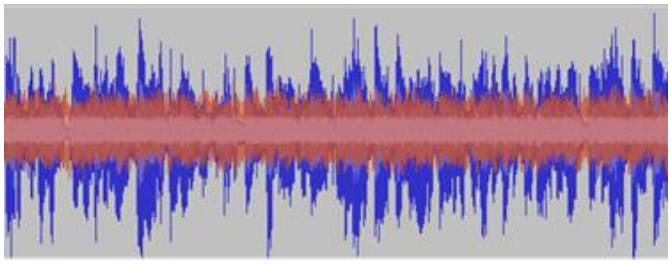

Fig. 1. This image shows the wave pattern of an audio file (blue colour) and wave pattern of noise (red colour). Peaks of audio file is not altered or destroyed by this noise

III. METHODOLOGY

We used a very powerful set of algorithms to extract audio features, create audio fingerprint and match against stored fingerprints. When we were designing the solution, following factors were considered.

- How to handle destroyed songs by noise or some other effects.
- How to handle implicit alterations like adding commercials, playing a part of a song and so on.
- How to improve the performance by removing non-song objects like commercials.
- Address the scalability requirements against thousands of audio tracks.

Most of the time, we will not be able to get the original song blue print from a radio channel. The reason is that they alter the songs by adding various unnecessary audio object like commercials before broadcasting. Considering all of those practical situations, we have divided our implementation into two major steps.

1. Song registration
2. Song identification

In the first step, required original songs will be registered into the database. After registering songs into the database, we can use them when finding unknown audio objects.

During the song registration process, audio features are extracted, and fingerprints are generated using them. Then, the created fingerprints will be stored in the database.

Song identification process can be divided further into four major steps.

1. Pre-processing
2. Feature extraction
3. Searching
4. Post-processing

The overall flow of the system is shown in the Fig 2. The major processes of both of above steps will be discussed in the following sub sections.

A. Registering

At the initial step, we store original songs in the database. There might be multiple versions of the same song. In monitoring process, only the registered version of a song will be identified. It is one of the limitations of this research. This process can be divided into several sub processes. Among them, feature extraction is one of the major processes and it is a common process for searching as well. Please refer the Fig. 3, which shows the user interface of song registration of the system. We use Service Oriented Architecture (SOA) to implement the system which has a number of advantages. For an example, we can plug any front end easily without changing the back end logics. We upload the song which we are going to register in the application server using FTP protocol. After that, we work with the uploaded copy of the song. This facilitates us to decouple client from the server. After uploading a song, the client can visit later to get the information of registered song like whether it was successfully registered to the system or not.

B. Preprocessing

Before starting the matching process, several operations are applied on the raw audio objects. All the operations which happen before the matching process can be considered as pre-processing actions. First, we split audio object (Song) into 40 seconds long sets of frames. Then, we can process those frames independently. This facilitates us to process frames parallel and gain efficiency. Apart from that, there is a number of advantages/reasons of framing. Following are the major reasons for framing.

- Songs might be destroyed by unwanted environmental effects such as noise or commercials. Usually, they are added into the middle of the song. But if a song is split into several frames, then only one or two frame/s is/are destroyed. We can identify the song using the other frames which are not destroyed.
- We can process frames parallel so that we can improve the efficiency of the overall process.
- A frame is a very small audio object hence processing is trivial and efficient.
- We can apply preprocessing (pre-filtering) on each frame and discard unwanted frames. This will improve the overall accuracy and performance of the system.

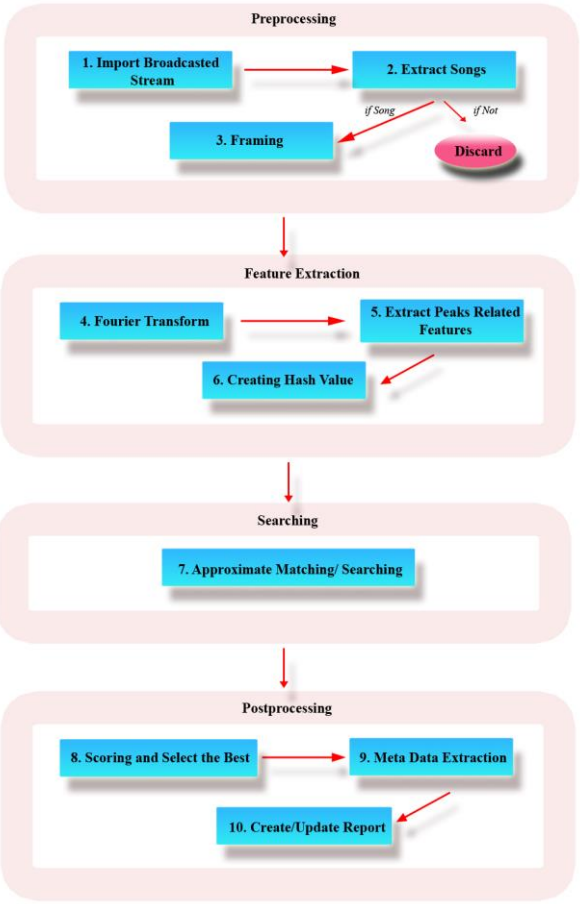

Fig. 2. The flow of the system. There are four major processes and sub processes of each major process.

Fig. 3. Song Registering User Interface. User can upload a song or set of songs to the server. Later user can start registration process which will be executed in the server.

For more information about this framing process, please refer the Fig 4. Those frames might be consisted of different audio objects like songs, commercials and so on.

We record broadcasted radio stream as raw audio files and store them in a storage device. We process files in offline.

After splitting audio files into 40 seconds long frames, those are processed one by one. Frames processing can be done in parallel. In here, we mainly perform two operations on each frame.

