

Approximate Nearest Neighbour-based Index Tree: A Case Study for Instrumental Music Search

Hai Thanh Nguyen*, Linh Dan Vo, Thien Thanh Tran

College of Information and Communication Technology, Can Tho University, Can Tho, Vietnam

Abstract – Many people are interested in instrumental music. They may have one piece of song, but it is a challenge to seek the song because they do not have lyrics to describe for a text-based search engine. This study leverages the Approximate Nearest Neighbours to preprocess the instrumental songs and extract the characteristics of the track in the repository using the Mel frequency cepstral coefficients (MFCC) characteristic extraction. Our method digitizes the track, extracts the track characteristics, and builds the index tree with different lengths of each MFCC and dimension number of vectors. We collected songs played with various instruments for the experiments. Our result on 100 pieces of various songs in different lengths, with a sampling rate of 16000 and a length of each MFCC of 13, gives the best results, where accuracy on the Top 1 is 36 %, Top 5 is 4 %, and Top 10 is 44 %. We expect this work to provide useful tools to develop digital music e-commerce systems.

Keywords – Index tree, instrumental songs, MFCC, piece of songs.

I. INTRODUCTION

In recent years, along with the instantaneous development trend of modern society, people from all treks of life and all occupations have to live a disorganized vitality with assignments and imperceptible pressures about thriftiness, family, and relationships. Since then, each individual's enjoyment needs have gradually increased and are considered an effective method to relieve stress and fatigue. One method that satisfactorily conforms to people's busy lifestyles today is music [1]. When it comes to music, instrumental music is seen as an effective solution for mental health [2] with unique benefits, including relieving anxiety and helping the brain, thereby helping to improve the work efficiency of each individual.

In terms of instrumental music, it is impossible not to mention the extraordinary miracles it brings to people of all generations. Firstly, numerous researchers have seen the potential of instrumental music for the health and well-being of the elderly and have produced many studies, such as the outcome of instrumental and vocal harmony on adherence to a biological restoration exercise agenda with individuals who are elderly [3], musical training and brain magnitude in more old grown-ups [4], and the ability of melody in the dynamism of

older adults [5]. Secondly, musical instruments also help improve the performance of young people, especially those who work in an office environment, requiring high concentration and having to sit for hours in front of a computer screen. The absence of lyrics and vocals will help people to concentrate sufficiently. It is a sort of introspection. Hearing more instruments can bring people into a flow state (FS) from which any complicated task can be accomplished. It helps thwart backdrop cacophony and provides listeners with calming effects that increase focus and help them stay productive to achieve any creative mission. Finally, instrumental music brings not only positive benefits to adults but also numerous godsend to the brain development of young children [6]. For instance, a study [7] has shown that learning to play an instrument positively affects mental skills and academic accomplishment. Besides, children and adolescents who self-select for musical training tend to perpetrate satisfactorily at baseline. Hence, the market for instrumental music is also expanding gradually. Based on Facts and Factors ¹, the international musical instruments market dimensions were cherished at around \$13 billion in 2021 and are estimated to extend almost \$15.2 billion by 2028, at a CAGR of roughly 2.34 % from 2022 to 2028.

Moreover, scrutinizing the possibility of the digital music industry in this era, revenue in the digital music segment is expected to acquire \$15.07 billion by 2022 ². The market colossal component is music streaming, with market magnitude expected to be \$13.66 billion by 2022. In international comparison, most of the payment will be induced in the United States in 2022 with \$5901.00 million. These figures have shown the music industry's conceivable and digitization trend in contemporary society. Singers and composers now predominantly have their primary source of income from e-commerce through streaming services, the most prevalent today being Spotify and Apple Music. It is also doable to monetize through a distribution network that allocates and manages royalty from those streaming courtesies, with some well-known distribution networks, including Landr, TuneCore, CD Baby, and Amuse.

Nowadays, with the development of Artificial Intelligence (AI) and the trend of applying machine learning models to

*Corresponding author's email: nthai.cit@ctu.edu.vn

¹ <https://www.linkedin.com/company/finfresearch/?originalSubdomain=in>

² <https://www.statista.com/outlook/dmo/digital-media/digital-music/worldwide#revenue>

multiple aspects of contemporary society [8], numerous deconstructions have assembled machine learning models to recognize instrumental music to make finding the desired music more accessible. Determining instruments from musical passages has miscellaneous applications in acoustic engineering and musicology. In musicology, identifying unique mechanisms in orchestral music is immensely important and plays a massive role in how the combinations of instruments evolve. Nevertheless, the identification task consistently needs to be improved. Accordingly, applying the machine learning model to this aspect is essential and brings high applicability in life in general and musicology in particular.

