

## Content-Based Audio Retrieval by using Elitism GA-KNN Approach

**Sarah Hussein Toman**

Roads and Transport Department  
College of Engineering

University of Al-Qadisiyah  
[sarah.toman@qu.edu.iq](mailto:sarah.toman@qu.edu.iq)

**Muna Ghazi Abdul Sahib**

Computer Science Department

University of Technology  
[110096@uotechnology.edu.iq](mailto:110096@uotechnology.edu.iq)

**Zinah Hussein Toman**

Computer Science Department  
College of Computer Science  
and Information Technology

University of Al-Qadisiyah  
[Zinah.hussein@qu.edu.iq](mailto:Zinah.hussein@qu.edu.iq)

**Recived : 5\3\2017**

**Revised : 27\4\2017**

**Accepted : 7\5\2017**

### Abstract

The digital audio became very popular and important in a computer user experience. The increasing amounts of audio data requires improvement, new methods and algorithms for processing this information. In this paper, our approach proposed the GA-KNN approach (Genetic Algorithm with K-Nearest Neighbor as fitness function) for content-based audio retrieval. The input is an audio file (query) and the output is a list of audio files ranked by their similarity. The system first extracts the features from an audio database and audio query. The query pattern is considered as a boundary for comparison. Then, the initial population in a genetic algorithm is constructed from a database containing all audio features. To improve the results, this paper uses Cosine measure in the genetic algorithm, and an improved selection method selection method to prevent the fittest chromosomes from being wasted in the new population by adding an Elitism feature, using 4% Elitism count. Furthermore, we proposed a new crossover method (Close Up-Feature Crossover) to create a new offspring by comparison between two audio patterns to query pattern. Finally, we evaluated our approach using a well-known audio database, which contains 409 sound samples of 16 classes to give 0.71475 as a precision of the audio retrieval.

**Keywords:** audio retrieval system, feature extraction, Genetic Algorithm (GA), K-Nearest Neighbor algorithm (KNN), elitism fitness, close up-feature crossover.

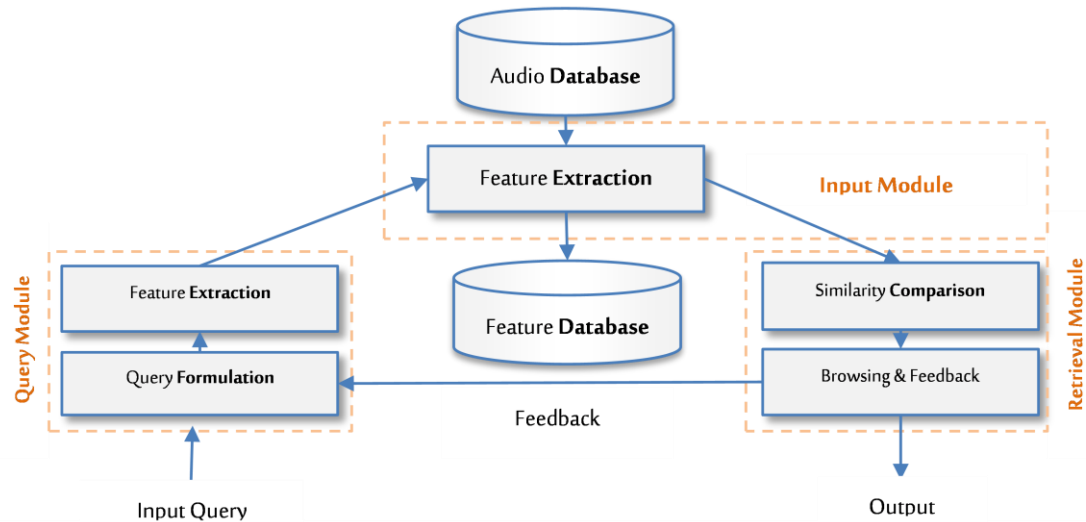
### 1. Introduction

Recently, there are many fields in content-bases audio retrieval, such as speech recognition, search engines, segmentation and music information retrieval. The identification of perceptually similar audio content is the main goal of content-based audio retrieval system (CBARS).

The typical Architecture of CBARS consists in three parts: input, Query, and

retrieval modules. The aim of the input and query module is to extract the features from an audio database and a query file. The purpose of feature extraction is to extract meaningful information from q signal and to decrease the amount of data. The retrieval module will perform a comparison between the features resulted from a query and the feature database records, to determine the items with similar properties. [1]. Figure 1 illustrates the components of a typical content-based audio retrieval system.

This work proposes content-based audio retrieval by using elitism GA-KNN approach: the relevant features are extracted and transformed from an audio database and query audio (those



**Figure 1:** The components and relations of a typical content-based audio retrieval system.

features will be used to compare and find items with similar properties). The fitness value of each chromosome is determined by applying the K nearest neighbor with cosine measure as the fitness function in the genetic algorithm, the proposed approach used cosine similarity due to the fact that the Euclidean distance suffers of some issues such as: different query patterns of the same class may have a considerable different retrieval results; also the average retrieval accuracy will be poor[2]. Genetic operators such as the selection and crossover are applied, using the elitism fitness proportional selection to identify parents and prevent the fittest chromosomes from being wasted in the new population, a more common selection

approach is improved by adding an Elitism feature. Also, in this paper a new crossover method (Close Up-Feature Crossover) is proposed in order to produce a new offsprings and match the similarity to retrieve from the audio database the best results.

This paper is structured as follows: Section 2 presents the related works; Sections 3, 4, 5 and 6 cover a background to Fourier transform, feature extraction, genetic algorithm and K-Nearest Neighbor algorithm, respectively; Section 7 elaborates the proposed approach; Section 8 presents the results regarding the precision of the proposed approach and a compares it to different similarity measures; Section 9 clarifies the conclusions.

