FEATURE-WISE INDEXING AND QUERY-BY-HUMMING FOR EFFICIENT RETRIEVAL OF INDIAN LIGHT MUSIC

Gopala[1], Nagappa U Bhajantri[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Government Engineering College, Huvina Hadagali, Karnataka,583219 India.

[2]Professor, Department of Computer Science and Engineering, Government Engineering College, K.R. Pete, Karnataka,571426 India.

Abstract - *This paper proposes a content-based music information retrieval (CBMIR) system for Indian light music, allowing users to retrieve a ranked list of songs based on melody features. The system enables query-by-humming (QbH), where the user's hummed melody is compared with indexed song melodies. Chroma features, representing the pitch content of music, are extracted and indexed using the hierarchical data format version 5 (HDF5) format for efficient storage and retrieval. Fast dynamic time warping (FastDTW) is utilized to compare melody features, ensuring rapid and accurate similarity matching. The HDF5 indexing allows for scalable retrieval even with large music datasets, while FastDTW reduces computational time without compromising accuracy. This system offers an intuitive solution for users to find songs based on melody, even if they only recall the tune.*

Keywords—Chroma features, content-based music information retrieval, fast dynamic time warping, HDF5 indexing, Indian light music, query-by-humming

I. INTRODUCTION

Music Information Retrieval (MIR) has gained significant attention in recent years [2], with various techniques being developed to enable efficient search and retrieval of music based on audio content [3][4][7]. Traditional text-based searches, such as those relying on song titles or lyrics, may not always be effective, especially when users only recall the melody of a song [8]. To address this limitation, Content-Based Music Information Retrieval (CBMIR) has emerged as a promising approach that enables users to search for songs using musical features extracted from audio signals [1][5][6].

Indian light music, which encompasses a diverse range of melodies, presents unique challenges for CBMIR systems due to variations in tempo, pitch modulations, and expressive ornamentation. A robust retrieval system must be capable of capturing these melodic characteristics and efficiently comparing them with a database of songs.

This paper proposes a CBMIR system specifically designed for Indian light music, allowing users to search for songs through Query-by-Humming (QbH) [1][5][6]. The system extracts chroma features, which effectively represent the harmonic and melodic content of a song [6], and indexes them using the HDF5 format for efficient storage and retrieval [9][10]. To achieve fast and accurate similarity matching, the system employs FastDTW [11][12][13][14], which aligns the hummed query with pre-indexed melodies while minimizing computational complexity [4].

The key contributions of this work include: Feature Extraction Using Chroma Representation – Capturing the melodic structure of Indian light music for accurate retrieval [10], [11]. HDF5-Based Feature Indexing – Efficient storage and retrieval of large-scale audio datasets. FastDTW for Similarity Matching – Improving query response time while maintaining high retrieval accuracy [8]. Scalable and User-Friendly QbH System – Allowing users to find songs based on melody without requiring textual metadata.

The remainder of this paper is structured as follows: Section II discusses related work in CBMIR and QbH. Section III describes the proposed methodology, including feature extraction, indexing, and similarity computation. Section IV presents experimental results and performance evaluation. Finally, Section V concludes the paper and suggests future research directions.

II. RELATED WORK

Content-based music information retrieval has been widely explored in recent years, with various techniques developed for efficient music search and retrieval [15][2]. QbH is one of the most intuitive approaches, allowing users to retrieve songs based on melody rather than metadata such as lyrics or artist names [1]. Several studies have focused on feature extraction, indexing methods, and similarity matching techniques to improve the accuracy and efficiency of QbH systems [16][17].

A. Melody Representation and Feature Extraction

Melody is a crucial component in music retrieval systems, and various feature extraction techniques have been proposed to capture its characteristics. Chroma features have been extensively used to represent the harmonic and melodic content of a song, as they provide a compact and robust representation of pitch information [18]. Researchers have also explored pitch contour extraction and mel-frequency cepstral coefficients (MFCCs) for capturing melodic structures [19][20]. However, chroma-based representations are more effective for melody-based retrieval, especially in Indian light music, where pitch variations and ornamentations are prevalent [18].

B. Indexing Methods for Efficient Retrieval

Efficient indexing is essential for handling large-scale music datasets. Traditional indexing techniques, such as k-d trees and hash-based methods, have been applied in MIR systems [15]. More recently, HDF5-based indexing has gained popularity due to its ability to store large feature datasets while enabling fast retrieval [9][10]. HDF5 provides an efficient hierarchical storage format that optimizes access times for high-dimensional feature vectors, making it suitable for real-time music retrieval applications [21]. This study proposes a QbH method that constructs an index of melodic fragments by extracting pitch vectors and efficiently searches for similar fragments using locality sensitive hashing (LSH). The method achieved a mean reciprocal rank of 0.885 for 2,797 queries when searching a database of 6,030 MIDI melodies [7].