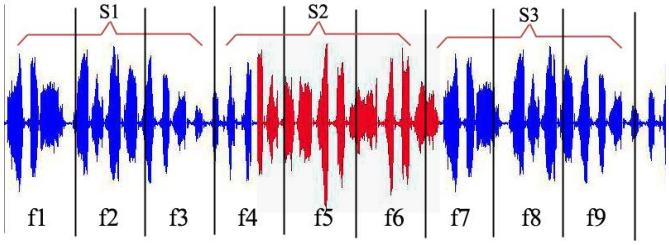

Fig. 4. Splitting input audio object into set of frames. This input audio object might be consisted with mixer of audio objects like song, commercials and so on.

1. Convert the format of the audio file into a new format which is more flexible and easy to use in future steps.
2. Identify non-song objects like commercials, dramas, vocals etc. Those non-song objects will be removed from the process.

We store broadcasted radio streams as mp3 or wav format. But we do not need much information to create a unique fingerprint. Therefore we convert the audio file format and resample the file. This resampled file will be used to generate fingerprints. This approach helps to reduce the size of the fingerprint storage, improve the performance and reduce the false positives as well.

Filtering non-song objects is another important operation in this preprocessing stage. Removing non-song objects like commercials gives us several advantages. To process non-song objects, we have to allocate valuable system resources unnecessarily. It will cause to go down the system performance. Secondly, if non-song frame matches with a song because of some reason, it will cause to reduce the overall system accuracy. However, identifying non-song objects is not the ultimate goal. Therefore, the accuracy of this process will not directly affect the accuracy of the overall system. For an example, suppose a non-song frame is not identified as a non-song object. Then, it will be processed further. But, ideally this will not be identified as a song by the song identification process which will be discussed later.

However, we can separate some audio objects like dramas from song objects easily. The reason is that we can see prominent differences between such an audio object and song object.

There are so many approaches to extract songs from a mixture of audio objects which contain non-song objects like commercials, speech, songs and so on. Here, we do not want to expect high accuracy since song extraction is not the final goal. Therefore, we use a very simple approach to distinguish/extract songs. If we listen to a song and a non-song object like drama carefully, we can observe a prominent key difference i.e. the distribution of silences or low energy points. Silences are very limited in song objects but those are very frequent in non-song objects. Refer the Fig. 5. When we are speaking, we keep pauses at the end of each word. When a singer is singing a song, again there are pauses between words but those gaps are filled by musical instruments. In this research, we extract songs using this key feature. Sometime, this is known as zero crossing rate of an audio object.

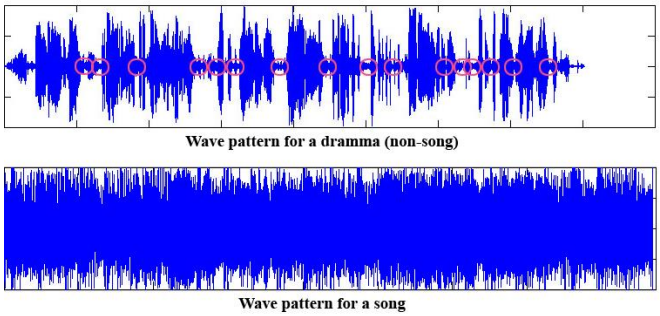

Fig. 5. Distribution of silent points over non-song and song objects. There are frequent silences on non-song objects such as dramas. But it is very hard to find silences on song objects.

Silent point cannot be defined absolutely since it depends on the overall energy of the frame. As a solution to this issue, we define silent cut off energy point to a frame. A particular energy level of a frame may be a silent but the same level may not be a silent for another frame.

C. Feature extraction

The feature extraction can be considered as the heart of the system since the accuracy of the system directly depends on this. The default data representation of an audio object is time domain and it is fragile. The reason is that a minor modification can change the time domain data representation drastically. Most of the time, we cannot get the original song without having any alteration from a radio channel. Therefore, if we use the time domain signal representation to process audio objects, it will not be robust. As a solution to this issue, we convert time domain signal into frequency domain. The frequency domain representation is proportional to the energy of the audio object. A considerable effect is required to change the energy of a frame. It means that frequency domain signal representation is very stable and robust [15] [6].

To convert time domain signal into frequency domain one, we used Short Time Fourier Transformation (STFT) on each frame. There are several control parameters of Fourier Transformation. The window size is one of such parameters. STFT applies on a sliding window. We used 4096 bits long window. The window overlapping size is another important parameter. We use 2048 bits long overlapping area. For more details, please refer the Fig 6.

The next step is to extract stable audio features to create an audio fingerprint. According to the past researches and our findings, peaks of an audio object are very stable high energy positions. Those points cannot be altered easily. If several peaks are destroyed due to external effect such as noise, that means all the other low energy points should also be destroyed by that noise.

In such case, we cannot hear a song at all. All the other cases peaks will be survived. Therefore, we can use peaks to create a unique signature for a given audio object.

We cannot use peaks independently to create a fingerprint. If we do so, we will not be able to use those to match unknown audio object with fingerprints already stored in the database. Instead of that, we use several adjacent peaks and combined them using a special hashing function.

In this research, we only consider Sinhala songs. Usually,

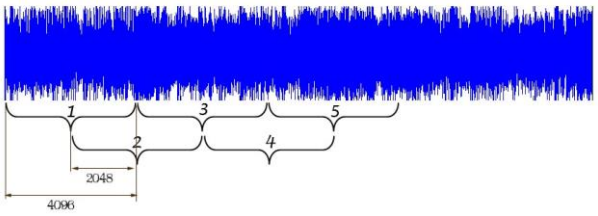

Fig. 6. Key controlling parameters on Short Time Fourier Transformation (STFT). We used 4096 bits long window with 2048 bits long overlapping area.

there are no very high frequencies or very low frequencies in Sinhala songs. Therefore, we select the mid-range of frequencies to extract features. After that we divided this range into five bins and find local peaks from each bin. Using a special hashing function as we discussed earlier, we combine those five peaks and create a special hash value. Please refer the Fig. 7 for more details. If we take one such a value it is corresponding to one particular window. We assigned a number for each window starting from 1. These numbers can be considered as the local time of each window since all the windows are disjoined. Then, we store the obtained hashes with its time value and a unique ID assigned to the song which can be considered as the song ID in the data base. According to this design, one hash key can be mapped into several songs.