This study uses the Approximate Nearest Neighbours³ (ANN) algorithm and some python libraries to search for songs by file and path to help users quickly find information about the song. Furthermore, this research performs additional audio-to-spectral image conversion and then uses machine-learning models to analyse those spectrograms. The following are our principal contributions to the search for instrumental music.

- We collect instrumental songs for the experiments to build the index tree, and for the testing phases with various lengths and played with different instruments.
- We build an index tree with investigations on hyper-parameters to find songs based on a piece of music. From the experiments, we provide an efficient method in terms of search results and time optimization to meet people's search ought as quickly as practicable.
- We integrate the model into the website to increase usability, provide an intuitive interface and make search operations easier to increase the applicability to help the system reach most users. Besides, the built website also advocates users to search with Top 1, Top 5, and Top 10 songs similar to the tunes given by users to explore, thus providing diverse search capabilities and fully satisfying the necessary pursuit information.

This paper is organised as follows. In Section II, we first familiarize the related work. Then, we investigate how to unravel the issue in Section III (Method Section). Next, the Experimental Section (Section IV) indicates our suggested technique experiments. Eventually, we provide some closing remarks in Section V.

II. RELATED WORK

Hitherto, considerable research has been published with continuously improved algorithms that are highly effective in searching for instrumental music. Most of the studies produce consequences with fantastic precision. They interest bibliophiles worldwide, especially in countries with a protracted musical history with multiple traditional instruments imbued with national cultures, such as China, Thailand, India, and Mexico. Incredibly instrumental music based on traditional instruments brings an unforgettable tune to the listener that sometimes people will accidentally hear somewhere, and searching for information about these songs takes much time. Thus, the research and development of a machine learning-

based traditional instrument recognition system in these countries is necessary and has received the support of numerous music enthusiasts.

Following this movement, a study in [9] presented Shruti and Sargam (terms in Indian Classical Music utilized to direct to the corresponding tone and musical note) recognition using machine learning for Instrumental Sounds. This examination used the Dunya dataset (Carnatic and Hindustani Instrumental) taken from the Comp Music oeuvres of Indian art music and has shown a wildly positive result. Besides, research in [10] compared the Mel-Frequency Cepstral Coefficient (MFCC) [11]–[12], Mel-Spectrogram feature, and CNN-based deep learning sample operated for Thai classical instrument sound distinction. The results showed that MFCC with CNN accomplished nicely in the test with 99.44 % precision and 0.99 F1-score.

Furthermore, researchers have constantly been learning and improving the approach. Hence, investigations are gradually showing improvements in performance as well as reliability. Some studies even give results achieved with almost absolute accuracy in a laboratory environment with available datasets. To illustrate, research [13] proposed a musical instrument classification algorithm combining multichannel elements and XGBoost to perform instrument recognition. This analysis used the Medley-solos-DB dataset and yielded excellent test results with up to 97.65 % accuracy. Besides, the study [14] desires to analyse and classify Indian musical instruments according to their features using the system intelligence. For category, the deconstruction utilized KNN and SVM classifiers and experimented with the algorithm on a database of 60 wav files, producing an exactness of 87 % with KNN and 92 % with SVM. While research [15] conducted musical instrument and song classification using spectrogram-founded contextual characteristics, the study used the Random Sample and Consensus (RANSAC) [16] technique. The outcomes indicated that precision was 95 % when stretching on 300 instrumental and 300 melody files. With a different approach, the study [17] used a newly offered architecture for sound processing and instrument detection utilizing Convolution Neural Networks [18] (CNNs), Multilayer Perceptrons [19] (MLPs), and Recurrent Neural Networks – Long Short Term Memory [20] (RNN-LSTM). The examinations were enforced operating actual data materialized including 20 000 records. Three deep learning methods were executed and analogized with existing possible pristine solutions. The data and illustrations above show the importance of studying a machine-learning system that determines instrumental music. Therefore, this research will aid the community, especially music enthusiasts.

III. METHOD

Regarding the strategy, an audio harmony search system is built with Python and ANN. First, we use libraries that support getting data from the machine with the input of files with the extension mp3 or wav. Then, with the music data in the database, we use the librosa library [21] for preprocessing to

³ <https://ignite.apache.org/docs/latest/machine-learning/binary-classification/ann>

convert the audio into a digitized form into a sequence of bit sequences for the computer to understand. Then, we get the MFCC feature extraction of the audio with the python speech features library [22] to get the files containing the features of the music store. Finally, we use the annoy library [23], which applies the Approximate Nearest Neighbours (ANN) algorithm to create indices for the music store. To explore, we need to assume the music we want to discover and preprocess it into a digitized form of bit sequences, re-quote the featured files, and then reload the indices created formerly. Next, we perform a search in the index using distinct parameters. Finally, the users enter the number of songs to encounter to print out a list of top songs that are similar to the piece of music – changing different parameters to find the parameter cluster that can provide the highest precision and estimate the search performance for each parameter. This section presents the system primary technique for finding instrumental music by analysing the sound. The data include instrumental music files from non-verbal music pieces randomly picked up on the internet and musical instruments such as piano, bamboo flute, western trumpet, and guitar. Besides, the duration of each segment constantly fluctuates in the range of 30, 60, or 90 s. The first step is to create a song-based index tree with different parameters built from the number of music files in the dataset (which will be covered in more detail in the following sections). The next step is to preprocess the data by returning it to a digitized form with an appropriate sampling rate. For the scope of the study and scrutiny, we use sampling rates like 16 000 and 40 000 to give the best results. The following step is to perform feature extraction steps. To seek a song using a piece of song, the piece of a song will compute a similarity score comparing the songs in the index tree.