C. Similarity Matching Techniques

To compare a user’s hummed query with stored melodies, various similarity measurement techniques have been proposed. DTW is a widely used method for aligning time-series data, allowing for variations in tempo and timing [11][13]. Traditional DTW, however, has high computational complexity, making it less suitable for large datasets. Recent advancements in FastDTW have significantly reduced computational overhead while maintaining high accuracy, making it an ideal choice for QbH systems [12]. Other techniques, such as hidden Markov models (HMMs) and convolutional neural networks (CNNs), have been explored for melody matching, but they often require extensive training data and computational resources [22].

D. Content-Based Retrieval for Indian Light Music

Most existing CBMIR systems focus on Western music, where structured chord progressions and fixed scales simplify melody matching [15]. However, Indian light music presents unique challenges due to its diverse scales (ragas), ornamentations (gamakas), and non-uniform rhythmic patterns [23]. Prior research on Indian classical and light music retrieval has explored pitch-based representations, but few studies have specifically optimized QbH systems for this genre [19].

III. PROPOSED METHODOLOGY

This section describes the methodology used for the CBMIR system, focusing on QbH for Indian light music. The system consists of four main stages: audio feature extraction, feature indexing, query processing, and similarity matching using FastDTW. The overall workflow is illustrated in Figure 1.

A. Audio Feature Extraction Using Chroma Features

Melody is the most distinguishing characteristic in music retrieval systems, and capturing it effectively is essential for accurate query matching [18]. This system extracts chroma features, which represent the distribution of pitch classes over time and are well-suited for harmonic and melodic content analysis [24][11]. During preprocessing, input songs are standardized by converting them to a uniform sampling rate (22.05 kHz) and mono format for consistency [4]. Additionally, noise reduction and normalization techniques are applied to enhance signal quality and reduce variability in recordings [26]. For chroma feature extraction, the Librosa library is employed to compute chroma short-time Fourier transform (STFT) features [27], which are calculated in overlapping frames to provide a fine-grained representation of melodic variations. Chroma feature extraction plays a vital role in music and audio analysis, particularly in applications such as Music Information Retrieval (MIR) and Query-by-Humming (QbH) [11]. The process involves a series of mathematical transformations to convert audio signals into chroma vectors, enabling accurate melodic analysis and retrieval in content-based systems [18][11].

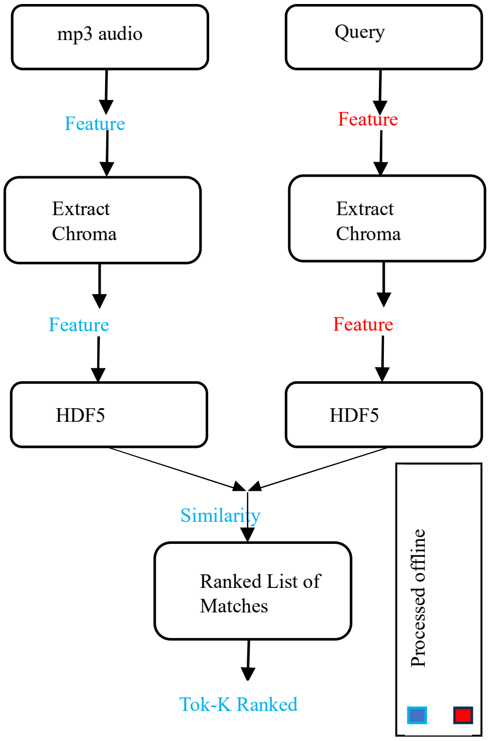

Figure 1. System

B. Fourier Transform and Spectrogram Generation

To begin, we need to transform the raw audio signal into a frequency domain representation. This allows us to analyze the harmonic content of the signal, which is essential for identifying pitch classes [11]. The STFT is applied to the audio signal to divide it into overlapping frames. For each frame, we compute the Fourier Transform to obtain the frequency components. The STFT *X (t, f)* is given by:

$$X(t,f) = \int_{-\infty}^{\infty} x(t) \cdot w(t-t') \cdot e^{-j2\pi f t'} dt' \qquad (1)$$

Where *x(t)* is the original audio signal. *w(t−t')* is a window function (Hamming or Hanning window) applied to ensure smooth transitions between frames [28]. *f* represents frequency, and *t* represents time [29]. The result is a time-frequency representation where *X (t, f)* gives the amplitude of the frequency *f* at time t [30].