## 2. Related Work

There are many researches focusing on audio retrieval with or without using a genetic algorithm. Some of these researches include:

- G.Guo and S. Z. Li, 2003,[2] - Proposed a content-based audio classification and retrieval by using support vector machines. For audio classification they used Support vector machines and binary tree strategy while for audio retrieval they proposed distance-from-boundary (DFB). Their system finds the boundary of query audio inside which the query pattern is located. Then, sorts all the audio patterns in the database by their distances to this boundary. The boundaries are learned by using support vector machines.
- J. Olajec, R. Jarina and M. Kuba, 2006, [3] proposed Extraction Feature for Clapping Sound Detection that automatically selects subspace of feature from a feature vector of audio streams by using Genetic Algorithm method in order to improve the representativeness of the features generic audio recognition task.
- T. Virtanen and M. Hel'en, 2007, [4] proposed a probabilistic model based similarity measures for audio query-by-example. Here they proposed measures for similarity of two audio signals. Both signals are represented by using a set of features and then the models of probabilistic are estimated for the distributions feature. The similarity of the signals is measured by the ratio test of cross likelihood which uses the likelihood of the first signal being generated by the model of the second signal, and vice versa.
- S. Sundaram and S. Narayanan,2008,[5] presented an audio retrieval system by latent perceptual indexing. They used query audio retrieval by indexing audio clips in a database as points in a latent perceptual space. This is done by characterizing the clip in terms of clusters. Each audio clip of the database is mapped into a point in the latent perceptual space. The similarity measure is used to retrieve matching clips. Each of the initial reference clusters represents a specific perceptual quality in a perceptual space used in indexing the retrieval system.
- S. Bojewar and J. Fulekar, 2012 ,[6] proposed a genetic algorithm for audio search with recommender system in order to reflect the current user's intend. They used real-time genetic algorithm for recommendation method that generates new mutations by providing optimized solution every time when the algorithm ran.

## 3. Fourier Transform

The Fourier transform is used to determine how often each frequency is present in an audio signal. The Fourier transform is the most widely used tool for mapping data into a different mathematical space such as from the time domain to frequency domain, or vice versa. The transformation of the time domain to frequency domain is performed by Discrete Fourier Transform (DFT), respectively Inverse Fourier Transform (IFT) for time domain to frequency domain transformation [1, 7]). The equation for Fourier transform is [8]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

Where,  $e$ : Sinusoidal varying "basis" function for the expansion

$f(t)$ : Scale factor for the Fourier transform  $F(\omega)$ ; the original signal in the time domain.

$f(\omega)$ : The Fourier transform; strength of frequency  $\omega$  contained in  $f(t)$

(1)

The computer science deals with discrete functions so that the discrete version of the Fourier Transform is called Discrete Fourier Transform (DFT). The equation for DFT is [2]:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-i2\pi nk}{N}}, k = 0 \text{ to } N - 1$$

(2)

While the equation of Inverse Discrete Fourier transform (IDFT) is:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{i2\pi nk}{N}}, n = 0 \text{ to } N - 1$$

(3)

The interpretation of the above equations is that: The point  $N$  represents the size of the data array;  $n$  the index of time domain array;  $k$  the index of frequency domain array;  $x(n)$  the time domain data and  $X(k)$  the frequency domain data.

The time complexity of the discrete Fourier transform is  $O(n^2)$ . Many techniques were introduced to reduce the time complexity, and the most efficient algorithm is Fast Fourier Transform (FFT) Cooley-Tukey which reduces the time complexity to  $O(n \log n)$ . FFT divided the computation of DFT into two equal parts. To work the best, it needs the length of the original sequence to be a value at the power of 2 [1, 3]. The equation for FFT is [8]:

$$\begin{aligned} f_j &= \sum_{k=0}^{\frac{n}{2}-1} x_{2k} e^{-\frac{2\pi i}{n} j(2k)} + \sum_{k=0}^{\frac{n}{2}-1} x_{2k+1} e^{-\frac{2\pi i}{n} j(2k+1)} \\ &= \sum_{k=0}^{n'-1} x'_k e^{-\frac{2\pi i}{n'} jk} + e^{-\frac{2\pi i}{n} j} \sum_{k=0}^{n'-1} x''_k e^{-\frac{2\pi i}{n'} jk} \\ &= \begin{cases} f'_j + e^{\frac{2\pi i}{n} j} f''_j, & \text{if } j < n' \\ f'_{j-n'} - e^{-\frac{2\pi i}{n} (j-n')} f''_{j-n'}, & \text{if } j \geq n' \end{cases} \end{aligned}$$

(4)

The DFT is used in applications such as: image processing, signal analysis, filtering, data compression, partial differential equations and many other fields [8].

#### 4. Feature Extraction

The features of any audio signal should be extracted before the signal is retrieved. Feature extraction is the process of analysis the input of the signal. These features are usually classified into two types: perceptual and physical [9].

Perceptual features are aspects characteristic to human hearing, such as: timbre, pitch and rhythm. To extract physical features from sound signals is easy, as they are based on mathematical and statistical properties. Physical features include: fundamental frequency (f0), Zero Crossing Rate (ZCR) and Energy. Some perceptual features are related to physical features: timbre is related to the spectral content feature and pitch is related to f0 feature [9].

To extract features from an audio signal, the signal is divided at first into short-terms of signal called frames or windows, and then a set of features is computed per frame. This type of processing provides a sequence,  $\mathbf{F}$ , of feature vectors per audio signal [10].

#### 4.1. Audio Signal Features Overview

##### 4.1.1. Energy

Energy measures the amount of signal in any unit of time, being used to locate silence. The energy is calculated using equation [10]:

$$E(i) = \sum_{n=1}^{W_L} |x_i(n)|^2 \quad (5)$$

Where  $x_i(n)$ ,  $n = 1, \dots, W_L$  represents the sequence of audio samples of the  $i$ th frame,

And  $W_L$  represents the length of the frame.