Furthermore, when searching for the number of songs with the identical melody as the piece of music to be searched, within the research dimensions, we advocate searching by Top 1, Top 5, and Top 10 songs with high resemblance, respectively. The search process will take place to check which song has the highest similarity with which song in the built index tree. The workflow is summarized in Fig. 1 and will be described in the following sections.

A. Digitizing

Fundamentally, audio digitization converts audio into a sequence of bits. Since the computer cannot comprehend the sound, it has to be converted to a bit sequence so that the computer can process it. We will take the piece of music that ought to be digitized and then use Python's librosa library to digitize the audio clip with a sampling rate of 16 000. The sampling rate of audio is the number of samples taken per second. They are measured as samples per second or Hertz (abbreviated as Hz or kHz with 1 kHz = 1000 Hz). In addition, a sound sample is a measurement representing the value of a sound wave at a particular time. These audio samples are taken at some point in 1 s. Therefore, a sampling rate of 16 000 does not mean that 16 000 samples were taken in one second; they must be taken 1/16 000th of a second apart. Furthermore, the

computer can understand and perform the following steps after digitizing it into a sequence of bits.

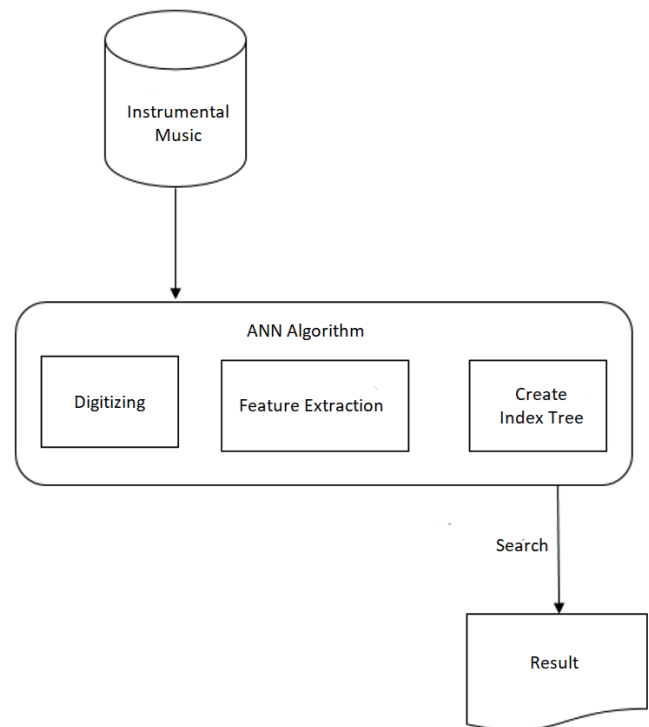


Fig. 1. Overall architecture model of the system.

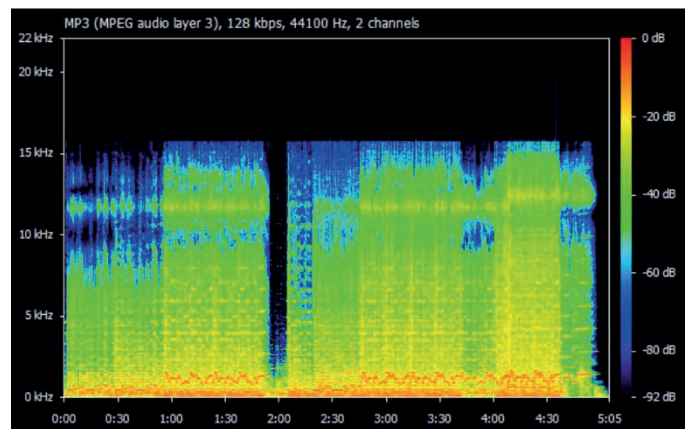


Fig. 2. An illustration of converting a piece of audio to a bit sequence.

B. Mel Frequency Cepstral Coefficients (MFCC)

For MFCC, using the python speech feature library for processing is an effective solution to save time [24]. In order to generate the full features of the songs in the database and avoid information loss, sliding on the time axis of the sound is the method used. At this point, the generation of feature vectors of the songs in the database is considered complete.