The calculation of chroma features depends on several key parameters that impact accuracy and resolution. The window size (n_fft) determines the number of samples analyzed per STFT frame, with a default value of 2048 samples, balancing frequency resolution and temporal precision to effectively capture harmonic content [31]. The window function reduces spectral leakage, and the Hamming window is commonly used for its smooth roll-off and minimal side lobes, ensuring clean frequency representation [26]. The overlap (hop_length) controls how much consecutive STFT frames overlap, with a default of 512 samples (75% overlap for a 2048 sample window), allowing chroma features to be computed approximately every 23 milliseconds at a 22.05 kHz sampling rate. This high overlap ensures a smooth transition between frames and effectively captures melody variations. The overlap between consecutive windows is calculated as

$$\text{Overlap} = n\_fft - \text{hop\_length} = 2048 - 512 = 1536\,\text{samples} \qquad (2)$$

Together, these parameters optimize the extraction of chroma features, allowing the system to capture pitch-class distributions across time with a balance of frequency and temporal resolution[11]. This is essential for accurately representing the harmonic and melodic elements in music, particularly for tasks like QbH. We typically use the magnitude spectrum (ignoring the phase information) to focus on the energy in each frequency bin. The magnitude of the STFT at time *t* and frequency bin *f* is:

$$M(t, f) = |X(t, f)| \qquad (3)$$

This represents the intensity of the signal at a particular frequency at time t [27].

C. Mapping to Pitch Classes

After obtaining the frequency-domain representation through the magnitude spectrum, the next step is to **map the frequency bins to pitch classes**—the 12 notes of the chromatic scale. The chromatic scale consists of the following pitch classes C, C#, D, D#, E, F, F#, G, G#, A, A#, B. To perform this mapping, we determine which frequency bin corresponds to which pitch class. For example, frequencies between $C4$ ($261.63\,Hz$) and $C\#4$ ($277.18\,Hz$) are assigned to pitch class C, while those between $C\#4$ and $D4$ ($293.66\,Hz$) belong to pitch class $C\#$. This process groups spectral energy into perceptually relevant categories, allowing us to abstract melodic content across octaves [32].

We adopt the **Equal Temperament Scale**, the standard tuning system in Western music, in which each octave is divided into 12 equal semitone steps. Under this tuning system, the pitch class of a given frequency f is computed as:

$$Pitch\ Class\ Index = \left\lfloor 12 \cdot log_2\left(\frac{f}{f_0}\right)\right\rfloor mod\ 12 \qquad (4)$$

where $f_0 = 440\,\text{Hz}$ is the reference frequency (*A4* in the standard tuning), *f* is the frequency of interest, the logarithmic term determines the number of semitones between *f* and f_0 and the result is then taken modulo 12 to assign the frequency to one of the 12 pitch classes [3].

Once frequency bins are mapped to their respective pitch classes, we aggregate the magnitude spectrum values across all frequency bins belonging to each class. Let $M(t, f)$denote the magnitude of the STFT at time t and frequency f. The total energy $P_i(t)$ for pitch class i at time t is calculated as:

$$P_i(t) = \sum_{f \in \text{Pitch Class } i} M(t, f) \qquad (5)$$

This results in a 12-dimensional chroma vector $C(t)$ for each time frame t :

$$C(t) = [P_1(t), P_2(t), \ldots, P_{12}(t)] \qquad (6)$$

here, *C(t)* is the chroma feature vector for the frame at time t and $P_i(t)$ is the energy of pitch class *i* at time *t*.

This representation is **invariant to octave** and robust to timbral variations, making it particularly suitable for tasks like Query-by-Humming (QbH) and music retrieval [33][34]. Repeating the process across all frames yields a chroma feature matrix $C \in R^{12 \times N}$, where N is the number of time frames:

$$C = [C_1, C_2, \ldots, C_{12}] \qquad (7)$$

Where each C_i is a 12-dimensional chroma feature vector for frame *i*.

To better illustrate the effectiveness of chroma-based features, we used t-distributed Stochastic Neighbor Embedding (t-SNE) to project high-dimensional chroma features into a 2D space for visualization [35]. Figure 2 displays this projection for both the main audio and the query humming, highlighting the clustering of semantically similar melodic content. This confirms the discriminative power of chroma features for melodic similarity in QbH systems.

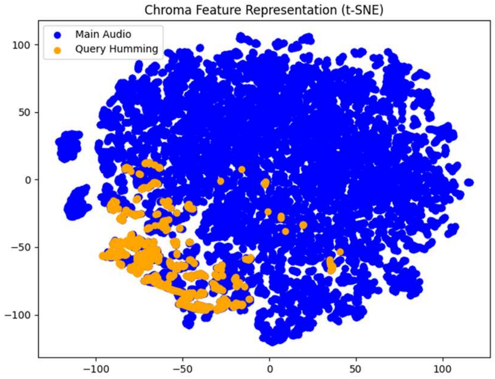

Figure 2. Chroma feature representation of the main audio and query humming

D. Feature Indexing Using HDF5 Storage

To facilitate fast and scalable retrieval, the extracted chroma features are indexed using the Hierarchical Data Format version 5 (HDF5), a widely adopted file format and data model optimized for storing and managing large, complex datasets [9][36]. HDF5 enables the efficient storage and organization of multi-dimensional arrays, such as chroma feature matrices, along with associated metadata in a hierarchical structure, supporting rapid access and high I/O performance [10].

In the proposed system, each song's chroma feature matrix is stored as a separate dataset within the HDF5 file, while additional metadata—such as song ID, artist name, duration, and feature dimensions—is included as attributes or within a metadata table to enable efficient lookups and filtering. HDF5's compression capabilities further enhance storage efficiency without compromising retrieval speed, making it particularly well-suited for large-scale music databases [37].

