INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

A Note Based Query By Humming System using Convolutional Neural Network

Naziba Mostafa, Pascale Fung

The Hong Kong University of Science and Technology
Department of Electronic and Computer Engineering
Clear Water Bay, Kowloon, Hong Kong
nmostafa@connect.ust.hk, pascale@ece.ust.hk

Abstract

In this paper, we propose a note-based query by humming (QBH) system with Hidden Markov Model (HMM) and Convolutional Neural Network (CNN) since note-based systems are much more efficient than the traditional frame-based systems. A note-based QBH system has two main components: humming transcription and candidate melody retrieval.

For humming transcription, we are the first to use a hybrid model using HMM and CNN. We use CNN for its ability to learn the features directly from raw audio data and for being able to model the locality and variability often present in a note and we use HMM for handling the variability across the time-axis.

For candidate melody retrieval, we use locality sensitive hashing to narrow down the candidates for retrieval and dynamic time warping and earth mover's distance for the final ranking of the selected candidates.

We show that our HMM-CNN humming transcription system outperforms other state of the art humming transcription systems by $\sim 2\%$ using the transcription evaluation framework by Molina et. al and our overall query by humming system has a Mean Reciprocal Rank of 0.92 using the standard MIREX dataset, which is higher than other state of the art note-based query by humming systems.

Index Terms: query by humming, humming transcription, CNN, raw audio

1. Introduction

Query-by-humming is a content-based music retrieval method that can retrieve melodies using users' hummings as queries. This allows users to find melodies only by humming the tune and does not require any knowledge of its related metadata or even lyrics. Due to the convenience of a QBH system, it has received a great deal of attention from researchers in recent years. However, the accuracy and the efficiency of query by humming systems still have a lot of room for improvement.

The biggest challenges of a query by humming system include i) queries sung by users often vary from the actual melody in pitch, tempo etc. so the melodic similarity matching must be done at a more abstract level in order to get meaningful results, ii) background noise is often present in users' queries which also makes it harder to identify the melody correctly and iii) efficient retrieval methods must be used that can search through a database and retrieve the correct melody in as little time as possible. Therefore, methods used to retrieve the melody in this case need to be robust to noise and inaccuracies in the singing or humming which is very challenging, and, for the system to be practical, the entire system should be able to perform in real time.

A query or a melody is mainly represented using frame-based or note-based methods [1]. The frame-based methods use the extracted pitch to represent the melody and then use template-matching similarity measures such as DTW (Dynamic time warping) to measure similarity between the main melody and the query [2, 3]. These methods have high accuracy but are also slower and have higher complexity. The note-based methods extract and transcribe the note sequences from the hummed query [4, 5], and then compare them against the note sequences of the main melodies in the database to retrieve the melody closest to the query. The humming transcription part of note-based methods often lowers the overall accuracy of the system, but they are much more efficient since comparing note sequences has significantly lower complexity than comparing the melodies frame by frame. [1, 6]. In this paper, we focus on note-based methods since they are more practical for commercial use [1].

Several of the prominent humming transcription systems in the literature use Hidden Markov Models (HMM) with Gaussian Mixture Models (GMM) for transcribing the notes [5, 4, 7]. However, no attempt on using neural network for humming transcription has been made yet, even though works in related fields such as speech transcription [8], piano music transcription [9], singing melody identification [10] etc. showed to have much better results by incorporating deep neural networks (DNN).

We used a feature-based HMM-DNN model in our previous paper [14] to model the notes, where we used trial and error to find the most optimal features as there is no standard feature set for this task. However, since the accuracy of the model is completely dependent on the features chosen for most machine learning tasks and it is very difficult to find the most ideal set of features, some of the more recent works in machine learning have focused on using raw input data directly as opposed to features. This allows the model to learn themselves from the original source. Recent works in speech transcription[11], music classification [12], emotion and sentiment recognition [13] etc. got really good results when raw audio data was used as input to train the deep convolutional neural network.

Therefore, in this paper, we propose a hybrid Hidden Markov Model and Convolutional Neural Network (CNN) for humming transcription. The CNN model learns the features directly from raw audio data and we show this method performs better than when feature engineering is used. The HMM is used to model the temporal aspect of note transcription. After transcribing the notes using the HMM-CNN model, we use locality sensitive hashing to narrow down the candidates for retrieval and dynamic time warping and earth mover's distance for the final ranking of the selected candidates.

http://dx.doi.org/10.21437/Interspeech.2017-1590

2. Methodology

An overview of the overall query by humming system is given below in Figure 1. The system takes the hummed query as input. The notes of the query are transcribed using our humming transcription system. The transcribed query is then passed onto a candidate melody retrieval system, which compares the query against the melody database that consists of a list of pre-transcribed melodies. The output is the ranked list of melodies that are most similar to the input query.