Flow-wise feature extraction can be visualized as follows: cutting the audio signal sequence into short equal segments and overlapping each other. Then, each audio clip is transformed and calculated to obtain features. Besides, each listed feature has high independence. Moreover, low noise is slight adequately to allow for computation and enough information to ensure quality for recognition algorithms.

C. Creating an Index Tree

To create the index tree, the process of using an indexing library is considered the best on the number vector search performed at that time, the annoy index. Therefore, the next stage is to add data to annoy indexing and construct an index tree. The approach is developed using the python programming language and supporting libraries such as annoy, librosa, os, soundfile, tqdm, numpy, mfcc, and pickle. Besides, the system website is built by HTML, CSS, Javascript, and Bootstrap to create an attractive and easy-to-use interface. As for the backend, due to it being a python-integrated website, it uses a Python Framework, Flask, to create an API, returning search results from a pre-written algorithm. To be able to use the system, users need a web browser, which can be: Google Chrome, Microsoft Edge, Firefox, or any browser that can connect to the system address.

IV. EXPERIMENTS

A. Environment Backgrounds and Dataset Characterization

Feature extraction and song search were carried out with the sampling rate $SR = 16\,000$, along with changing the parameter n_{filt} (Length of each MFCC) from 10 to 14 and parameter f (Vector dimension) varying from 100 to 140. The experiment consists of two test cases. The first is to search separately for 10 lessons of 30 s, 10 lessons of 60 s, and 10 lessons of 120 s in the test file. The other case conducts the test by simultaneously searching all 30 songs in the test file. The training file is 200 songs, with detailed data in Table I.

Specifically, the 30 songs used to search are performed in different musical instruments, including bamboo flute, piano, guitar, and trumpet. In addition, there are numbers mentioned in Table II.

TABLE I

NUMBER OF MUSIC USED TO BUILD THE INDEX TREE AND DEPLOYED IN THE TESTING PHASE

Files for building the index tree	Files for test cases
200 songs	30 songs

TABLE II

NUMBER OF SONGS PLAYED BY DIFFERENT INSTRUMENTS USED IN THE TESTING PHASE

Bamboo flute	19 songs
Piano	9 songs
Guitar	1 song
Trumpet	1 song

We collected song audio files with various lengths. As a result, 30 files are uniformly contributed (10 files for each) to average lengths of 30 s, 60 s, and 120 s, respectively, as shown in Table III.

During the experiment, the choice of the dataset is exceptionally critical because it can directly affect the algorithm results. For this study, the dataset possesses 200 songs randomly taken from the internet: the training set consists of 100 instrumental songs, and the test set includes 100 instrumental melodies with the same name as the training set, but 30 songs are played with a bamboo flute, 40 songs are recreated on the piano, and 30 songs are cut about 2 min from

the training set. After testing, it is necessary to select the most satisfactory SR parameter set with the data set to be decreased to 100 lessons in both the training and test files (100 lessons out of 200 in the training file). The test file is the instrumental songs partially cut from the training set, and the harmonies are played with a different instrument than the songs in the training set. The parameter is still n_{filt} (Length of each MFCC) will be modified from 10 increments to 20, and the parameter f (Number of vector dimensions) will alter from 100 increments to 200. Then, it is necessary to approximate the search results to find the best parameters for the algorithm. After testing and selecting the best SR parameter set from the above data set, it is necessary to continue with each train and test dataset with 100 songs taken from 200 songs in the first test. Table IV shows the number of music in the training file and the number of music used for testing.

TABLE III

NUMBER OF SONGS ACCORDING TO EACH AVERAGE LENGTH

30 s	10 songs
60 s	10 songs
120 s	10 songs

TABLE IV

NUMBER OF SONGS IN TRAINING FILE AND TEST FILE

File train	File test
100 songs	100 songs

More specifically, the 100 songs used to search are performed in different instruments, including bamboo flute, piano, and some cuts from the training set. There are numbers mentioned in Table V. Besides, the training set is the background music set of the songs.

TABLE V

A NUMERAL OF SONGS BY INSTRUMENTS IN A TEST FILE

Bamboo flute	30 songs
Piano	40 songs
From the training set	30 songs

The project is based on Python programming language using its libraries, and Anaconda simplifies the installation, management, and deployment of packages (numpy, scipy, and librosa) is an open-source data science platform on Python. Together with Keras [25] – an open-source neural network that supports running the model on the CPU. There is also a song search on the website platform.

B. Assessment Implementation Using ANN

After conducting the first experiment and getting the best set of SR parameters, we used this set of parameters to experiment. With the following set of parameters: sampling rate $SR = 16\,000$, n_{filt} (length of each MFCC) varied from 10 to 20, f (number of vector dimensions) varied from 100 to 200. We experimented with the above parameter set, got preprocessing time, and created an index tree, as shown in Table VI.