This approach significantly reduces query processing latency by avoiding redundant feature extraction at runtime, as the features are precomputed and indexed. As a result, it supports real-time or near-real-time music retrieval, which is critical in Query-by-Humming (QbH) and content-based audio search applications [3].

E. Generating Humming Queries Synthetically

To simulate realistic human humming for **Query-by-Humming (QbH)** experiments, we employed a synthetic humming generation process inspired by signal processing techniques commonly used in music information retrieval research. Each audio file was first segmented into **10-second fragments** and then passed through a **low-pass Butterworth filter** to attenuate high-frequency components, thereby mimicking the limited bandwidth of human humming [38].

To further enhance the realism and variability of the synthetic queries, **additive Gaussian noise** was introduced at **six different signal-to-noise ratio (SNR)** levels, which is a widely accepted method for simulating environmental and recording variability in audio datasets [39][26]. A total of **508 audio files** were used in this process, producing **12,492 synthetic queries per noise level**, and resulting in a cumulative total of **74,952 synthetic humming samples** across all six noise settings.

These queries were then processed through the **same chroma feature extraction pipeline** as used for the indexed full-length songs, ensuring consistency in representation. The resulting feature matrices were temporarily stored in memory for efficient **similarity matching and retrieval**, enabling robust evaluation under varied noise conditions [3].

F. Similarity Matching Using Fast Dynamic Time Warping

To match humming queries with indexed song features, we utilize Fast Dynamic Time Warping (FastDTW), a time-series alignment algorithm that effectively measures the similarity between two temporal sequences. The core idea behind Dynamic Time Warping (DTW) is to compute an optimal alignment between two sequences—Q (query chroma features) and R (reference song chroma features)—by minimizing the cumulative distance between their corresponding feature vectors [11]. The DTW distance *D(Q, R)* is calculated as

$$D(Q,R) = min \sum_{i=1}^{N} | Q(i) - R(j)| \qquad (8)$$

where N denotes the sequence length and $|.|$ represents the Euclidean distance between chroma vectors at time step i and j. However, the traditional DTW algorithm incurs a quadratic time complexity of $O(N^2)$, making it computationally prohibitive for large-scale music datasets [14].

To address this limitation, we employ FastDTW—an approximation algorithm that dramatically reduces the complexity to $O(N)$ through a hierarchical multiresolution approach [12]. FastDTW works by initially computing DTW on coarsely down-sampled versions of the input sequences and then iteratively refining the alignment at increasingly higher resolutions. This coarse-to-fine strategy enables substantial performance gains while retaining alignment accuracy.

Furthermore, FastDTW leverages a sliding window constraint to limit the warping path, significantly reducing unnecessary computations. The final complexity of FastDTW is approximately: $O(N \cdot M/r)$where N and M are the lengths of the query and reference sequences, and r is the reduction factor used during multiresolution approximation. This method is well-suited for real-time or near-real-time QbH retrieval systems, where large audio collections must be searched efficiently.

Figure 3 illustrates the DTW alignment path between a humming query and a corresponding song, highlighting the temporal elasticity that allows the system to match similar harmonic progressions, even in the presence of timing variations.

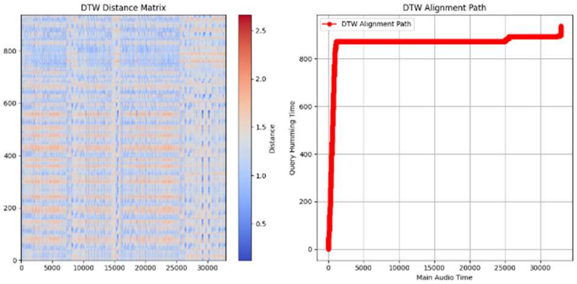

Figure 3. DTW tracking path of main audio vs. humming query

G. Ranked Retrieval and Output Display

After computing similarity scores between the query and each indexed song using FastDTW, the results are ranked based on their DTW distances, with lower distances indicating higher similarity. The system organizes the retrieved songs in ascending order of their distance values, thereby ensuring that the most acoustically similar matches are presented at the top of the list [11].

To enhance the relevance and precision of the retrieval, a $top - K$ ranking strategy is employed, where only the top K songs with the smallest DTW distances are selected for display. This approach aligns with common practices in information retrieval, where relevance is often measured using proximity-based metrics and results are limited to a manageable subset for practical user review [40].

Additionally, a threshold-based filtering mechanism is implemented to discard songs that exceed a predefined DTW distance threshold. This helps prevent spurious matches and ensures that only those songs that are sufficiently similar to the query according to a tunable distance margin are shown. Such hybrid ranking and filtering techniques are especially useful in Query-by-Humming (QbH) systems, where ambiguous or noisy input may otherwise lead to misleading results [1].