##### 4.1.2. Zero Crossing Rate (ZCR)

The zero-crossing rate (ZCR) of an audio frame is the measure of often the signal is crossing zero in a unit of time (it is the number of times the signal changes its value). The ZCR is a time time-domain feature [9] fast to calculate using the following equation [10]:

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} |sgn[x_i(n)] - sgn[x_i(n-1)]|$$

where  $sgn(\cdot)$  is:

$$sgn[x_i(n)] = f(x) = \begin{cases} 1, & x_i(n) \geq 0 \\ -1, & x_i(n) < 0 \end{cases} \quad (6)$$

##### 4.1.3. Spectral Centroid and Spread

The spectral centroid and the spectral spread are two simple measures of spectral brightness [11]. The equation for the spectral centroid  $C_i$ , of the  $i$ th audio frame is [10, 11]:

$$C_i = \frac{\sum_{k=1}^{W_{fL}} kX_i(k)}{\sum_{k=1}^{W_{fL}} X_i(k)} \quad (7)$$

While the equation for the spectral spread is:

$$S_i = \sqrt{\frac{\sum_{k=1}^{W_{fL}} (k - C_i)^2 X_i(k)}{\sum_{k=1}^{W_{fL}} X_i(k)}} \quad (8)$$

##### 3.1.4. Spectral Flux

Spectral flux is the measure of the spectral change between two successive frames. The equation for the spectral flux is [10, 11]:

$$Fl_{(i,i-1)} = \sum_{k=1}^{W_{fL}} (EN_i(k) - EN_{i-1}(k))^2$$

Where  $EN_i(k)$  represents the  $k$ th normalized DFT coefficient at the  $i$ th frame.

$$EN_i(k) = \frac{X_i(k)}{\sum_{l=1}^{W_{fL}} X_i(l)} \quad (9)$$

#### 4.1.5. Spectral Rolloff

Spectral rolloff is the measure of the spectral shape of an audio signal. The equation for the spectral rolloff is [10, 11]:

$$\sum_{k=1}^m X_i(i) = C \sum_{k=1}^{Wf_L} X_i(k)$$

where  $C$  is the certain percentage (user parameter).

(10)

#### 4.1.6. MFCCs

Mel-Frequency Cepstrum Coefficients (MFCCs) are important features in several audio analysis applications such as in the field of speech processing, speaker clustering, musical classification and in many others.

To extract MFCCs If  $\tilde{O}_k$ ,  $k = 1, \dots, L$  is the power at the output of the  $k$ th mel-scale filter, then the equation for the MFCCs is [5]:

$$C_m = \sum_{k=1}^L (\log \tilde{O}_k) \cos \left[ m \left( k - \frac{1}{2} \right) \frac{\pi}{L} \right], m = 1, \dots, L$$

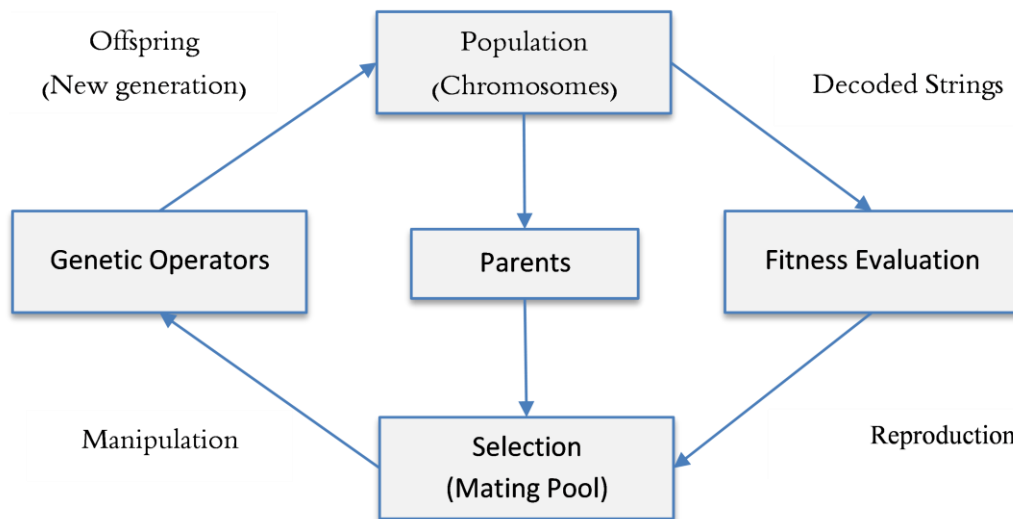
(11)

#### 4.1.7 Chroma Vector

The Chroma vector consists in twelve different [pitch classes](#) (12-element representation). The Chroma vector feature is widely used in speech-music distinguishing applications. The equation is [10]:

$$v_k = \sum_{n \in S_k} \frac{X_i(n)}{N_k}, \quad k \in 0, \dots, 11$$

(12)



**Figure 2: The cycle of GA**

The interpretation of the previous equations is that: the  $S_k$  represents a subset of the frequencies that correspond to the DFT coefficients,  $N_k$  is the cardinality of  $S_k$ ,  $V$  is a matrix of the sequence of Chroma vectors.

## 5. Genetic Algorithm

The genetic algorithm (GA) is an algorithm that has the ability to solve a multitude of problems in a numerous problem domains and is a very powerful optimization tool [12].

The basic steps of the cycle of a genetic algorithm are [8, 9]:

1. Generate a set of solutions randomly. These solutions are called a population;
2. Evaluate each solution based on the determined fitness values;
3. Create a mating pool by selecting the best solutions depending on the fitness;
4. Use genetic manipulation (crossover and mutation) to generate a new population of strings.

Each cycle in the GA results in a new generation of possible solutions and the process is terminated when the solution to the given problem is found or when the condition is satisfied [12, 13]. Figure 2 shows the cycle of a genetic algorithm (GA) [13].