Table VI shows the timing data after preprocessing the 200-song data with different MFCC lengths and vector dimensions. For example, from Table VI, it can be seen that with $SR = 16\,000$, the preprocessing time is the briefest when the length of each MFCC is 15, and the number of vector

dimensions is 150. Besides, the time to create the index tree is the swiftest when the length of each MFCC corresponds to 10, and the number of vector dimensions is 100. Vice versa, each MFCC length corresponds to 20 and the number of vector dimensions is equal to 200 results in the lengthiest preprocessing time and index tree creation time. Finally, the results of accuracy and time when searching for 100 songs in the experiment are shown in Table VII.

TABLE VI

PREPROCESSING TIME AND INDEX TREE GENERATION WITH SR = 16 000

No.	Length per MFCC	Vector's number of dimensions	Preprocessing time (s)	Index tree creation time (s)
1	10	100	2307.6093	97.5171
2	11	110	2163.7187	109.0306
3	12	120	2042.8906	123.0786
4	13	130	2205.0468	143.5533
5	14	140	2355.0000	162.4916
6	15	150	2030.5312	174.5036
7	16	160	2350.1250	203.1874
8	17	170	2217.1250	215.1088
9	18	180	2117.6562	235.3296
10	19	190	2095.7031	256.4556
11	20	200	2486.2187	316.7184

TABLE VII

EXPERIMENTAL SEARCH ACCURACY AND DURATION

No.	MFCC length (nfil)	Preprocessing time (s)	Average search time per song (s)	Standard deviation	Top	Accuracy (%)
1	10	1568.8281	18.1651	9.2946	1	34
2	10	1644.5156	20.8691	13.9799	5	40
3	10	1715.2500	22.4102	16.2807	10	44
4	11	1508.2500	15.6184	6.7765	1	34
5	11	1531.1093	15.8730	15.8730	5	41
6	11	1671.9843	20.5006	14.2068	10	46
7	12	1636.8437	18.1613	8.9513	1	35
8	12	1653.5781	21.3629	14.3757	5	42
9	12	1548.8593	20.0558	14.3557	10	44
10	13	1530.8437	16.2874	8.4268	1	36
11	13	1561.7812	16.0468	7.4862	5	41
12	13	1601.1718	17.2047	7.7870	10	44
13	14	1498.5156	15.1299	6.9555	1	35
14	14	1675.7812	18.1618	8.2767	5	41
15	14	1540.5781	17.6279	9.2006	10	47
16	15	1613.4531	17.5622	7.8999	1	35
17	15	1550.1406	16.0454	7.2342	5	39
18	15	1602.5156	21.9663	16.6067	10	43
19	16	1736.2187	20.4187	8.7606	1	34
20	16	1797.8906	21.0904	9.7973	5	39
21	16	1798.0781	19.7155	9.9768	10	47
22	17	1542.2968	16.0699	7.3036	1	36
23	17	1599.0937	16.8367	7.5731	5	42
24	17	1671.7968	17.5608	7.6081	10	47
25	18	1580.8750	16.1403	6.4486	1	36
26	18	1800.5468	21.1081	10.2044	5	40
27	18	1656.7812	17.3278	7.6420	10	45
28	19	1841.9843	20.7565	8.8598	1	35
29	19	1850.1250	20.1163	9.5773	5	41
30	19	1703.6875	18.3319	9.4746	10	46
31	20	1810.6093	21.5093	8.8304	1	35
32	20	1700.5625	21.8444	13.5740	5	41
33	20	1875.0312	29.8705	16.0555	10	47

According to the results shown in Table VII, it can be seen that with the parameter nfil (length of each MFCC) = 17, the accuracy of the results is the highest in Top 1, while the parameter nfil = 17 gives the highest accuracy. Furthermore, the best accuracy is in the Top 5, while the parameter nfil = 17 gives the best accuracy in the Top 10. Finally, the results shown in Fig. 3 can be observed for better visualization.

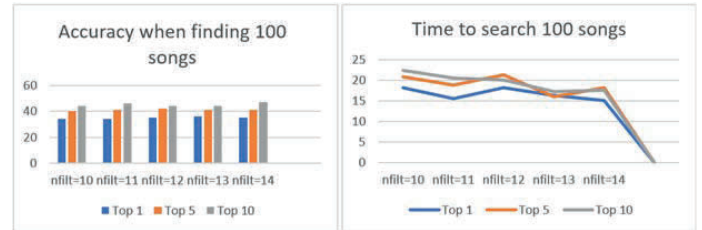


Fig. 3. Graph showing search accuracy and time.

Besides, with a sampling rate of SR = 16 000, the study experimented to find 10 songs in 30 s, 10 songs in 60 s, and 10 songs in 120 s to observe the influence of music length on algorithm search results. The results are illustrated in Tables VIII–X.