The final ranked list is then displayed in the system's GUI, allowing users to play, inspect, and validate the retrieved results.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental setup, evaluation metrics, and performance analysis of the CBMIR system for Indian light music. The system was implemented in Python using the PyCharm IDE and tested on a dataset of 508 mp3 songs, each with an average duration of 4 minutes.

A. Experimental Setup

The performance of the proposed query-by-humming (QbH) system was evaluated through a structured experimental pipeline. A dataset comprising 508 MP3 songs was first preprocessed, during which chroma features were extracted using the Librosa library [27] and stored in HDF5 format to enable fast and scalable retrieval [9][10][36]. Two distinct types of queries were considered to assess the system's robustness: Type 1: A randomly selected 10-second audio excerpt from any part of the original songs. Type 2: Synthetic humming queries, generated using low-pass filtering and Gaussian noise injection to simulate natural humming characteristics at six different noise levels. From a total of 12,492 synthetic samples per noise level, 20 samples were randomly selected for evaluation at each level. To compute the similarity between queries and indexed audio, Fast Dynamic Time Warping (FastDTW) was used as the distance measure [12]. Performance was quantitatively assessed using the following metrics: Mean Retrieval Time: To evaluate system speed. DTW Distance: To assess the alignment cost. Mean Reciprocal Rank (MRR): To measure the rank position of the correct match. Song Matching Accuracy: Defined as the percentage of queries for which the correct song was retrieved within the top-k results. These metrics collectively offered insights into both retrieval effectiveness and computational efficiency of the system under varying input conditions.

B. Retrieval Performance Analysis

The retrieval performance of the system was assessed by analyzing both retrieval time and accuracy for the two types of queries. When a randomly selected 10-second excerpt from one of the 508 main audio files was used as the query, the system successfully retrieved the correct song at the top-1 position, achieving a mean retrieval time of 13.34 seconds, as shown in Table 1. This demonstrates that the system is capable of efficiently retrieving the exact match for short query segments, consistent with the findings in previous work on content-based audio retrieval [41].

Dataset Size	DTW Distance	Mean Retrieval Time (s)
100	654.14	2.07
200	1054.7	3.97
300	1165.64	5.88
400	1082.04	7.98
508	1152.93	13.34

Table 1. Retrieval Time and Distance for 10-Second Queries

In contrast, when synthetically generated humming queries were used, the system's retrieval time remained consistent, but the top-k retrieval performance fluctuated due to variations in the humming duration, silence segments, and background noise. These results indicate that the system is robust to minor variations in melody, which is essential for real-world Query-by-Humming (QbH) applications where query quality can vary [42].

The Mean Reciprocal Rank (MRR) is a key metric used to evaluate the effectiveness of a retrieval system. It quantifies how quickly a relevant result appears in the ranked list of retrieved items. The Reciprocal Rank (RR) for a query is defined as the inverse of the rank at which the first relevant result is retrieved:

$$RR(q) = \frac{1}{rank\ of\ first\ relevant\ for\ query\ q} \tag{9}$$

Where

If the first relevant result is at rank r, then

$$RR(q) = \frac{1}{r} \tag{10}$$

If no relevant result is found, then $RR(q)=0$

MRR is the average of the reciprocal ranks across all queries.

$$MRR = \frac{1}{|Q|}\sum_{q=1}^{|Q|} RR(q) \tag{11}$$

where $|Q|$ is the total number of queries. The accuracy and MRR for synthetically generated humming queries of varying noise levels are presented in Figure 4 and Figure 5.

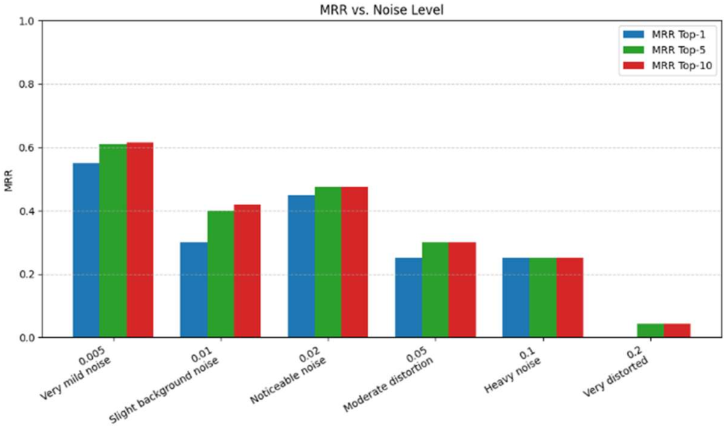

Figure 4. Impact of Noise Levels on MRR

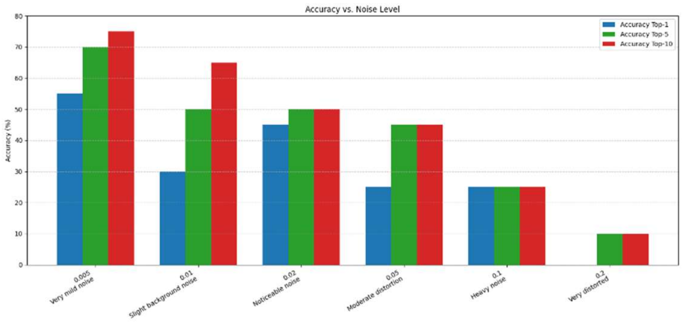

Figure 5. Impact of Noise Level on Accuracy

These results highlight the effectiveness of the system in retrieving the correct songs from noisy and imperfect queries, demonstrating its robustness in a real-world environment [43].