Our humming transcription and candidate melody retrieval systems are explained further in Sections 3 and 4 respectively.

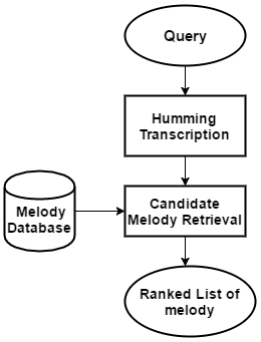

Figure 1: *Overview of a QBH system*

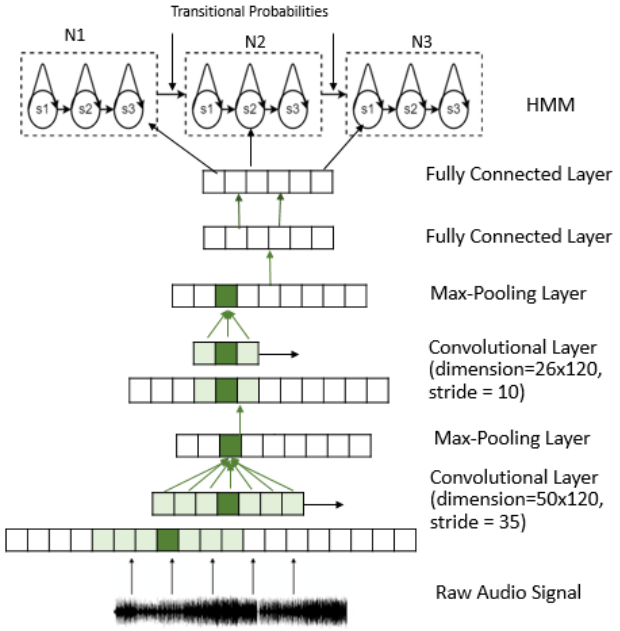

Figure 2: *Overview of the main model for the note transcription system, where N1,N2,N3 represents the notes and s1,s2,s3 represents the states of the HMM model*

3. Humming Transcription

The goal of the transcription system is to output the most likely sequence of notes, $N = n_1, n_2, ...n_k$ given the acoustic signal $A = a_1, a_2, a_n$. Therefore, we use the HMM model to maximize $P(N|A)$ [15]:

$$P(N|A) = \frac{P(A|N)P(N)}{P(A)} \tag{1}$$

- where $P(A|N)$ is the acoustic model, which captures the probability of observing a sequence of acoustic observations A given a note sequence N,
- $P(N)$ is the musicological model, which provides a prior probability for the note sequence N

The transcriber evaluates and combines both models through generating and scoring a large number of alternative note sequences during a complex search process. The main model used for transcription is given in Figure 2.

The acoustic model and the musicological model are described more in detail in Section 3.1 and 3.2 respectively.

3.1. Acoustic Modelling

The acoustic modelling is used to find the probability, $P(A|N)$ of observing an acoustic sequence given a note sequence. Similar to phoneme modelling in speech recognition systems [16], each note is represented by a 3-state Hidden Markov Model (HMM). The three states in the HMM represent the transitions between the fluctuations in the beginning of a note followed by the steady state in the middle and a decaying state in the end.

A Convolutional Neural Network (CNN) is used to model the posterior probability of each state of the HMM from the hummed query raw audio sample. We have chosen to use a CNN model for the task instead of a DNN model that we used previously [14] because a CNN model can be used to learn the features directly from raw audio data, which is significant since there is no standard feature set for this task. The main architecture of our model is shown in Figure 2. Our CNN model consists of a sequence of:

- convolutional layers, whose outputs q_j can be computed as:

$$q_j = \sigma(\sum_{(i=1)}^{I} o_i * w_{i,j} + b_j) \tag{2}$$

  where $*$ is the convolution operator, o_i represents the i-th input feature map, $w_{i,j}$ represents the weight matrix, b_j is the trainable bias attached to q_j and σ is the logistic sigmoid activation function.

- max-pooling layers, which are added on top of the convolution layers, and outputs the maximum function within each non-overlapping groups from the previously generated output vectors.

Our CNN model contains 2 convolutional layers, 2 max pooling layers and 2 fully connected layers. At first, the features are fed to the first convolutional layer and the dimension of this first layer is set to be 50x120 and the stride is set to be 35. Each column vector of the output matrix is the result of each moving step by multiplying the first layer input and the first layer matrix. This first convolutional layer acts as a feature extractor and the first max pooling layer behaves as a non-overlapping max-pooling function and each time it takes the maximum value of the adjacent columns horizontally.