Table VIII shows the details of time and accuracy when searching for songs with length MFCC (nfil = 10). Search results with a 30-second piece of music were still looking for the right result. The 60-second tracks only found the correct results in the Top 5 and Top 10. The 120-second tracks produced better results in the Top 1, Top 5, and Top 10.

TABLE VIII

SEARCH ACCURACY RESULTS WITH NFILT = 10

Piece of song duration	Preprocessing time (s)	Average search time per song (s)	Standard deviation	Top	Accuracy (%)
30	29.125	4.3413	1.3634	1	0
30	28.5312	13.3411	5.1754	5	0
30	31.5468	20.5923	9.1846	10	0
60	67.9843	6.9322	1.3846	1	0
60	55.9062	11.7524	3.6777	5	10
60	62.1250	14.4790	7.3293	10	10
120	108.3906	17.666	5.4814	1	20
120	103.5625	18.4674	5.5713	5	30
120	113.6875	33.8562	12.3153	10	30

TABLE IX

SEARCH ACCURACY RESULTS WITH NFILT = 11

Piece of song duration (s)	Preprocessing time (s)	Average search time per song (s)	Standard deviation	Top	Accuracy (%)
30	30.0781	8.2648	2.9622	1	0
30	30.0000	9.1231	3.4095	5	0
30	30.7656	19.5927	6.7794	10	0
60	58.2812	5.6676	2.5416	1	0
60	61.7343	11.8657	3.6887	5	20
60	64.0937	16.3280	4.1652	10	20
120	101.2343	10.0245	0.6019	1	30
120	106.6718	18.4831	6.0516	5	30
120	112.2656	26.2612	15.4866	10	40

Table IX shows the details of time and accuracy when searching for songs with length MFCC (nfilt = 11). Search results with a 30-second piece of music still show unfavourable results. The 60-second tracks only found the correct results in the Top 5 and Top 10. The 120-second tracks produced better results in the Top 1, Top 5, and Top 10.

Table X shows the details of the time and accuracy when searching for songs with length MFCC (nfilt = 12). Search results for a 30-second piece of music still need improvement. The 60-second tracks only found accurate results in the Top 5 and Top 10. The 120-second tracks delivered more profitable results in the Top 1, Top 5, and Top 10.

TABLE X
SEARCH ACCURACY RESULTS WITH NFILT = 12

Piece of song duration (s)	Preprocessing time (s)	Average search time per song (s)	Standard deviation	Top	Accuracy (%)
30	31.0313	6.9129	1.5951	1	0
30	31.4687	11.7315	4.3668	5	0
30	31.2343	19.2569	4.1307	10	0
60	58.8281	5.7569	0.8015	1	0
60	60.3750	13.3693	3.8110	5	20
60	69.3906	14.4604	6.9438	10	20
120	101.2343	10.0245	0.6019	1	30
120	101.8906	16.2980	3.2247	5	40
120	114.1093	19.3399	6.0749	10	40

Firstly, after conducting a test, the results show that 30-second pieces of music give shallow search results, even not finding the right one. Secondly, 60-second tracks will have higher detection accuracy than 30-second tracks, but at Top 1, the accuracy still needs to improve. Finally, 120-second tracks give the highest accuracy compared to 30-second and 60-second tracks. When running with parameters $f = 120$ and $\text{nfilt} = 12$ when searching for music pieces, the accuracy is highest. Therefore, when searching for music tracks, the results will have the highest accuracy with the best parameter set of $f = 120$, $\text{nfilt} = 12$.

C. Testing Web-based Functionality

In order to help streamline the search with an intuitive interface, the song search website by music has been concocted to help users find the desired song more effortlessly. The interface is illustrated in Fig. 4.

Figure 4 is a website search result for the Top 1 song. The results are sorted by similarity from highest to lowest. The song with the most similarity will be displayed at the top. To demonstrate more plainly, in Fig. 5, with the desired selection results, the system returns five songs with the most similar melody to the searched song (DaoNuong.wav).

From Fig. 4 and Fig. 5, the intuitive interface and accuracy of the algorithm when operating on the web are clearly illustrated, providing the expected results and great potential for improvement, development, and producing of similar systems in the hereafter.