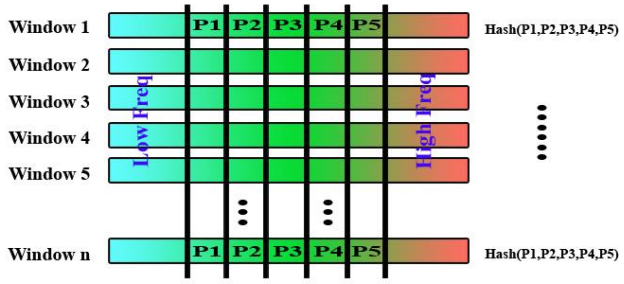

Fig. 7. The way of extracting peaks. Those peaks are combined together and create a hash value

D. Searching audio objects

Searching is also a very important process since this directly affects the overall accuracy of the system. We are working with millions of records hence we have to design this process very carefully.

To identify an unknown peace of audio file, as we discussed earlier we extract features and create fingerprints following the same process as done in registration. In here, we find the mostly matched audio track by scanning the database which contains a huge amount of fingerprints of millions of tracks. We cannot use any traditional searching algorithms like brute force search since we have to do this within a restricted time frame.

To do this, we use a special searching algorithm. Obviously, we cannot expect one to one mapping here. Therefore, we perform an approximate matching and scoring base mechanism to identify songs.

Searching algorithm takes set of previously created hash values as the input. Our goal is to find out the most similar database track for those values. But, the challenge is that most of the time these two sets of hash values might not be matched 100%. Because of this nature, we used the term approximate matching. Assume that an unknown song is converted to N number of hashes. In this stage, matching song IDs are obtained for each and every hash from the set N. As the next step, the received result was analysed to identify the unknown song. We use scoring-based approach to extract the most matched song.

Assume that we have inserted 15,000 songs to the database. Each song was divided into 7 frames and 50 hashes were extracted from each frame. According to this, the total number of hashes is equal to 15,000*7*50 = 5,250,000. Even if it is possible for the brute force search to find a matching hash against 5,250,000 hashes, it is not practical as this takes a considerable amount of time as well as resources. Therefore, we should introduce a new approach to search fingerprints. Instead of searching fingerprint against millions of tracks, we will align them using a special way. Searching algorithm will be discussed later in detail. To search fingerprints quickly, we will store them in a look up table. Please refer Fig 8 for more details. There might be several corresponding songs for a hash key.

As we discussed earlier, we store unique fingerprints and local time information of the original song blueprint into the database. The starting point of the song is considered as the local time zero. Then, we extract features from an unknown audio clip by following the same set of steps as we did in the song registering process and calculate hash values and its local time values. Again, the clip starting point will be considered as the local time zero. Even if those unknown hashes should be matched with some set of hashes of the database, the time portion will not be matched. The reason behind this is that we have registered the complete song to the database. But, an unknown segment of a song can be extracted from anywhere in the original song.

According to the above analysis, we should align the unknown audio clip with its original songs by sliding it from the beginning. However, we cannot do this easily since we cannot get the starting point of songs by scanning the database. Instead, we follow another efficient way as described below.

We have sets of hashes and time values of the unknown clip. First, we took one hash key (H1) and its time value (T1). Then, we recorded the time differences between T1 and all the time values of matching hashes in the database. At the same time, we keep the unique song IDs of those matching hashes. After doing this for all the hashes of the unknown clip, we could see that there is a considerable amount of equal delta time offsets for a specific song. If so, we could take this song as the matching one otherwise there is no any matching song in the database. We used some threshold value to determine whether an unknown clip matches with existing song or not. Refer the Fig. 9 for more details.

A. Post-processing

This is not a dificult task after doing all the above steps successfully. At this moment, we have idetified the matching songs correctly. Now, we can use those information to provide findings in human readable manner. In this newly identified song. We keep all the Meta data like singer, musician and the author of identified song. As well as when it was played in the radio channel. Finally, we will update the payment information corresponding to the identified song as well.

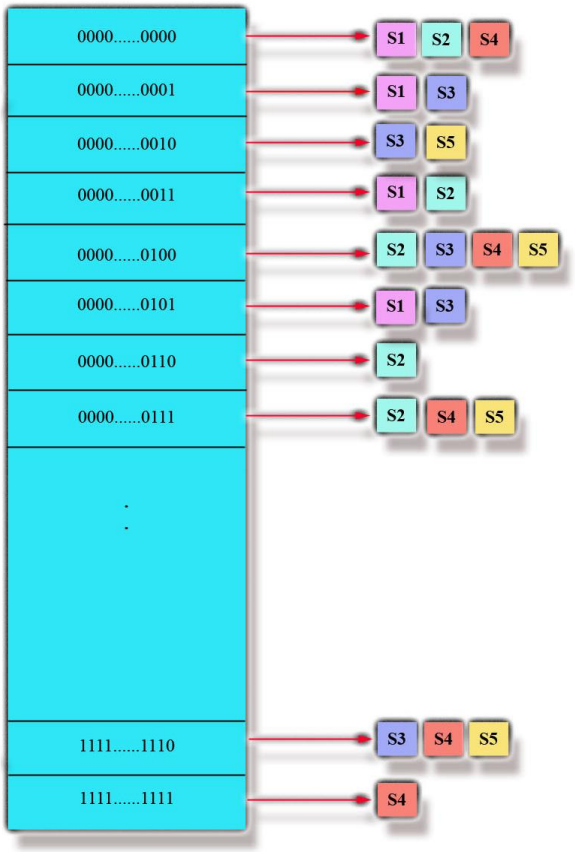

Fig. 8. Fingerprint hashes storage structure. This structure helps us to implement efficient searching algorithms. One hash key may hold more than one song. As well as a song has multiple hash keys.