The shape of the second convolutional layer matrix is 26 by 120 and the second stride is set to be 10. The second max-pooling layer takes the maximum value of all the output vectors horizontally from the output of the second convolutional

layer. Two final fully connected layers are then applied, which perform similarly to the fully connected Deep Neural Networks (DNN), followed by a softmax layer that computes the posterior probabilities for all the HMM states.

For this task, we trained notes in the range of 35-85 MIDI note numbers since this range covers all the notes used for human humming.

3.2. Musicological Modelling

The musicological model, $P(N)$, calculates the prior probability for a note sequence N. It is the equivalent of the language model used in speech recognition [16]. We use an existing algorithm [17] for this purpose, which uses musical keys and note bigrams to determine note transitions, since the musical key of a tune is important in determining note transitions.Knowing the musical key is important as some note sequences are more common than others in each key.

The model first estimates the key of the musical piece [7]. Then different note bigrams are defined for each key which are then used to calculate the note bigram probabilities.

4. Candidate Melody Retrieval

After getting the note transcription for each query, we need to retrieve the most similar melody from the database. We first convert the note sequence obtained into a vector form that can be used for measuring the similarity. Then we use a locality sensitive hashing method to get the candidate melodies most similar to the query. Finally we use a combination of dynamic time warping and earth movers distance to do the final ranking of the candidates and retrieve the candidate most similar to the query.

4.1. Note Sequence Conversion

When listening to a melody, we generally perceive how the pitches of successive notes relate to each other [18]. Therefore, instead of absolute notes, we convert the note sequences of the queries and the melody in the database in the form of relative notes and duration.

Each of the query and melody sequences in the database are represented as vectors in the form of $\mathbf{p} = ((R1, d1), (R2, d2), (R3, d3))$, where $R1, R2, R3$ represent the relative note sequence and $d1, d2, d3$ denote the duration of each note in seconds. For example, a sequence of MIDI notes $53, 53, 50, 54$ with duration $0, 0.5, 2, 1.5$ will be represented as $((0, 0), (0, 0.5), (-3, 2), (4, 1.5))$.

4.2. Locality sensitive hashing for narrowing down the candidates

We first narrow down the candidates for retrieval using the local sensitive hashing (LSH) algorithm [19], which uses sublinear search time over the database. We first create melodic segments from the melody database, which are then normalized to create pitch contours within a fixed-length time window. We then create an index which stores their positions in time within the database melodies, and identifiers that show the candidate melody from which the particular melodic segment has been extracted.

The similarity of melodic segments is measured using Euclidean distance between two pitch vectors p_i and p_j. The distance is given by:

$$||p_i - p_j|| = \sqrt{\sum_{m=1}^{M} |p_i(m) - p_j(m)|^2} \quad (3)$$

and is calculated in M-dimensional space.

For each pitch vector extracted from melodic segments of a query, we find similar segments in the index by searching for all the points to which the distance is less than a specified threshold. Instead of simply measuring the distance of the pitch vectors to all the vectors in the database, we use locality sensitive hashing to obtain a sublinear time complexity.

The LSH returns the most similar pitch contour and their distances to the query pitch vector as matches. The entire query and the candidate segment are then normalized both in pitch and time for distance calculation to get the entire pitch contour, which is ultimately used for finding the similarity for final ranking of candidates.

4.3. Final ranking of candidates

The final ranking of the candidates is done by using a slight alteration of the method used in [2], which uses a fusion of note-based EMD (Earth Mover's Distance) measure and a frame-based DTW (Dynamic Time Warping) measure. We chose to use a note-based DTW measure similar to [20] with the note-based EMD method [21]. Then we use a parallel voting strategy to rank the available candidates. using the final scoring function given below:

$$score(q_i, m_j) = \sum_{n=1}^{N} w(i) s(q_i, m_j) \quad (4)$$

where q_i represents the query sequence, m_j represents the melody sequence, $s(q_i, m_j)$ represents the distance score between the query and the melody and $w(i)$ represents the weights assigned to each distance score. The weights are assigned based on the overall accuracy of each of the systems. The melodies in the database are then ranked using their distance scores, so that the melody with the lowest distance in the database is ranked 1st and so on.

5. Experiments

5.1. Corpora

For training the CNN-HMM acoustic model, we used a corpus of 16 hours of humming data collected by us. We only collected humming data from the melodies in the 48 ground truth MIDI files from Roger Jang's MIREX corpus [1], whose note transcriptions are already available to us, which makes the humming dataset easier to annotate.

For training the musicological model, we used the ESAC database [2], which consists of 7055 transcribed melodies from different parts of the world.