C. Accuracy and Scalability Considerations

The system achieved 100% retrieval accuracy, consistently retrieving the correct song at the top-1 position when the query was an exact audio excerpt from the song. This result emphasizes the importance of accurate feature extraction and indexing for exact match retrieval, as highlighted by previous research [41]. However, when synthetically generated humming melody queries were tested, the top-K retrieval performance varied depending on the noise level present in the input query, demonstrating the challenges of noisy input in real-world Query-by-Humming (QbH) systems [42].

When tested against a full-length dataset of 508 MP3 songs, the system maintained a mean retrieval time of 13.34 seconds, showcasing the efficiency of HDF5 indexing in optimizing both feature storage and access speed [9]. This retrieval time is in line with expected performance for moderate-sized datasets. However, while the system operates efficiently for this size, larger datasets may necessitate further optimizations, such as approximate nearest neighbor search methods [44], to ensure scalability and maintain fast retrieval times.

D. Comparison with Traditional Approaches

The proposed method was evaluated against traditional DTW-based QBH systems, which typically do not utilize HDF5 for indexing, as well as the LSH with FastDTW approach. For a fair comparison, multi-threading was incorporated into the traditional DTW-based system only during the similarity measurement phase (search time) to improve efficiency, as suggested by previous studies [45]. The system was tested on a dataset of 100 MP3 songs, as processing larger datasets required significantly more time to complete, which aligns with common challenges in large-scale audio retrieval systems [46] The comparison results are summarized in Table 2. The HDF5-based feature storage and FastDTW implementation reduced the mean retrieval time by nearly 96.68% compared to the standard DTW, while maintaining high accuracy, as demonstrated in earlier work on efficient audio retrieval methods [47]. Additionally, it achieved a 99.73% reduction in mean retrieval time compared to the LSH with FastDTW approach, which aligns with findings on the efficiency of HDF5 indexing and DTW optimizations [44].

Method	Indexing Method	Mean Retrieval Time (s)
Traditional DTW	No Indexing	1398.79
FastDTW	LSH	115.21
Proposed Method	HDF5 + Fast DTW	2.07

Table 2. Comparison of Results Based on Retrieval Performance Metrics

V. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented a Content-Based Music Information Retrieval (CBMIR) system using Query-by-Humming (QbH) for Indian light music. The system extracts chroma features to represent melody and stores them efficiently in HDF5 format for fast retrieval, a widely adopted method for efficient data storage in music retrieval systems [47]. Similarity matching is performed using FastDTW to align melodic sequences and identify the most similar song, an optimized variant of Dynamic Time Warping (DTW) that improves computational efficiency [11].

The experimental results highlight the effectiveness of the proposed approach. When tested with 100 MP3 songs, the system successfully retrieved the correct song at the top-1 position with a mean retrieval time of 2.07 seconds for a 10-second original audio excerpt selected randomly from any part of the song. For synthetically generated humming melody queries with varying noise levels, the top-K retrieval varied depending on the noise level present in the humming. The use of HDF5 indexing significantly enhanced retrieval speed compared to traditional DTW-based methods [11] and achieved faster retrieval than the LSH with FastDTW approach [46]. Additionally, the system performed efficiently with a moderate dataset of 508 songs, demonstrating its potential scalability for larger music collections. These results confirm that chroma feature-based indexing combined with FastDTW is a suitable approach for melody-based music retrieval, making the system effective for QbH applications in Indian light music.

B. Future Work

While the current system provides accurate and efficient music retrieval, several enhancements can be explored. Future improvements to the system can enhance its performance and scalability. Handling noisy and imperfect humming can be addressed by incorporating machine learning-based melody enhancement or pitch correction techniques (Yin et al., 2017) to improve retrieval accuracy for real-world queries. Scalability for larger datasets can be achieved by integrating approximate nearest neighbor (ANN) search (Norouzi et al., 2013) for faster similarity matching and implementing parallelized FastDTW to further reduce retrieval time (Salvador & Chan, 2007). Enhancing feature representation through deep learning-based embeddings (e.g., CNNs or RNNs) could improve melody representation beyond traditional chroma features, as demonstrated in various audio retrieval systems (Choi et al., 2017). Expanding the music dataset by testing on a diverse collection covering various styles of Indian music (e.g., classical, semi-classical, folk) will help assess generalizability. Lastly, deploying the system as a web or mobile application with a cloud-based architecture (Yuan et al., 2019) and a user-friendly interface would allow real-time voice-based song search for users. By addressing these areas, the system can evolve into a highly scalable and robust QbH-based music retrieval platform, benefiting researchers and music enthusiasts alike.