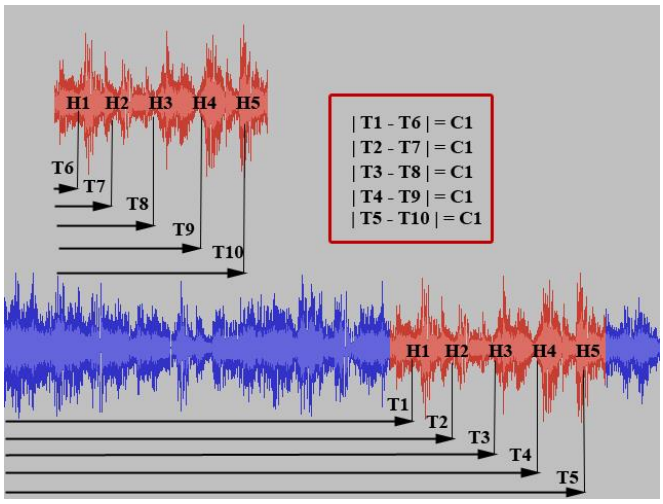

Fig. 9. Aligning unknown audio object against already registered songs in the database. Use delta time offset to find the matching song.

Sometime, we might want to verify manually whether the identified song is correct or not. We keep one frame of the matching song as a backup to use in such a case.

When registering songs to the database, we take payment information for that song as well. Those information will also be used when creating the final report. All the generated reports will be stored in the server. Any client who has permission can download those repots and can be used to protect the ownerhip.

IV. Results

It was a bit hard to evaluate the research since there is an infinite number of situations which can occur in a radio channel. However, we tried to cover most of the practical situations. Before evaluating the research, we registered 1500 original songs to the database. Even though the amount of songs is small, there were 54,685 unique hash keys and 3,892,715 hash values. We selected songs which belong to different genre as well. Therefore, our evaluation is fair and unbiased.

We evaluated the research under three major categories.

- Performance evaluation
- Song extraction process evaluation
- Accuracy evaluation

We executed all the test cases in a normal application server which has very limited hardware configurations as below.

- Processor: Intel core i3
- Processor Speed: 3.1 GHz
- RAM: 4 GB
- OS: Windows7 (64 bit)

A. Performance evaluation

The performance is a very important evaluation criterion. We processed audio files offline. As we discussed earlier, we divided a larger audio file into a set of 40 seconds long frames. Under this performance evaluation, we measure the average time taken for processing one frame. Fig. 10. shows the obtained result for 500 frames. There are some outliers as well. The reason for those outliers might be unexpected behaviors of database connection and JVM. However, we obtained 14.13 s average time to process 40 seconds long frame.

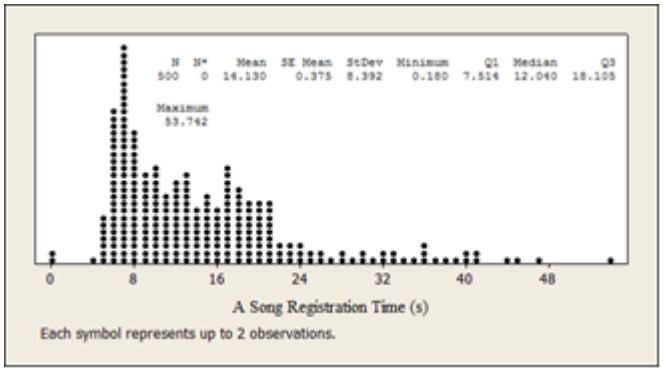

Fig. 10. Distribution of frames processing time. Result obtained by processing 500 frames.

This performance level is acceptable since we are processing audio frames offline. However, when database is growing with songs this time is also increased. We can manage this issue by splitting the database into several fragments.

B. Song extraction process evaluation

As we discussed earlier, we eliminate non-song frames without processing. We have evaluated the accuracy rate of this classifier. We extracted non-song objects such as commercials, talks, dramas, educational programs like

quizzes, news and so on from real broadcasted audio stream. We prepared 300 frames i.e. approximately 3.33 hours long audio stream. Then, we directed these frames to the song extraction module. This module classifies each frame as a non-song object or as a song object. Simultaneously, we do the same for song objects. We tested the system against 800 frames. It is equal to 150 songs. The behaviour of the module is shown in the Fig 11.

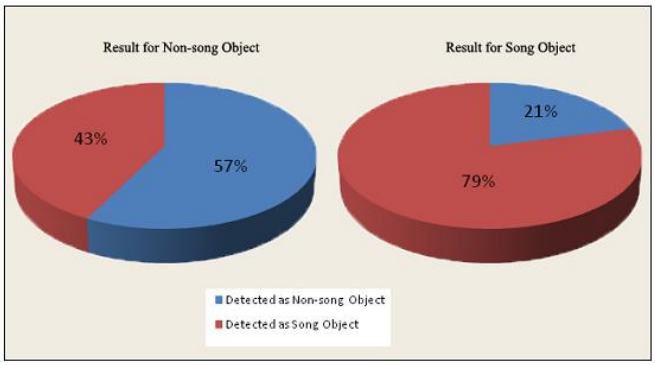

Fig. 11. Results Distribution of Song Extraction Module

Detecting non-song frame as a song object: Most of the commercials behave like songs. Therefore, we could see that some of non-song frames containing commercials were classified as song frames. Even though this is an error, it will not directly affect the accuracy of the overall system. Those frames will be discarded in song identification process since there will not be a valid matching song. But, this will cause to go down the performance of the overall system. However, we have achieved above 57% accuracy rate.

Detecting a song frame as a non-song frame: This is a critical error. According to the results, there is a 21% error rate. Actually here we have presented the result of the frames. Usually, a song is split into several frames. Even if a frame is classified as non-song frame, it will not be an issue since other frames will be classified as song frames. If we consider this test case under song level, we can achieve 100% accuracy rate. It means that all the frames of a song are not classified as non-song frames.

C. Accuracy evaluation

If we take an unknown audio frame, it can either have matching audio songs in the database or not. If it has a matching song, the system should identify it correctly. If it does not have a matching one, the system should say that there is no valid matching song. All the other cases are considered as false positives or errors.