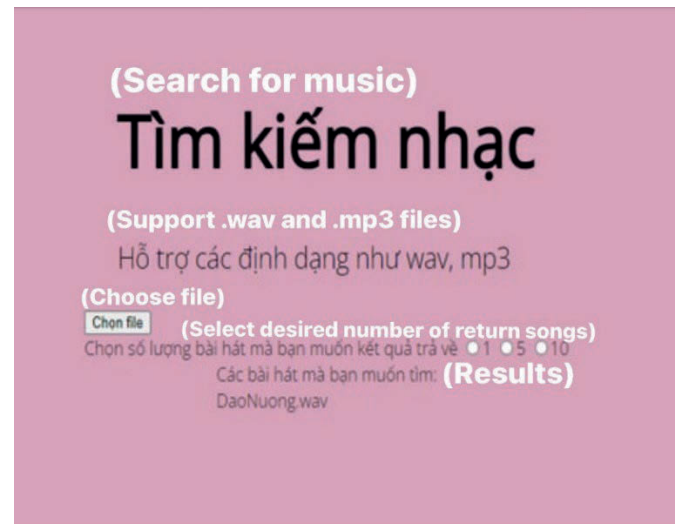


Fig. 4. The interface returns the desired song search results.



Fig. 5. Illumination of search results with the Top 5 most matching songs.

V. CONCLUSION

After researching and implementing the matter, “Instrumental Music Search with Approximate Nearest Neighbours” has obtained favourable results. Research has applied the ANN algorithm and MFCC feature extraction to perform song searches by harmony. It helps to choose the most appropriate parameter to give the most authentic results and the promptest search time. Besides, building a website that integrates algorithms makes it more straightforward for users to find songs. Additionally, the study has presented the dataset, ordeal scenarios, evaluation methods, and test consequences of using the algorithm to search for songs. Thereby, the most suitable parameters have been selected to search for songs quickly and with the highest possible exactness. The algorithm has been successfully developed on the web platform from the positive test outcomes, allowing users to perform song searches smoothly, quickly, and efficiently. Moreover, the study has also conducted training and evaluation of the most intuitive approach to the expected results.

CONFLICT OF INTEREST STATEMENT

All authors declare that they have no conflicts of interest.