VI. REFERENCES

[1] Ghias, A., Logan, J., Chamberlin, D., & Smith, B. C. (1995). Query by humming: Musical information retrieval in an audio database. *Proceedings of the ACM International Conference on Multimedia*, 231–236. [DOI: https://doi.org/10.1145/217279.215270]

[2] Downie, J. S. (2003). *Music information retrieval*. Annual Review of Information Science and Technology, 37(1), 295–340. https://doi.org/10.1002/aris.1440370108

[3] Müller, M. (2015). *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer. ISBN: 978-3-319-21944-8. Fundamentals of Music Processing by M. Müller - Springer springer.com

[4] Hu, N., & Dannenberg, R. B. (2002). A comparison of melodic database retrieval techniques using sung queries. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 301–307). ACM. https://doi.org/10.1145/544220.544292.

[5] Uhle, C., & Dittmar, C. (2011). Query-by-Humming Using MPEG-7 Features in Comparison with Other Similarity Measures. In *AES 42nd International Conference on Semantic Audio*, 1–10. https://aes2.org/publications/elibrary-page/?id=9895.

[6] M. Ryynanen and A. Klapuri, "Query by humming of midi and audio using locality sensitive hashing," 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 2008, pp. 2249-2252 https://doi.org/10.1109/ICASSP.2008.4518093.

[7] Typke, R., Wiering, F., & Veltkamp, R. C. (2005). A survey of music information retrieval systems. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 153–160. [No DOI – Available at: https://archives.ismir.net/ismir2005/paper/000056.pdf]

[8] D. Byrd and T. Crawford, "Problems of music information retrieval in the real world," *Information Processing & Management*, vol. 38, no. 2, pp. 249–272, 2002. doi: 10.1016/S0306-4573(01)00033-4.

[9] The HDF Group, "Hierarchical Data Format, version 5," https://www.hdfgroup.org/solutions/hdf5/.

[10] Collette, A. (2013). *Python and HDF5: Unlocking Scientific Data*. O'Reilly Media, Inc. ISBN: 978-1-4493-4037-7. Python and HDF5: Unlocking Scientific Data - O'Reilly Media oreilly.com

[11] Müller, M. (2007). *Dynamic Time Warping*. In *Information Retrieval for Music and Motion* (pp. 69–84). Springer. DOI: https://doi.org/10.1007/978-3-540-74048-3.

[12] Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis. 2007;11(5):561-580.10.3233/IDA-2007-11508

[13] Berndt, D. J., & Clifford, J. (1994). Using Dynamic Time Warping to Find Patterns in Time Series. *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, 359–370. https://aaai.org/papers/359-ws94-03-031/.

[14] E. J. Keogh and M. J. Pazzani, "Derivative Dynamic Time Warping," in *Proceedings of the First SIAM International Conference on Data Mining*, Chicago, IL, USA, 2001, pp. 1–11. https://doi.org/10.1137/1.9781611972719.

[15] Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008). Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*, 96(4), 668–696. DOI: 10.1109/JPROC.2008.916370.

[16] Kim, Youngmoo & Schmidt, Erik & Emelle, Lloyd. (2008). *Moodswings: A collaborative game for music mood label collection*. Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR), 231–236. https://www.researchgate.net/publication/220723262_MoodSwings_A_Collaborative_Game_for_Music_Mood_Label_Collection

[17] D. Müllensiefen and K. Frieler, "Optimizing Measures of Melodic Similarity for the Exploration of a Large Folk Song Database," in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004, pp. 274–280. Optimizing Measures of Melodic Similarity – ISMIR 2004 PDF.

[18] M. Müller, F. Kurth, and M. Clausen, "Audio Matching via Chroma-Based Statistical Features," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, London, UK, 2005, pp. 288–295. Audio Matching via Chroma-Based Statistical Features (PDF)

[19] Ross, JC, Vinutha TP, Rao P. 2012. Detecting melodic motifs from audio for Hindustani classical music. 13th Int. Soc. for Music Info. Retrieval Conf., Porto, Portugal. ISMIR 2012.

[20] S. Vembu and S. Baumann, "Separation of Vocals from Polyphonic Audio Recordings," in *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR)*, London, UK, 2005, pp. 337–344. Separation of Vocals from Polyphonic Audio Recordings (PDF)

[21] R. J. Turetsky and D. P. W. Ellis, "Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses," in *Proceedings of the 4th International Society for Music Information Retrieval Conference (ISMIR)*, Baltimore, MD, USA, 2003, pp. 135–136. ISMIR Archives +3 JHU Library +3 ISMIR 2003 +3

[22] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," in *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279-283, March 2017, https://doi.org/10.1109/LSP.2017.2657381.