We have identified nine scenarios which are commonly occurred in FM channels. Then, we collected a considerable amount of test cases for each and every scenario. The statistical information of each case will be discussed in following sections.

1) *Test Case 1(General Case)*: A radio station can play a song with alterations or without alterations. Here, adding commercials, vocals or some other audio objects to the middle of the song will be considered as alterations. Sometime, we can see that implicit alterations such as adding noise are also possible. Sometime radio stations play songs without altering the original blue print. In this test case, we considered this normal behavior. But, practically this is a really rare case.

The system provides four kinds of output for this test case.

- Provided unknown song can be identified correctly (success).
- When providing an unknown song which was not registered to the system and identify this case correctly (success).
- Provided unknown song can be matched with incorrect one (failure).
- Provided unknown song which was not registered can be matched with existing song (failure).

We extracted twenty random songs which are already registered in the database and twenty other songs which are not currently in the database. We have achieved 98.56% success rate and 1.44% error rate which occurred due to the incorrect match. The system could identify all the existing songs correctly as well as non-existing songs were totally omitted without matching with an existing one. But, the system introduced a completely new song as a match, it is an error. Refer the Fig 12 for more details which shows the distribution of the results.

2) *Test Case 2(Optimal Duration of Clips)*: In this case, we tried to obtain "required minimum broadcasting duration" of a song to be matched correctly. Again, we selected ten random songs which are currently in the database and ten other songs which are not currently in the database. Then, we extracted, 10 seconds from each and every songs, 15 seconds from each and every songs, 20 seconds, 25 seconds and up to 40 seconds. After that, we execute our system on each and every sample. Results are shown in the Fig 13. According to the Figure 10, to identify a clip correctly, it should be played at least 35 seconds. The detection rate is increased step by step when playing duration is increased. Therefore, we select 40 seconds as the optimal duration for the framing process (frame length).

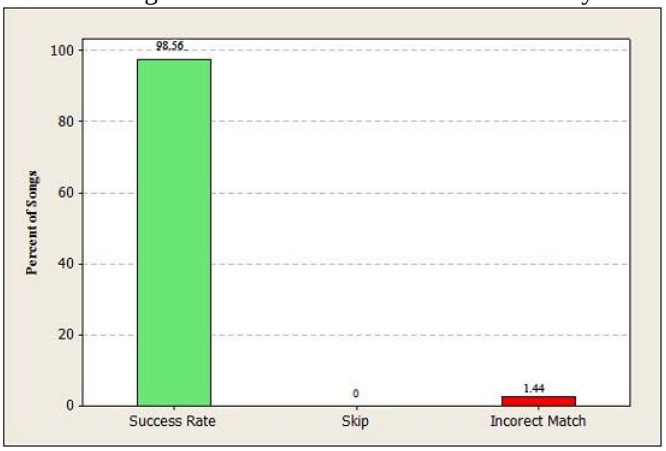

Fig. 12. Results Distribution of the general Case (Test Case 1)

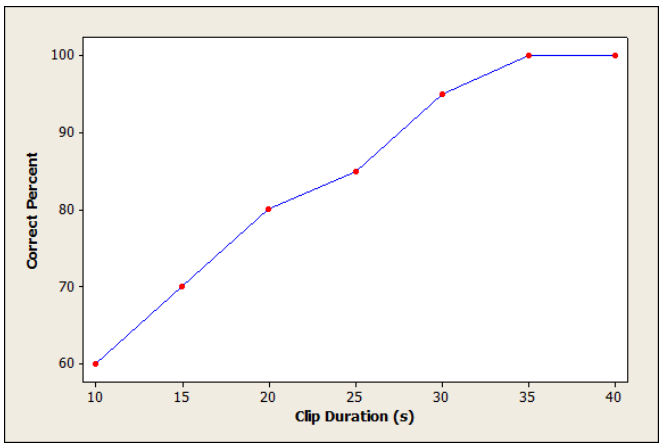

Fig. 13. Song identification Percentage against the Clip Durations (s)

3) Sequence of Songs from Same Singer (Test Case 3): Here, we test the behaviour of the system when more than one song are being played as a list without keeping any silence in between two songs. Most of the time, we feel that songs which are sung by the same singer as similar. Therefore, there is a high probability to match such a song with an incorrect one of the same singer. This probability may increase when two songs from the same singer fall into the same frame. Therefore, we select ten singers and three songs from each singer which are currently registered in the database. Then, these three songs are appended one after the other so that we obtained 10 clips. Moreover, we create similar data sets for songs which are not currently in the database. Ultimately, we obtained 60 songs and then we directed these songs to the system. The obtained results are shown in the Figure 14. We obtained 94.88 success rate and 5.12 error rate.

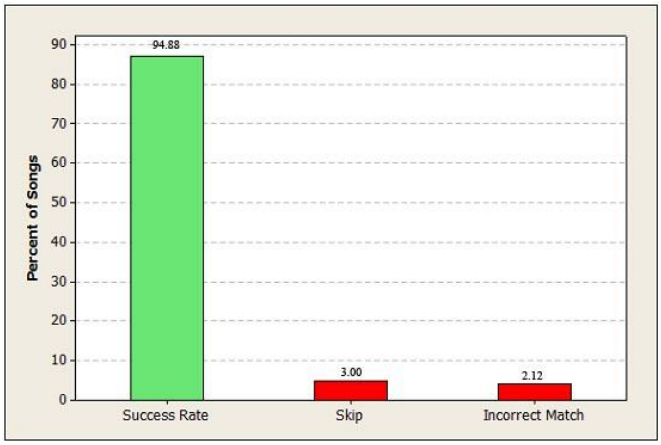

Fig. 14. Results distribution of test case 3. Three Songs from a Singer are Joined one after the other.

4) Sequence of Songs from Different Singers (Test Case 4): This is more or less similar to the test case 3. The difference is that we consider three songs from different singers instead of the same singer. Again, two songs can be mapped into the same frame. As the above test case, we prepared 60 songs and executed the system on this test case. The obtained results are shown in the Fig 15. In this case, we have achieved 90.16% success rate and 9.84% error rate. However, this is also a rare case. But, we have obtained higher success rate than the test case 3.