The following sections discuss the need for each of these steps [13, 14]:

**Population** – consists in the initial solutions selected from randomly chosen input values. There is no recommended size of a population, instead it varies according to type of the problem.

**Fitness** – the measure used to compare multiple solutions to determine which is better.

**Selection** – the genetic algorithm operates over a number of generations, so that the purpose of selection is to choose the fittest solutions of the population that will create offsprings for the new generation, commonly known as mating pool.

**Crossover** – the first step of the genetic manipulation process. It recombines the two selected solutions (parent's chromosomes) to produce new solutions (offspring) for the next generation. The idea of the crossover is that the new solutions may be better than both of the parents.

**Mutation** – is the second step of the genetic manipulation process, which randomly selects a small portion of the solutions (one or more bit positions of the chromosomes) and changes it (e.g., a bit of 0 to 1 or 1 to 0). The mutation can overcome trapping at local maxima.

## 6. K-Nearest Neighbor

K-Nearest Neighbor (K-NN) is a machine learning algorithm, which is often used for classification, retrieval, estimation and prediction. In this algorithm, the training data set is stored and the test set is classified by finding the closest pattern in the training data set that is called nearest neighbor [12, 15].

The KNN is working by taking the nearest k neighbors and using the majority rule to vote the correct class. The closest neighbors are determined by calculating the distance of the test set to each pattern in the training set, which is performed in some methods, the most popular one using Euclidean Distance [12]. The equation of Euclidean distances for patterns  $p$  and  $q$  is [11, 16]:

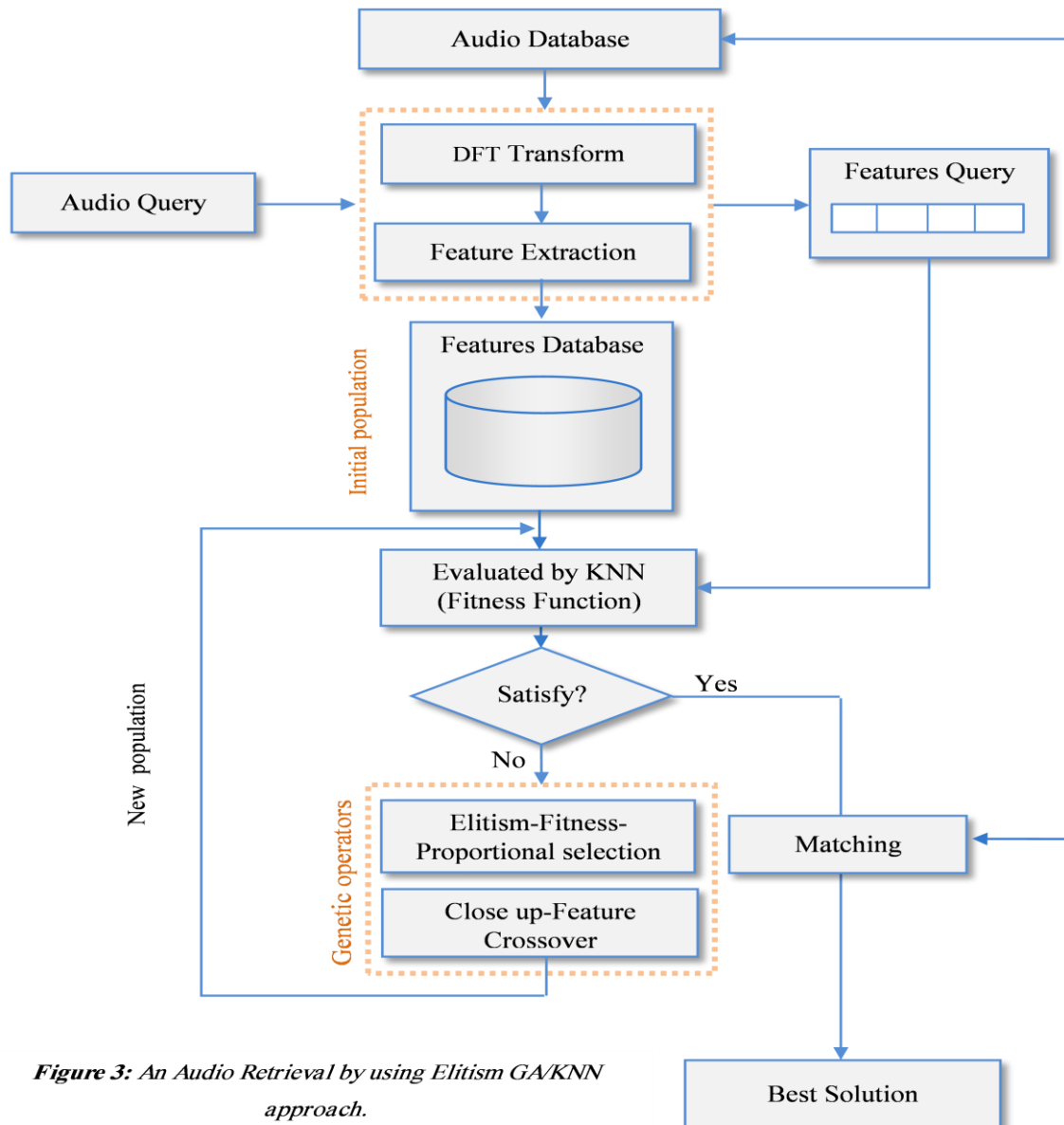
$$d = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

(13)



### 7. The Proposed Elitism GA-KNN Approach

The proposed approach uses as the optimization of the Audio Retrieval System (ARS) a mathematical algorithm equivalent to Biological Genetics. Figure 3 illustrates the procedure of the proposed approach.