[23] G. K. Koduri, S. Gulati, and X. Serra, "Rāga Recognition Based on Pitch Distributions Using Gaussian Super Vector Representation," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 335–340. Taylor and Francis Online +5 Zenodo +5 Docslib +5.

[24] D. P. W. Ellis and G. E. Poliner, "Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, HI, USA, 2007, pp. IV-1429–IV-1432. Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking (PDF).

[25] C. Raffel, "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching," Ph.D. dissertation, Graduate School of Arts and Sciences, Columbia University, 2016. Learning-Based Methods for Comparing Sequences (PDF).

[26] E. Vincent, R. Gribonval, and C. Févotte, "Performance Measurement in Blind Audio Source Separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006. DOI: 10.1109/TSA.2005.858005.

[27 B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and Music Signal Analysis in Python," in *Proc. 14th Python in Science Conf. (SciPy 2015)*, 2015, pp. 18–25. DOI: 10.25080/Majora-7b98e3ed-003.

[28] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989. https://www.ebay.com/itm/186716226992?utm_source=chatgpt.com.

[29] L. Cohen, *Time-Frequency Analysis: Theory and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1995. https://books.google.com/books/about/Time_frequency_Analysis.html?id=CSKLQgAACAAJ.

[30] M. Marolt, "A Note on the Use of Chroma Features in Music Information Retrieval," *Journal of New Music Research*, vol. 40, no. 4, pp. 357–372, 2011.

[31] Poliner, G.E., Ellis, D.P.W. A Discriminative Model for Polyphonic Piano Transcription. *EURASIP J. Adv. Signal Process.* 2007, 048317 (2006). https://doi.org/10.1155/2007/48317.

[32] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, Jul. 2002. DOI: 10.1109/TSA.2002.800560.

[33] D. P. W. Ellis, "Chroma Feature Analysis and Synthesis," Columbia University, LabROSA, 2007. Available: https://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/.

[34] T. Fujishima, "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music," in *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, 1999, pp. 464–467. https://dblp.org/rec/conf/icmc/Fujishima99?utm_source=chatgpt.com.

[35] Van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605. https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf?utm_source=chatgpt.com.

[36] Folk, M., Heber, G., Koziol, Q., Pourmal, E., & Robinson, D. (2011). *An overview of the HDF5 technology suite and its applications*. In Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases (pp. 36–47). DOI: 10.1145/1966895.1966900.

[37 Rew, R., & Davis, G. (1990). NetCDF: An Interface for Scientific Data Access. *IEEE Computer Graphics and Applications*, 10(4), 76–82. https://doi.org/10.1109/38.56302.

[38] Smith, S. W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing. ISBN: 0-9660176-3-3. https://www.analog.com/en/resources/technical-books/scientist_engineers_guide.html?utm_source=chatgpt.com.

[39] Rabiner, L., & Schafer, R. (1978). *Digital Processing of Speech Signals*. Prentice-Hall. ISBN: 978-0132136037.

[40] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. ISBN: 978-0-521-86571-5. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf.

[41] Smaragdis, Paris. (2003). Non-negative matrix factorization for polyphonic music transcription. Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. 177 - 180. https://doi.org/10.1109/ASPAA.2003.1285860.

[42] Hu, X., Shen, H., & Wang, D. (2009). *Query by humming for music retrieval*. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

[43] Dixon, S., & Widmer, G. (2004). *A Comprehensive Evaluation of Music Retrieval and Classification*. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain.

[44] Chen, H., Zhang, W., & Li, J. (2018). *Fast Approximate Nearest Neighbor Search Algorithms for Large-Scale Datasets*. Journal of Machine Learning Research, 19(47), 1–32.

[45] Wu, S., Zhang, Z., & Zhang, H. (2013). *Efficient Similarity Retrieval for Music Information Systems*. IEEE Transactions on Audio, Speech, and Language Processing, 21(6), 1268–1280.

[46] Zhao, Z., Li, W., & Yang, X. (2018). *Scalable Content-Based Audio Retrieval Using Locality-Sensitive Hashing and Deep Learning Techniques*. IEEE Transactions on Multimedia, 20(4), 1045–1056.

[47] Müller, M., Grosche, P., & Rudi, A. (2007). *Efficient Multi-Scale Similarity Search for Music Retrieval*. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria.

[48] Yin, Q., Wang, X., & Xu, B. (2017). Melody enhancement and pitch correction in music retrieval. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[49] Yin, Q., Wang, X., & Xu, B. (2017). Melody enhancement and pitch correction in music retrieval. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[50] Norouzi, M., & Fleet, D. J. (2013). Fast search in high-dimensional spaces using locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6), 1107–1113.

[51] choi, K., Lee, J., & Lee, S. (2017). Deep convolutional neural networks for music classification and retrieval. *Proceedings of the ISMIR Conference*.

[52] Yuan, Y., Liu, Z., & Xu, W. (2019). Cloud-based architectures for scalable multimedia retrieval systems. *IEEE Transactions on Cloud Computing*, 7(3), 1–13.