REFERENCES

- [1] R. Ernst, "Music for life," *Music Educators Journal*, vol. 88, no. 1, pp. 47–51, Jul. 2001. <https://doi.org/10.2307/3399777>
- [2] C. Jordan, "When i'm 64: A review of instrumental music-making and brain health in later life," *Experimental Gerontology*, vol. 123, pp. 17–23, Aug. 2019. <https://doi.org/10.1016/j.exger.2019.05.006>
- [3] G. Johnson, D. Otto, and A. A. Clair, "The effect of instrumental and vocal music on adherence to a physical rehabilitation exercise program with persons who are elderly," *Journal of Music Therapy*, vol. 38, no. 2, pp. 82–96, Jul. 2001. <https://doi.org/10.1093/jmt/38.2.82>
- [4] L. Chaddock-Heyman, P. Loui, T. B. Weng, R. Weissappel, E. McAuley, and A. F. Kramer, "Musical training and brain volume in older adults," *Brain Sciences*, vol. 11, no. 1, Jan. 2021, Art. no. 50. <https://doi.org/10.3390/brainsci11010050>
- [5] A. Creech, S. Hallam, H. McQueen, and M. Varvarigou, "The power of music in the lives of older adults," *Research Studies in Music Education*, vol. 35, no. 1, pp. 87–102, Mar. 2013. <https://doi.org/10.1177/1321103x13478862>
- [6] K. L. Hyde, J. Lerch, A. Norton, M. Forgeard, E. Winner, A. C. Evans, and G. Schlaug, "The effects of musical training on structural brain development," *Annals of the New York Academy of Sciences*, vol. 1169, no. 1, pp. 182–186, Jul. 2009. <https://doi.org/10.1111/j.1749-6632.2009.04852.x>
- [7] R. Román-Caballero, M. A. Vadillo, L. J. Trainor, and J. Lupiáñez, "Please don't stop the music: A meta-analysis of the cognitive and academic benefits of instrumental musical training in childhood and adolescence," *Educational Research Review*, vol. 35, Feb. 2022, Art. no. 100436. <https://doi.org/10.1016/j.edurev.2022.100436>
- [8] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Apr. 2021. <https://doi.org/10.1007/s12525-021-00475-2>
- [9] M. Chapparband, M. G. Kulkarni, D. K. Sameeksha, A. V. Krishna, and A. Bhat, "Shruti detection using machine learning and sargam identification for instrumental audio," in *Advances in Speech and Music Technology. Advances in Intelligent Systems and Computing*, A. Biswas, E. Wennekes, T.P. Hong, and A. Wiczorkowska, Eds., vol. 1320. Springer, Singapore, 2021, pp. 145–156. https://doi.org/10.1007/978-981-33-6881-1_13
- [10] A. Huaysrijan and S. Pongpinigpinyo, "Deep convolution neural network for Thai classical music instruments sound recognition," in *2021 25th International Computer Science and Engineering Conference (ICSEC)*, Chiang Rai, Thailand, Nov. 2021, pp. 283–288. <https://doi.org/10.1109/icsec53205.2021.9684611>
- [11] W. Han, C.-F. Chan, C.-S. Choy, and K.-P. Pun, "An efficient MFCC extraction method in speech recognition," in *2006 IEEE International Symposium on Circuits and Systems*, Kos, Greece, May 2006. <https://doi.org/10.1109/iscas.2006.1692543>
- [12] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for MFCC feature extraction," in *2010 4th International Conference on Signal Processing and Communication Systems*, Gold Coast, QLD, Australia, Dec. 2010, pp. 1–5. <https://doi.org/10.1109/icsps.2010.5709752>
- [13] Y. Liu, Y. Yin, Q. Zhu, and W. Cui, "Musical instrument recognition by XGBoost combining feature fusion," *arXiv*, Jun. 2022. [Online]. Available: <https://arxiv.org/abs/2206.00901>
- [14] Y. H. Dandawate, P. Kumari, and A. Bidkar, "Indian instrumental music: Raga analysis and classification," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, India, Sep. 2015, pp. 725–729. <https://doi.org/10.1109/ngct.2015.7375216>
- [15] A. Ghosal, R. Chakraborty, B. C. Dhara, and S. K. Saha, "Song/instrumental classification using spectrogram based contextual features," in *CUBE'12: Proceedings of the CUBE International Information Technology Conference*, ACM Press, Sep. 2012, pp. 21–25. <https://doi.org/10.1145/2381716.2381722>
- [16] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *Computer Vision – ECCV 2008. ECCV 2008. Lecture Notes in Computer Science*, D. Forsyth, P. Torr, and A. Zisserman, Eds., vol. 5303. Springer, Berlin, Heidelberg, 2008, pp. 500–513. https://doi.org/10.1007/978-3-540-88688-4_37
- [17] S. M. Elghamrawy and S. E. Ibrahim, "Audio signal processing and musical instrument detection using deep learning techniques," in *2021 9th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, Alexandria, Egypt, Dec. 2021, pp. 146–149. <https://doi.org/10.1109/jac-ecc54461.2021.9691427>
- [18] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv*, 2015. [Online]. Available: <https://arxiv.org/abs/1511.08458>
- [19] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5–6, pp. 183–197, Jul. 1991. [https://doi.org/10.1016/0925-2312\(91\)90023-5](https://doi.org/10.1016/0925-2312(91)90023-5)
- [20] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306. <https://doi.org/10.1016/j.physd.2019.132306>
- [21] B. McFee *et al.*, "librosa/librosa: 0.10.0.post2," Mar. 2023. [Online]. Available: <https://zenodo.org/record/591533>
- [22] J. Lyons *et al.*, "jameslyons/python speech features: release v0.6.1," Jan. 2020. [Online]. Available: <https://zenodo.org/record/3607820>
- [23] E. Bernhardsson, "Annoy: Approximate Nearest Neighbors in C++/Python optimized for memory usage and loading/saving to disk," Python package version 1.17.3, Jun. 2023. [Online]. Available: <https://pypi.org/project/annoy/>
- [24] P. V. Toan, "Xay dung he thong tim kiem nhac bang am thanh voi Python va Approximate Nearest Neighbors," 2018. [Online]. Available: <https://viblo.asia/p/xay-dung-he-thong-tim-kiem-nhac-bang-am-thanh-voi-python-va-approximate-nearest-neighbors-924IjbXWIPM>
- [25] T. B. Arnold, "keras: R interface to the Keras deep learning library," *The Journal of Open Source Software*, vol. 2, no. 14, Jun. 2017, Art. no. 296. <https://doi.org/10.21105/joss.00296>



Hai Thanh Nguyen is a Lecturer of the College of Information and Communication Technology, Can Tho University, Vietnam. He received his Master's degree in Computer Science and Engineering from National Chiao Tung University, Taiwan, in 2014, and his PhD degree in Computer Science from Sorbonne University, France, in 2018. His current research includes bioinformatics, healthcare systems, recommendation systems, and applications using computer vision, machine learning, and data analysis.

E-mail: nhai.cit@ctu.edu.vn

ORCID ID: <https://orcid.org/0000-0002-1386-1390>



Linh Dan Vo is a student at the College of Information and Communication Technology, Can Tho University, Can Tho, Viet Nam. His primary areas of interest lie in machine learning, deep learning, and computer vision. He ardently pursues his passion for expanding knowledge in these fields, aspiring to make valuable contributions to technological advancements.

With a solid academic background and an unwavering dedication to research, he aims to engineer intelligent systems capable of enhancing diverse industries.

E-mail: danlinhvo1903@gmail.com



Thien Thanh Tran is a student at the College of Information and Communication Technology at Can Tho University, Can Tho, Viet Nam. His research interests revolve around machine learning, deep learning, and computer vision. He is dedicated to expanding knowledge in these fields and aims to contribute to technological advancements. With a solid academic foundation and a passion for research, he strives to develop intelligent systems that can improve various industries.

E-mail: trtthien2304@gmail.com