**Figure 3:** An Audio Retrieval by using Elitism GA/KNN approach.



The methodology elaborated here, uses Elitism Genetic Algorithm with KNN classifier as the fitness function to retrieve audio features of an audio files that consists in the following steps:

**1. Represent the signals by applying the Discrete Fourier Transform (DFT)** in order to acquire a spectral-domain (frequency) of the signal.

**2. Feature extraction** – is a fundamental processing step in Audio Retrieval Systems, where the feature vectors are generated for Query and audio Database. So, each audio file (chromosome) will be represented as a vector of seven features. The extracted features include: **Energy, Zero Crossing Rate (ZCR), Spectral Centroid and Spread, Spectral Flux, Spectral Rolloff, MFCCs and Chroma Vector**. The outputs of this step are Query Features Vector and Audio Features Database. Figure (4) exemplifies the feature vector of an audio file. The equations of calculating those features are described in section 4.

Energy	ZCR	Spectral Centroid & Spread	Spectral Flux	Spectral Rolloff	MFCCs	Chroma Vector
--------	-----	----------------------------	---------------	------------------	-------	---------------

**Figure 4:** Features Vector of an audio file.

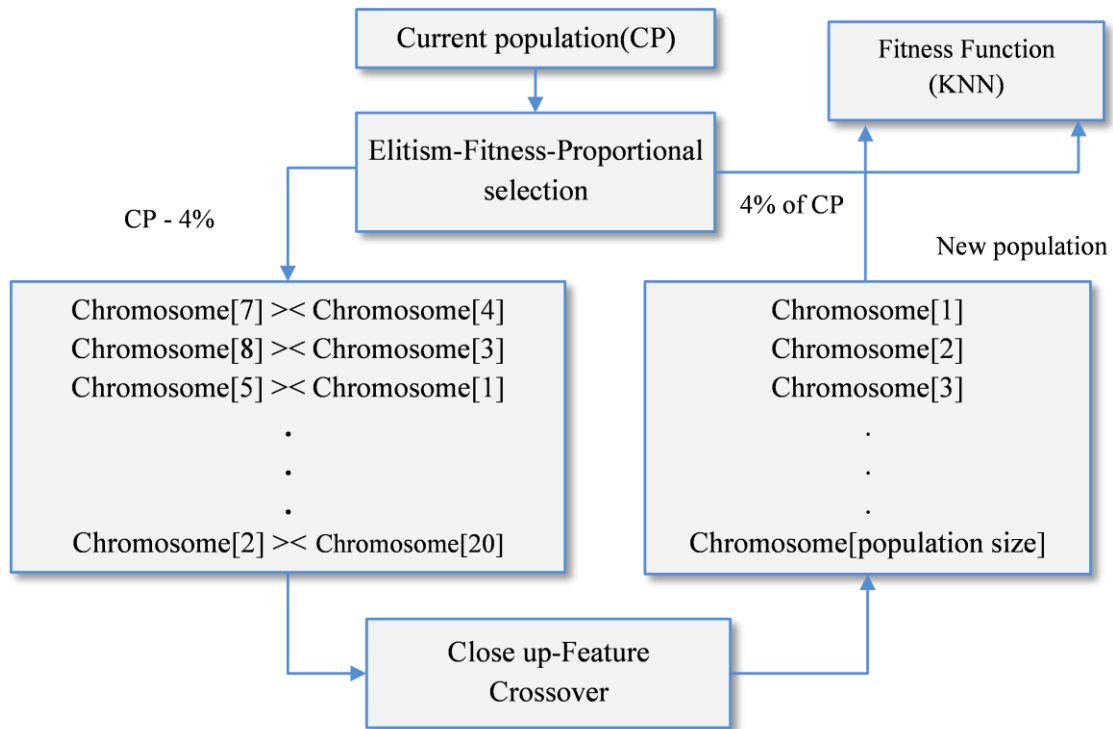
**3. Initialize a population** – Construct the initial population from all Audio Features Database.

**4. Chromosomes representation** – By consuming a Float form, the audio files (chromosomes) are represented as a vectors of data without any adjustment, each vector containing seven values of audio features.

**5. K-Nearest Neighbor (KNN)** – Find the fitness value of each chromosome by applying the K—Nearest Neighbor as the fitness function in Genetic Algorithm. For each chromosome, the KNN- Fitness function is as follows:

**Algorithm K-NN Fitness:**  
**Input:** Query Feature Vector (QFV),  
 Audio Features Database (AFD)  
**Output:** K neighbors of QFV  
**Begin**  
**Step1:** Determine the count, K of nearest neighbor;  
**Step2:** Compute the cosine of angle between QFV and each vector in AFD by using the Cosine distance, Dc (QFV, AFD) = 
$$\frac{\sum_{i=1}^n QFV(i).AFD(i)}{\sqrt{\sum_{i=1}^n QFV(i)^2 \cdot \sum_{i=1}^n AFD(i)^2}}$$
  
**Step3:** Rank the fitness value depending on the minimum distance;  
**Step4:** Find k nearest neighbors for QFV.  
**End.**

**6. Satisfy** – check the end condition. If the genetic iteration count is less than the population size (Audio Files Database) then continue, else go to the Matching step to return the best solution.



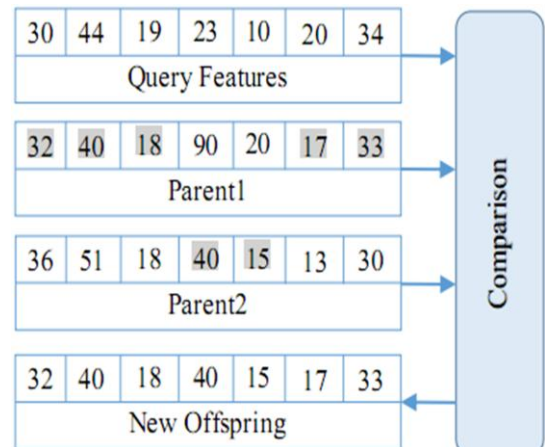
*Figure 5: The proposed Genetic Operator.*

**7. Genetic Operators** – construct a new population by applying the genetic operators such as the selection and crossover, but there is a big chance to lose the fittest chromosomes in the next generation (i.e. the chromosomes in offspring will be weaker than their parents). To improve the performance of GA, our approach used a feature called **Elitism**. The basic idea is to transfer a small portion of the best chromosomes, without altering the next generation. Figure 5 presents the proposed Genetic Operators.