For evaluation of the humming transcription system, we use the humming/singing evaluation dataset used in [22], which consists of melodies sung by adult and child untrained singers, which are manually transcribed.

For evaluation of the overall query by humming system, we use a corpus of 4431 queries from the MIR-QBSH corpus as used by MIREX. The queries are used to retrieve songs from a labelled melody database that consists of 48 ground truth MIDI

[1] http://www.music-ir.org/mirex/wiki

[2] http://www.esac-data.org/

files from MIR-QBSH corpus [3] with an additional 2000 files from ESSEN corpus [4] as used by MIREX.

5.2. Baseline Systems

As a baseline for the humming transcriptions system, we train a traditional CNN model with features. We first extract the features, consisting of pitch trackers, since previous works in humming transcription [7, 23] have always achieved the best results using pitch and other similar prosodic features. Since none of the currently available pitch extraction algorithms are completely accurate, we decided to use three of the best pitch extraction algorithms according to [24] as features to improve our systems accuracy, which includes the Autocorrelation-Leiwang [5], melodia [25] and pyin [26] algorithms.

The feature set is chosen empirically and all the features are derived using VAMP plugin [6] in sonic annotator tool [7] in overlapping 46.4 ms frames with 2.9 ms interval between the beginnings of successive frames.

The structure of the CNN model is similar to the one used for our model with raw audio. In this case, only the dimensions of the convolutional layers are 10x2 and 6x2 respectively and the strides are 4 and 1 respectively.

We also compare our humming transcription system with other state-of-the-art baseline systems used in [22].

We compare our overall query by humming systems with systems submitted to MIREX in the last three years.

5.3. Experimental setup and evaluation of the humming transcription system

We trained our acoustic model using the Kaldi-PDNN toolkit [27]. For training the CNN model, an initial learning rate of 0.04 is used which stays unchanged for 10 epochs. Then the learning rate is halved at each epoch until the cross-validation accuracy on a held-out set stops to improve. A momentum of 0.5 is used for fast convergence and the mini-batch size is set to be 256. Here, 10% of the training set is used as validation set to tune the hyper-parameters and determine the best network layout.

We decode the note transcription using the Kaldi decoder [8]. The transcription of our system is evaluated using the F-measure scores [28] of correct onset and pitch. The results are shown in Table 1.

We achieve an overall F-measure score of 0.55, which is 1.5% higher than the best state-of-the-art transcription system. Both the CNN models outperform the system based on HMM-GMM along with others. The CNN model with raw audio data gives the best results, which confirms our hypothesis that using raw audio data with convolutional neural network can provide a more optimal transcription system.

5.4. Evaluation of the overall retrieval system

Using the candidate melody retrieval system mentioned in Section 4, each query generates a list of most likely candidate melodies. The query by humming system is evaluated using

[3] http://www.music-ir.org/mirex/wiki

[4] http://www.esac-data.org/

[5] http://www.atic.uma.es/ismir2014qbsh/

[6] http://www.vamp-plugins.org/

[7] http://www.vamp-plugins.org/sonic-annotator/

[8] http://kaldi-asr.org/

Table 1: *Humming transcription evaluation result compared to other transcription algorithms*

Algorithm	F-Measure
Our CNN-HMM Model with Raw Audio	**0.55**
Baseline (CNN-HMM Model with Features)	0.525
Ryynanen (HMM-GMM)	0.49
Melotranscript (Auditory Model based System)	0.535
Gomez and Bonada (Tuning Frequency Method)	0.513
Molina et al	0.373

Table 2: *Evaluation result compared to other systems submitted in MIREX*

Algorithm	MRR
Our system	**0.919**
BS1(Frame-based)	0.8577
TYCX4(Combination of frame-based and note-based)	0.9281
ZH1(Frame-based)	0.8898
WHLX1(Note-based)	0.4687
LNL1(Combination of frame-based and note-based)	0.8550

Mean Reciprocal Ranking (MRR):

$$MRR = \frac{1}{|Q|} \sum_{(i=1)}^{|Q|} (1/rank_i) \tag{5}$$

Results are shown in Table 2. Our system performs better than all the other systems, in particular the pure note-based one, except TYCX4 [9], which is a partial frame-based system and therefore, has a much longer running time and higher algorithmic complexity.

6. Conclusion

In this paper, we have used Convolutional Neural Networks (CNN) with Hidden Markov Model (HMM) for note transcription, with a note-based retrieval method. We have shown that using a hybrid CNN-HMM model with raw audio data gives a $\sim$ 2% higher F-measure than any other humming transcription system including systems using HMM-GMM models and feature-based CNN model. We have also shown that our overall query by humming system has am MRR of 0.919, which is much better than other note-based methods and comparable to even the best frame-based systems in the literature.