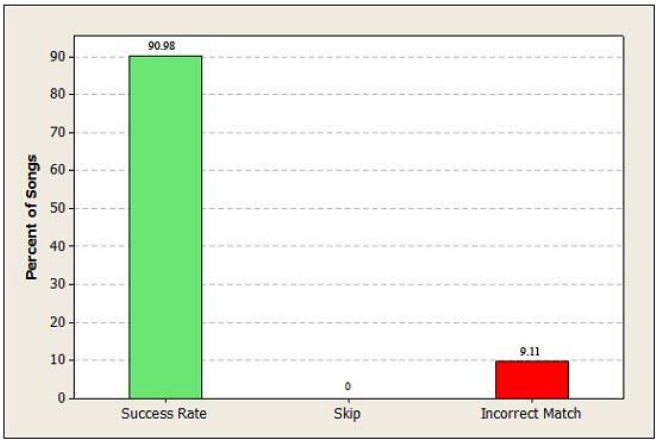

Fig. 15. Results distribution of test case 4. Three Songs from Different Singers are Joined One after the other

5) *Test for Non-song Objects (Test Case 5)*: This is another important test case as well as this is the most probable test case. We test the behaviour of the system for non-song object such as commercials, talkers and discussion and so on. We recorded actual non-song object from radio channels and then executed the system on this audio clips. First, we consider the obtained result which is shown in the Fig 16.

We directed 60 non-song frames to the system. None of them was matched with a song in the database. Every frame was skipped that means we achieved 100% accuracy rate in this case. It means our system performs very well against non song objects.

As we discussed earlier, during the pre-processing stage, the system tries to extract song objects. Suppose that the system identifies a non song object as a song object.

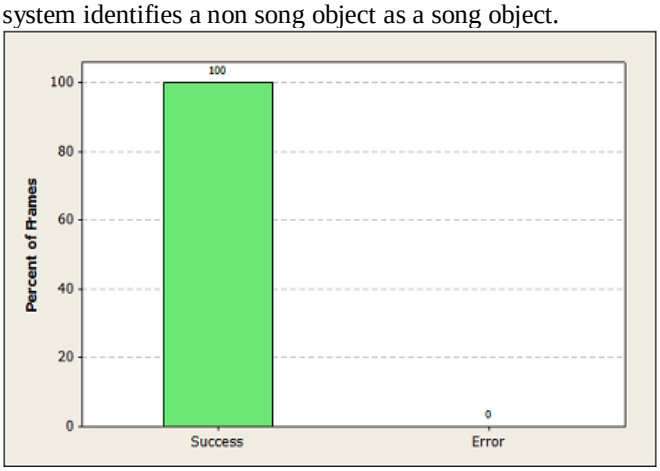

Fig. 16. Behaviour of the System against Non-song Objects.

In that case, non- song object will be processed without discarding. But according to this test case, it will not be matched with a song. Therefore, the overall system accuracy will remain unchanged.

6) *Test for Position Independency (Test Case 6):* In this test case, we tried to test two things. The first one is to test the system behavior against a part of a song presented and the second one is to test the system behaviour when we

present a position independent part of a song. Radio channel can play a part of a song which can be extracted from anywhere of the song. To test this, we extracted 100 second long clip from a random position of the song. Again we extracted ten, 100 second long clips from songs which are currently in the database and ten other, 100 second long clips from songs which are currently not in the database. Our framing approach will help us to handle this kind of situations successfully.

We have achieved 95% success rate and 5% skip error rate. This higher accuracy rate shows the ability of handling position independent short audio clips. Refer the Fig 17 for more details about the distributions of result.

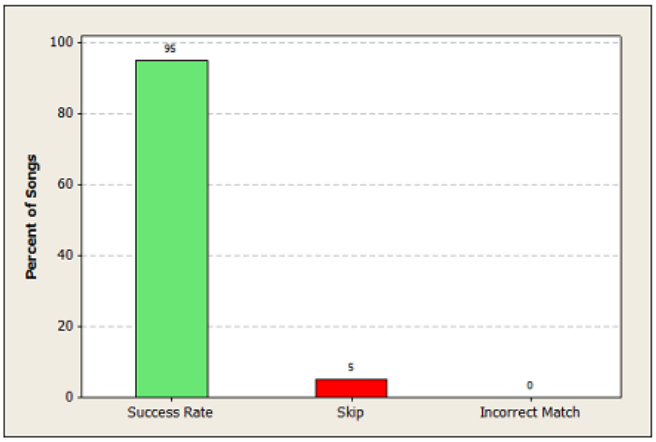

Fig. 17. Behaviour of the system against position independent short time (100 s) clips.

7) *Adding Commercials at the Middle of Song (Test Case 7):* Consider the following cases which are frequently occured in radio channels.

- Playing some commercials in the middle of the song.
- Stop playing song and play some short time commercial and resume the same song.

In this test case, we observed the behavior of the system against these two kinds of scenarios. We selected 10 cases with its song which are currently in the database and 10 other cases with its songs which are not currently in the database. Adding commercials did not affect the overall accuracy of the system since the framing process takes care of this kind of case as we discussed earlier. Here, we obtained 100% accuracy rate. Refer the Fig 18 for more details.

8) Songs with Background Watermark (Test Case 8): Usually radio channels add some commercials or talks as watermarks. In such a case, the song is not stopped. But, while it is playing, commercials are also playing in the background. There are two major kinds of watermarks.

- Volume of the song is reduced up to some extent and commercials are also playing.
- Song and commercial are playing at the same volume but we can hear song and commercial separately.

Normally, the second one is used for short time commercials. In this test case, we take 10 samples from the first type and 10 others from the second type. The obtained results from these 20 cases are shown in the Figure 19.