**7.1. Elitism-Fitness-Proportional selection** – to prevent the fittest chromosomes from being wasted in the new population, a more common selection approach is improved by adding an Elitism feature. The probability of selection is proportional to an individual's fitness. The pair's selection is continued until reaching the assumed population size. Such approach is known as Fitness-Proportional selection, using an algorithm that is outlined in below:

```

Algorithm Elitism-Fitness-Proportional selection:
Input:  $P_{size}$  (K neighbors of QFV)
Output: Pair of Parents
Begin
  Step1: Set 4% from the size of
    population  $P_{size}$  to elite
    count, copy them to the
next
  generation;
  Step2: Find the summation of all
    The population individuals
     $P_{size} - \text{Elite Count}$ ).
    Call this  $P_{sum}$ ;
  Step3: For index = 1 : 2
    - Pick an arbitrary
    number,
      N, between 0 and  $P_{sum}$ ;
    - Add the fitness value
    of
      the population
      individuals together
      (one at a time)
      halting quickly when
    the
      new sum is greater than
      an arbitrary number N.
      The last remaining
      individual is the
    chosen
      individual.
    End For
End.
  
```



**Figure 6: Close Up-Feature Crossover**

The Close Up-Feature Crossover Algorithm runs a comparison between two chromosomes (parents) depending of the query features vector. After crossover, one chromosome (offspring) will be produced by taking the close up features to the Query Feature Vector, where the number of pairs is equal to population size. The close up-feature crossover is delineated by the algorithm found on the next page.

## 7.2. The Proposed Close Up-Feature Crossover

Performs the proposed crossover between each two chromosomes (parents) to generate a new chromosome (offspring). Figure 6 shows the proposed crossover.

8. **Matching** – this step determines the similarity between the last population and audio features database by using the cosine measure. Once a similarity is found, those items are retrieved from the audio database and considered as best results.

```
Algorithm Close Up-Feature  
Crossover:  
Input:  Query Feature Vector  
(QFV),  
        Psize = Psize - Elite Count  
Output: offsprings  
Begin  
  Step1: Set index to 1;  
  Step2: Call Elitism-Fitness  
        Proportional selection,  
        select a pair of  
        chromosomes as FirstParent  
        and SecondParent;  
  Step3: Offspring[index] =  
        Compare_Exchange(FirstParent  
        , SecondParent);  
  Step4: Index = index + 1;  
  Step5: If (index <= Psize) then  
        goto step2;  
End.  
  
Function [offspring] =  
Compare_Exchange(parent1, parent2)  
  
For i = 1 : 7  
  if (|QFV[i] - parent1[i]|  
    <= |QFV[i] - parent2[i]|)  
  then  
    Offspring[i] = parent1[i]  
  else  
    offspring[i] = parent2[i]  
  End For  
End Compare_Exchange
```

```
Algorithm Close Up-Feature Crossover:  
Input:  Query Feature Vector (QFV),  
        Psize = Psize - Elite Count  
Output: offsprings  
Begin  
  Step1: Set index to 1;  
  Step2: Call Elitism-Fitness  
        Proportional selection,  
        select a pair of  
        chromosomes as FirstParent  
        and SecondParent;  
  Step3: Offspring[index] =  
        Compare_Exchange(FirstParent  
        , SecondParent);  
  Step4: Index = index + 1;  
  Step5: If (index <= Psize) then  
        goto step2;  
End.  
  
Function [offspring] =  
Compare_Exchange(parent1, parent2)  
  
For i = 1 : 7  
  if (|QFV[i] - parent1[i]|  
    <= |QFV[i] - parent2[i]|)  
  then  
    Offspring[i] = parent1[i]  
  else  
    offspring[i] = parent2[i]  
  End For  
End Compare_Exchange
```

8. **Matching** – this step determines the similarity between the last population and audio features database by using the cosine measure. Once a similarity is found, those items are retrieved from the audio database and considered as best results.