7. Acknowledgements

This research was supported by CERG 16214415 of the Hong Kong Research Grant Council. We would also like to thank Dario Bertero from HLTC lab at HKUST for assistance with this paper.

8. References

[1] V. Kharat, K. Thakare, and K. Sadafale, "A survey on query by singing/humming," *International Journal of Computer Applications*, vol. 111, no. 14, 2015.

[2] L. Wang, S. Huang, S. Hu, J. Liang, and B. Xu, "An effective and efficient method for query by humming system based on multi-similarity measurement fusion," in *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*. IEEE, 2008, pp. 471–475.

[9] http://www.music-ir.org/mirex/abstracts/2015/TYCX4.pdf

[3] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis, "A comparative evaluation of search techniques for query-by-humming using the musart testbed," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 5, pp. 687–701, 2007.

[4] H.-H. Shih, S. S. Narayanan, and C. J. Kuo, "A statistical multidimensional humming transcription using phone level hidden markov models for query by humming systems," in *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, vol. 1. IEEE, 2003, pp. I–61.

[5] J. Shifrin, B. Pardo, C. Meek, and W. Birmingham, "Hmm-based musical query retrieval," in *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2002, pp. 295–300.

[6] J. Yang, J. Liu, and W. Zhang, "A fast query by humming system based on notes." in *INTERSPEECH*, 2010, pp. 2898–2901.

[7] M. P. Ryynänen and A. P. Klapuri, "Modelling of note events for singing transcription," in *ISCA Tutorial and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing*, 2004.

[8] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[9] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 121–124.

[10] F. Rigaud and M. Radenen, "Singing voice melody transcription using deep neural networks."

[11] D. Palaz, M. M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4295–4299.

[12] T. Park and T. Lee, "Musical instrument sound classification with deep convolutional neural network using feature fusion approach," *arXiv preprint arXiv:1512.07370*, 2015.

[13] D. Bertero, F. B. Siddique, C.-S. Wu, Y. Wan, R. H. Y. Chan, and P. Fung, "Real-time speech emotion and sentiment recognition for interactive dialogue systems."

[14] N. Mostafa, Y. Wan, U. Amitabh, and P. Fung, "A machine learning based music retrieval and recommendation system," in *Language Resources and Evaluation Conference, Portorož (Slovenia)*, 2016, p. 1.

[15] E. Trentin and M. Gori, "A survey of hybrid ann/hmm models for automatic speech recognition," *Neurocomputing*, vol. 37, no. 1, pp. 91–126, 2001.

[16] L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition," 1993.

[17] M. Ryynänen and A. Klapuri, "Transcription of the singing melody in polyphonic music." in *ISMIR*. Citeseer, 2006, pp. 222–227.

[18] J. H. McDermott and A. J. Oxenham, "Music perception, pitch, and the auditory system," *Current opinion in neurobiology*, vol. 18, no. 4, pp. 452–463, 2008.

[19] M. Ryynanen and A. Klapuri, "Query by humming of midi and audio using locality sensitive hashing," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 2249–2252.

[20] L. Cao, P. Hao, and C. Zhou, "Music radar: A web-based query by humming system," *Computer Science Department, Purdue University*.

[21] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, R. Van Oostrum *et al.*, "Using transportation distances for measuring melodic similarity." in *ISMIR*, 2003.

[22] E. Molina, A. M. Barbancho, L. J. Tardón, and I. Barbancho, "Evaluation framework for automatic singing transcription," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 567–572.

[23] J.-S. R. Jang, C.-L. Hsu, and H.-R. Lee, "Continuous hmm and its enhancement for singing/humming query retrieval." in *ISMIR*. Citeseer, 2005, pp. 546–551.

[24] E. Molina, L. J. Tardón, I. Barbancho, and A. M. Barbancho, "The importance of f0 tracking in query-by-singing-humming," 2014.

[25] J. Salamon, E. Gomez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *Signal Processing Magazine, IEEE*, vol. 31, no. 2, pp. 118–134, 2014.

[26] P. M. Brossier, "Automatic annotation of musical audio for interactive applications," Ph.D. dissertation, Queen Mary, University of London, 2006.

[27] Y. Miao, "Kaldi+ pdnn: building dnn-based asr systems with kaldi and pdnn," *arXiv preprint arXiv:1401.6984*, 2014.

[28] Wikipedia, "F-measure score," [Online; accessed 09-September-2016]. [Online]. Available: https://en.wikipedia.org/wiki/F1_score