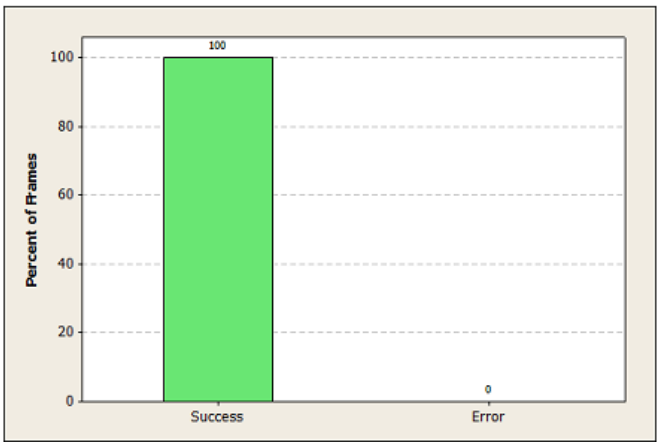

Fig. 18. The distribution of the results of playing commercials at the middle of the song.

We can see that the system has performed very well for this test case as well. There is no any positive errors. Therefore, we can conclude the fact that background watermarks are not a problem at all for the system accuracy. The framing process helps us very much in this case as well.

9) *Destroyed songs by external effects such as noise (Test Case 9):*This is the really important and the most probable case. When we are listening a radio channel, adding noises to the audio channel is obvious due to several external factors. Noises can be added to the radio channel due to channel frequency related issues, environmental disturbances, weak radio signal strength and so on. In this test case, we test the noisy level which can be handled by the system.

We generated different levels of continuous noises and then added them to the song. First, we generate a noise with the amplitude level of 0.1 and then it is mixed with 10 songs which are already in the database and 10 other songs which are not in the database. This process is repeated by changing the noisy level from 0.10 to 0.50. Ultimately, we generate 180 noisy destroyed songs. Those were processed by the system. The obtained results are shown in the Fig 20.

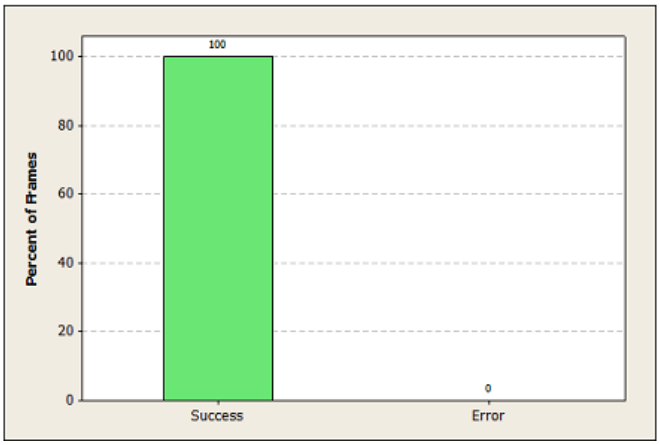

Fig. 19. The distribution of the results when playing songs with background watermarks

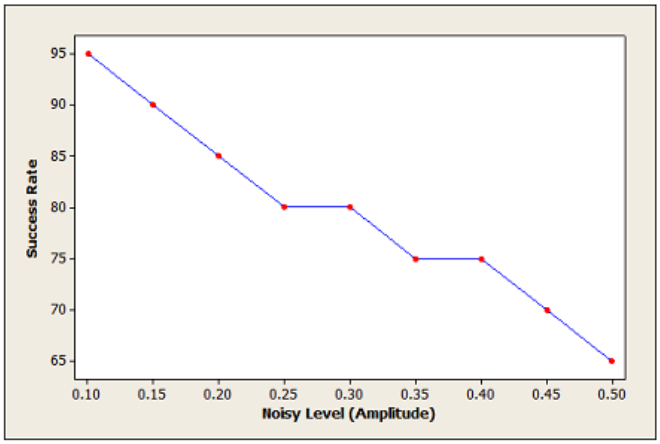

Fig. 20. The distribution of the results when playing songs with background watermarks

We have achieved a higher accuracy rate even if song is completely destroyed by the noisy with the amplitude level of 0.5. However, the noise is not held throughout the song when we come to a real situation. Therefore, we can expect even more higher accuracy rate.

10) *Different songs with the same melody (Test Case 10):* There can be more than one song with the same melody. For an example, there may be a sinhala song with the same melody of a hindi song. Suppose that sinhala version is already inserted into the database. We analysed the sytem behaviour when hindi version of that song is played. Here, we find 10 songs which are not currently in the database but each one has another song with the same melody which is in the database. Then, we execute those test cases on the system. We have achieved 100% success rate in this case as well. We can say that if the melody is already inserted to the database, different songs with the same melody will not be matched with it.

TABLE I
OVERALL SUMMARY OF TEST CASES

Test Case	***Error Rate (%)***	***Success Rate (%)***
General Case	*1.52*	*98.48*
Sequence of Songs from the Same Singer	*5.12*	*94.88*
Sequence of Songs from Different Singers	*9.11*	*90.89*
Continuous Non-song Objects	*0*	*100*
Short Position Independent Clips	*5*	*95*
Adding Commercials at the Middle of Song	*0*	*100*
Songs with Background Watermark	*0*	*100*
Noisy Destroyed Songs	*5*	*95*
Different Songs with the Same Melody	*0*	*5*
Overall Accuracy Level	***2.86***	***97.14***

11) *The summary of the overall accuracy of the research:* Following table shows the summary of all the test cases. Those test cases covered most of the scenarios which can happen in a live radio channel.

V. DISCUSSIONS

In this research we have addressed a practical problem especially in developing countries like Sri Lanka. Most of the countries monitor radio channels manually but it is not a trivial task as well as it is a very inefficient and less accurate process. When the number of radio channels and song repositories are growing, this process will be even more difficult. However, protecting copyright ownership is very important in corporative human societies.

We could provide a successful solution to this issue. According to the obtained result, we can see that we have achieved a high success rate. We could successfully handle almost all the real situations which occur in radio channels. Still this is in research level, but we can introduce this to the industry.