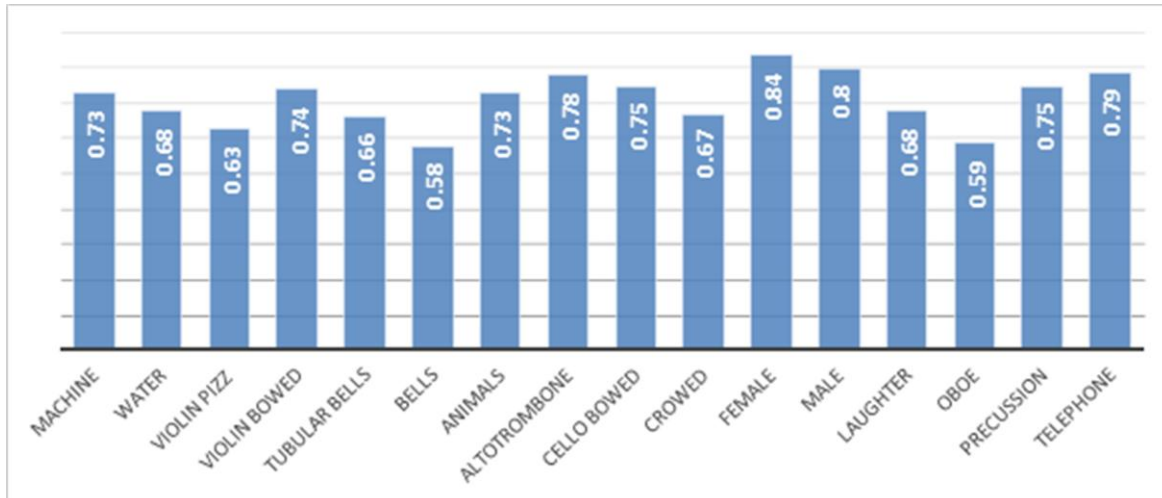
## 8. Experimental Results

In this experiment we used Muscle Fish, an audio database with audio samples for experimental purposes. It consists of 409 sound files categorized in 16 classes. The audio classes are named as following: alto trombone, animals, bells, cello bowed, crowds, female, laughter, machines, male, oboe, percussion, telephone, tubular bells, violin bowed, violin pizz and water. Each audio file has a duration ranging between 1 to 10 seconds. Muscle Fish database can be obtained from [17] and it is used in [2] and [19].

One signal is randomly chosen from database and considered as a query, while the rest are treated as database samples. Seven features are extracted from the audio database and the query, with their names mentioned in figure 4. The initial population is constructed, with the size of all database samples minus one (400 - 1) is constructed. The fitness value of each chromosome is determined by applying the K nearest neighbor when  $k = 20$ . Figure 7 and Table 1 present the GA-KNN Precision Results of 16 classes for top 10 retrieved signals by using the Cosine measure. Precision is a measure that shows how much of the results look as relevant for the user. The resulting value is a real number between 0 and 1, which is defined as [13]:

$$\text{Precision (P)} = \frac{\text{relevant items retrieved}}{\text{relevant items}}$$

Furthermore, we compared the precision of retrieval with other similarity measures such as: City block, Jaccard, Dice and Cosine. The Cosine similarity measure gives the best average retrieval, where the average precision for 16 classes was 0.71475. Table 1 and Figure 8 show the Retrieval Precision Comparison Results for Elitism GA-KNN Approach.



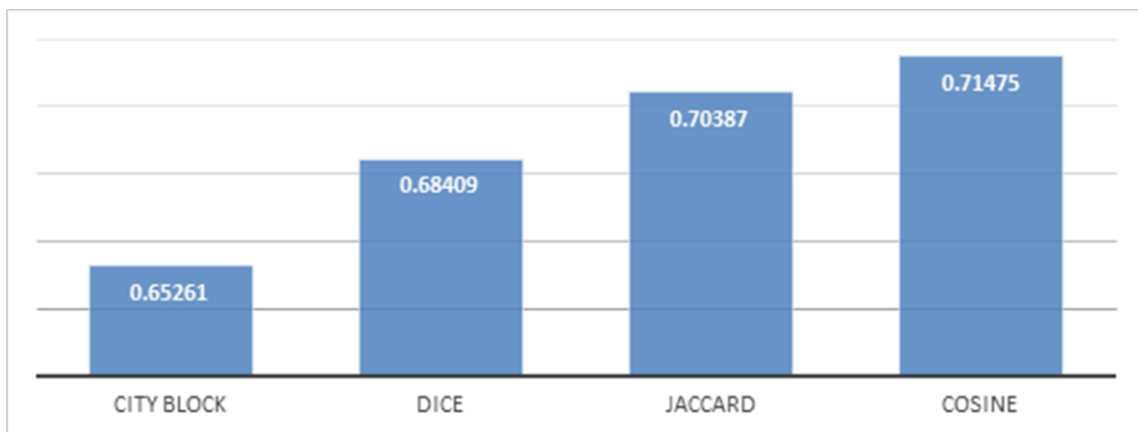
**Figure 7: The Chart of Retrieval Precision Results for Elitism GA-KNN Approach by using Cosine measure**

Class Name		Precision
1	Machine	0.73
2	Water	0.68
3	Violin Pizz	0.63
4	Violin Bowed	0.74
5	Tubular Bells	0.66
6	Bells	0.58
7	Animals	0.73
8	Alto Trombone	0.78
9	Cello Bowed	0.75
10	Crowed	0.67
11	Female	0.84
12	Male	0.8
13	Laughter	0.68
14	Oboe	0.59
15	Percussion	0.75
16	Telephone	0.79

**Table(1):** The Retrieval Precision Results for Elitism GA-KNN Approach by using Cosine measure

Similarity Measures			Average Precision
1	City block	$d_{CB} = \sum_{i=1}^d  P_i - Q_i $	0.65261
2	Dice	$S_{Dice} = \frac{2 \sum_{i=1}^d P_i Q_i}{\sum_{i=1}^d P_i^2 + \sum_{i=1}^d Q_i^2}$	0.68409
3	Jaccard	$S_{Jac} = \frac{\sum_{i=1}^d P_i Q_i}{\sum_{i=1}^d P_i^2 + \sum_{i=1}^d Q_i^2 - \sum_{i=1}^d P_i Q_i}$	0.70387
4	Cosine	$S_{Cos} = \frac{\sum_{i=1}^d P_i Q_i}{\sqrt{\sum_{i=1}^d P_i^2} \sqrt{\sum_{i=1}^d Q_i^2}}$	0.71475

**Table(2):** Retrieval Precision Comparison of using Elitism GA-KNN Approach with different measures



**Figure 8:** The Chart of Retrieval Precision Comparison Precision Results for Elitism GA-KNN Approach among different similarity measures.

## 9. Conclusion

This paper presented a new approach for the retrieval and classification of audio files. This approach fused two algorithms: genetic algorithm and KNN, which hybridizes the optimization and search power of the genetic algorithm with the classification power of the KNN algorithm. The resulted algorithm requires computational capabilities higher than KNN algorithm, but accomplishes higher classification/retrieval performance and accuracy in a reasonable time.

In KNN, our approach used cosine similarity due to the fact that the Euclidean distance suffers of some issues such as: different query patterns of the same class may have a considerable different retrieval results; also the average retrieval accuracy will be poor. Consequently, it has been established as a common selection method in elitism and a proposed Close Up-Feature Crossover for this work. Furthermore, it gave us the best results after compared with City block, Jaccard and Dice.

## References

1. H. Shatkay, "*The Fourier Transform\_A Primer*", Brown University, November 1995.
2. G. Guo and Stan Z. Li, "*Content-Based Audio Classification and Retrieval by Support Vector Machines*", IEEE Transactions on Neural Networks, Vol. 14, No. 1, January 2003.
3. J. Olajec, R. Jarina and M. Kuba, "*GA-Based Feature Extraction for Clapping Sound Detection*", IEEE Transactions on Neural Networks, 25-27, 2006.
4. T. Virtanen and M. Hel'en, "*Probabilistic Model Based Similarity Measures For Audio Query-By-Example*", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, October 21-24, 2007.
5. S. Sundaram and S. Narayanan, "*Audio Retrieval By Latent Perceptual Indexing*", IEEE, 2008.
6. S. Bojewar and J. Fulekar, "*Application Of Genetic Algorithm For Audio Search With Recommender System*", International Journal of Advanced Computer and Mathematical Sciences , Vol 3, Issue 2, pp 224-226, 2012.
7. N. Dahnoun , "*Theory and Implementation*", Bristol University, Texas Instruments, 2004.
8. N.Nincic, "*Fourier Transform and Applicatio ns*",[http://www.efunda.com/math/fourier\\_transform](http://www.efunda.com/math/fourier_transform).
9. D. Gerhard, "*Audio Signal Classification: History and Current Techniques*", University of Regina, ISSN 0828-3494, November 2003.
10. T. Giannakopoulos and A. Pikrakis ,"*Introduction to audio analysis: A Matlab Approach*", Academic Press is an imprint of Elsevier, 2014.
11. G. Tzanetakis, G. Essl and P. Cook ,"*Automatic Musical Genre Classification of Audio Signals*", <http://www.cs.princeton.edu/~gtzan>.
12. M. Tim Jones, "*Artificial Intelligence: A Systems Approach*", Infinity Science Press LLC, 2008.



13. A. Konar, "Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain", Jadavpur University, CRC Press LLC, 2000.
14. T. Munakata, "Fundamentals of the New Artificial Intelligence", Second Edition, Springer, 2008.
15. D. T. Larose, "Discovering Knowledge in Data: An Introduction to Data Mining", ISBN 0-471-66657-2, 2005.
16. A. G. Karegowda and Kishore B., "Enhancing Performance of KNN Classifier by Means of Genetic Algorithm and Particle Swarm Optimization", IJAFRC, Volume 1, Issue 5, ISSN 2348 — 4853, May 2014.
17. <http://www.musclefish.com/cbrdemo.html>.
18. Ch. D. Manning and P. Raghavan, "An Introduction to Information Retrieval", Cambridge, England, 2009.
19. S. Z. Li, "Content-based classification and retrieval of audio using the nearest feature line method", IEEE Trans. Speech Audio Processing, vol. 8, pp. 619–625, Sept. 2000.

### استرجاع الصوت بالاعتماد على المحتويات باستخدام أسلوب النخبة GA-KNN

زينة حسين تومان  
جامعة القادسية  
كلية علوم الحاسبات وتكنولوجيا المعلومات  
قسم علوم الحاسوب  
zinah.hussein@qu.edu.iq

منى غازي عبد الصاحب  
الجامعة التكنولوجية  
قسم علوم الحاسوب  
110096@uotechnology.edu.iq

سارة حسين تومان  
جامعة القادسية  
كلية الهندسة  
قسم هندسة الطرق والنقل  
Sarah.toman@qu.edu.iq

#### المستخلص :

الصوت الرقمي شائعة جدا ومهمة في تجربة مستخدم الكمبيوتر. تتطلب الكميات المتزايدة من البيانات الصوتية تحسين طرق وخوارزميات جديدة لمعالجة هذه المعلومات. الطريقة المقترحة تستخدم طريقة GA-KNN ( الخوارزمية الجينية مع الجار K الأقرب كدالة الافضل ) لاسترجاع محتوى الملفات الصوتية. المدخل هو الملف الصوتي (الاستعلام) والمخرجات هي قائمة من الملفات الصوتية التي تكون متدرجة على اساس تشابهها. بالبداية، النظام يستخلص الخصائص ذات الصلة من قاعدة بيانات واستعلام الصوت. ثم يعتبر نمط الاستعلام كحد للمقارنة. انشاء العناصر الأولية من قاعدة بيانات جميع خصائص الصوت. ومع ذلك، فإن مقياس التشابه الذي أساسها المسافة الإقليدية بين أنماط الصوت إلى الاستعلام يعاني من مشاكل: نتائج الاسترجاع المقابلة لأنماط الاستعلام المختلفة داخل نفس الفئة قد تكون تختلف كثيرا وسوف يكون متوسط دقة الاسترجاع منخفضة، لذلك في هذا البحث يستخدم قياس جيب التمام في KNN. تم تحسين طريقة الاختيار لمنع الكروموسومات الافضل من ان تفقد في العناصر الجديدة، وذلك بإضافة ميزة النخبوية، ٤٪ النخبوية. اقترح طريقة تقاطع جديدة (تقاطع الخاصية الاقرب) بواسطة المقارنة بين انماط الصوت ونمط الاستعلام. وأخيرا تم تقييم اسلوبنا باستخدام قاعدة بيانات شائعة للصوت تتألف من ٤٠٠ صوت و ١٦ صنف وكانت دقة الاسترجاع هي 0.71475.

**الكلمات المفتاحية :** نظام استرجاع الصوت، استخلاص الخصائص، الخوارزمية الجينية (GA)، خوارزمية K من الجار الاقرب (KNN)، النخبة الافضل، تقاطع الخاصية الاقرب.