There are several areas to be improved in this research. This is our first attempt hence we covered only the basic requirements. Most of the time, radio stations might not broadcast the original songs. But, in the copy right point of view, those are referred to the same song. In order to track those kinds of situations, we have to register all the versions of a song. But, it is very difficult as well as there can be some practical issues as well. Even though we have achieved a high accuracy rate we could not achieve a high-performance rate. Therefore, we have to find ways to improve performance. For an example, we can categorize songs under several categories such as genre [19][18][16][17]. After that, we can find an unknown audio object in a narrowed searching space. Also, we can fragment the database under some conditions and it will also help us to improve the performance.

VI. ACKNOWLEDGMENT

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. K.L. Jayaratne, who has supported me throughout my research project with his patience and knowledge whilst allowing me the room to work in my own way.

I would also like to thank Outstanding Song Creator's Association (OSCA). This is a requirement of OSCA and they helped/direct us to do this research all the time.

VII. REFERENCES

[1] H. Neuschmied, H. Mayer, and E. Batlle, "Content-based identification of audio titles on the internet," in Web Delivering of Music, 2001. Proceedings. First International Conference on, pp. 96–100, IEEE, 2001.

[2] B. Denby, O. Romain, and S. Hariti, "A software radio approach to commercial fm content indexing," in *11th International Workshop on Systems, Signals and Image Processing, IWSSIP*, vol. 4, pp. 13–15, 2004.

[3] E. Nishan, W. Senevirathna, and K. L Jayaratne, "A highly robust audio monitoring system for radio broadcasting, Proceedings of sixth Annual International Conference on Computer Games, Multimedia and Allied Technology" *GSTF Journal on Computing (JoC)*, vol. 3, no. 2, pp. 87-98, 2013.

[4] N. Senevirathna and K. L Jayaratne, "Automated content based audio monitoring approach for radio broadcasting," Proceedings of sixth Annual International Conference on Computer Games, Multimedia and

Allied Technology (CGAT 2013), Singapore, pp. 110–118, CGAT, 2013.

[5] I. Cox, M. Miller, and J. Bloom, "Watermarking applications and their properties," in *Information Technology: Coding and Computing, 2000. Proceedings. International Conference on*, pp. 6–10, IEEE, 2000.

[6] G. Reynolds, D. Barry, T. Burke, and E. Coyle, "Towards a personal automatic music playlist generation alogorithm: the need for contextual information," in *Conference papers*, p. 11, 2007.

[7] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *ProcIntSymp Music Info Retrieval* (M. Fingerhut, ed.), vol. 32, pp. 107–115, Ircam - Centre Pompidou, 2002.

[8] E. N. W. Senevirathna and K. L. Jayaratne, "Audio music monitoring: Analyzing current techniques for song recognition and identification," *GSTF Journal on Computing (JoC)*, vol. 4, no. 3, pp. 23-34, 2015.

[9] A. Wang, "An industrial strength audio search algorithm," in *International Conference on Music Information Retrieval (ISMIR)*, vol. 2, 2003.

[10] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *In International Symposium on Music Information Retrieval*, 2000.

[11] Y. Demir, E. Erzin, Y. Yemez, and A. M. Tekalp, "Evaluation of audio features for audio-visual analysis of dance figures," in *2008 16th European Signal Processing Conference*, pp. 1–4, Aug 2008.

[12] M. Lakshitha and K. L Jayaratne, "Melody analysis for prediction of the emotions conveyed by sinhala songs," *European Journal of Compute Science and Information Technology (EJCSIT) by European Centre for Research Training and Development UK*, vol. 5, no. 1, pp. 11–32, 2016.

[13] M. Lakshitha and K. L Jayaratne, "Melody analysis for prediction of the emotion conveyed by sinhala songs," in *Proceedings of IEEE International Conference on Information and Automation for Sustainability (ICIAfS 2016), Sri Lanka.*, 2016.

[14] R. Amarasinghe and K. L Jayaratne, "Supervised learning approach for singer identification in Srilankan music," *European Journal of Computer Science and Information Technology (EJCSIT) by European Centre for Research Training and Development UK*, vol. 4, no. 6, pp. 1–14, 2016.

[15] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," *Proc. AES 112th Int. Conv*, pp. 1–7, 2002.

[16] R. Peiris and K. L Jayaratne, "Musical genre classification of recorded songs based on music structure similarity," *European Journal of Computer Science and Information Technology (EJCSIT) by European Centre for Research Training and Development UK*, vol. 4, no. 5, pp. 70–88, 2016.

[17] R. Peiris and K. L Jayaratne, "Supervised learning approach for classification of Sri Lankan music based on music structure similarity," in *Proceedings of ninth Annual International Conference on Computer Games, Multimedia and Allied Technology CGAT 2016), Singapore*, pp. 84–90, 2016.

[18] D. Chaturanga and K. L Jayaratne, "Automatic music genre classification of audio signals with machine learning approaches," *International Journal of Computing (JOC) by Global Science and Technology Forum (GSTF)*, vol. 3, no. 2, pp. 137–148, 2013.

[19] D. Chaturanga and K. L Jayaratne, "Musical genre classification using ensemble of classifiers," in *Proceedings of fourth International Conference on Computational Intelligence, Modeling and Simulation (CIMSim 2012), Kuantan, Malaysia*, 2012.

[20] K. Umapathy, S. Krishnan, and R. K. Rao, "Audio signal feature extraction and classification using local discriminant bases," *IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, pp. 1236–1246*, May 2007.

[21] *P. Cano, E. Batle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in 2002 IEEE Workshop on Multimedia Signal Processing., pp. 169–173, Dec 2002.*

[22] *S. Dubnov, "Generalization of spectral flatness measure for non-gaussian linear processes," IEEE Signal Processing Letters, vol. 11, pp. 698–701, Aug 2004.*

[23] E. D. N.W. Senevirathna and K. L Jayaratne, "Automated Audio Monitoring Approach for Radio Broadcasting in Sri Lanka," Proceedings of International Conference on Advances in ICT for Emerging Regions (ICTer 2017), Sri Lanka, pp. 92–98, 2017.