IEICE TRANS. INF. & SYST., VOL.E101–D, NO.12 DECEMBER 2018

PAPER

Hidden Singer: Distinguishing Imitation Singers Based on Training with Only the Original Song

Hosung PARK[†a)], **Seungsoo NAM**[†b)], **Eun Man CHOI**[††c)], ***Nonmembers***, ***and*** **Daeseon CHOI**[†d)], ***Member***

SUMMARY Hidden Singer is a television program in Korea. In the show, the original singer and four imitating singers sing a song in hiding behind a screen. The audience and TV viewers attempt to guess who the original singer is by listening to the singing voices. Usually, there are few correct answers from the audience, because the imitators are well trained and highly skilled. We propose a computerized system for distinguishing the original singer from the imitating singers. During the training phase, the system learns only the original singer's song because it is the one the audience has heard before. During the testing phase, the songs of five candidates are provided to the system and the system then determines the original singer. The system uses a 1-class authentication method, in which only a subject model is made. The subject model is used for measuring similarities between the candidate songs. In this problem, unlike other existing studies that require artist identification, we cannot utilize multi-class classifiers and supervised learning because songs of the imitators and the labels are not provided during the training phase. Therefore, we evaluate the performances of several 1-class learning algorithms to choose which one is more efficient in distinguishing an original singer from among highly skilled imitators. The experiment results show that the proposed system using the autoencoder performs better (63.33%) than other 1-class learning algorithms: Gaussian mixture model (GMM) (50%) and one class support vector machines (OCSVM) (26.67%). We also conduct a human contest to compare the performance of the proposed system with human perception. The accuracy of the proposed system is found to be better (63.33%) than the average accuracy of human perception (33.48%).

key words: *singer authentication, autoencoder, neural network, artificial intelligence*

1. Introduction

Hidden Singer [1] is a television program of the JTBC broadcasting company in Korea. The objective of this show is to distinguish the original singer from a group of imitators. The four imitating singers are cast via audition and trained by vocal trainers before the show. During the show, the five singers, including the original, alternately sing a certain part of a song in hiding behind a screen. The subject song is famous enough that people already know the song. The audience and TV viewers determine who the original singer is by just listening their singing, relying on their memory of the original song. Even for a human, this mission is not easy, because the imitating singers are highly trained and very skilled. The occurrence of a correct answer is less than a 40% in most episodes.

We develop a computerized system to tackle this challenge (i.e. distinguishing the original singer from a group of imitating singers). The system is operated under the same conditions as the TV show. During training phase, the system learns only the original singer's song as the audiences have heard only original singer's song. During the testing phase, the songs of five singers are input to the system as the audience listens to the songs of imitating singers for the first time.

The system uses an autoencoder [2] as a classifier, which is a regenerating neural network. The autoencoder is trained by original data. During the testing phase, the trained autoencoder regenerates input (test) data as output. The autoencoder regenerates output that is very close to the input if the input data has similar patterns with previously trained data. By contrast, the regenerated output is as different from the input data as the input data is different from the trained data. Using this characteristic of the autoencoder, the system produces similarity scores for each candidate singer. The best scored singer is selected as the original.

Main contributions of this paper are as follows.

- To the best of our knowledge, this is the first attempt of authenticating an original singer among imitators. There has been related music retrieval works [3]–[12] that classify or identify singers. The purpose of those studies was usually to automate fast singer-retrieval. Therefore those studies distinguish different singers whose voices are clearly different from one another. In such a case, a human can easily differentiate the singers. Unlike those approaches, our work distinguishes very similar singers, including those who are highly trained to imitate the original singer. Moreover, our model can be trained with only the original singer's data, because there is no imitation data provided in advance.
- We conduct a human-machine contest that compares human performance to the performance of an artificial intelligence (AI). The average accuracy of the humans (33.48%) is lower than the accuracy of the proposed system (63.33%). Only the best result by a human shows a performance similar to the proposed system. This means that a well-trained AI can surpass hu-

Manuscript received April 19, 2018.
Manuscript revised July 18, 2018.
Manuscript publicized August 24, 2018.

[†]The authors are with Kongju National University, Kongju-si, 32588 Korea.
[††]The author is with Dongguk University, Seoul-si, 04620 Korea.
a) E-mail: hspark@kongju.ac.kr
b) E-mail: tnfok815@kongju.ac.kr
c) E-mail: emchoi@dgu.ac.kr (Corresponding author)
d) E-mail: sunchoi@kongju.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2018EDP7140

man perception. Conventionally, machines are meant to help automate certain processes or deal with massive standardized data, whereas they have not yet been able to replace human perception. After a contest between AlphaGo [13] and humans, this traditional view is changing, and the interest in AI is growing. Our results help us understand the capabilities and possible applications of this AI in other areas.

- We propose a 1-class authentication method that uses an autoencoder, in which only the subject model is made. In this problem, unlike in previous methods of artist identification, we cannot utilize multi-class classifiers and supervised learning. We therefore provide the performances of other possible 1-class learning algorithms, Gaussian mixture model (GMM) [14] and one class support vector machines (OCSVM) [15] including autoencoder. Because skillfully forged data is usually closer to the subject model, the authentication method requires high precision. The experiment results show that the autoencoder performs better (63.33%) than GMM (50%) and OCSVM (26.67%). The results can help understand which algorithm is more efficient for identifying the original singer from within a group including highly skilled imitators. The 1-class authentication method can be applied to any authentication problem where skilled forgery detection is important. In [16]–[20], a voice-based user authentication system was fooled by voice impersonation. Other authentication methods using human behavior characteristics, such as signature [21], gait [22], and gesture [23], are also vulnerable to skillfully forged works.

The rest of this paper is organized as follows. In Sect. 2, we introduce some background and related work. In Sect. 3, we explain our method. In Sect. 4, we describe experiments for evaluating our method and the formation and result of a human-machine contest. In Sect. 5, we discuss our experimental findings. Finally, in Sect. 6, we present our conclusions and future work.

2. Background and Related Works

Voice recognition addresses the conventional problem of identifying one speaker among a group of pre-registered speakers, or of verifying a testing voice as the claimed speaker. These methods generally exploit a Gaussian mixture model (GMM) [24], [25] for modeling the long-term distribution of spectral vectors. Early studies of voice recognition use maximum likelihood [24] and maximum a-posteriori [25] to train speaker-dependent GMMs. In the latter, the adaptation of a pre-trained universal background model (UBM) enables speaker-dependent GMMs called GMM super-vectors. The GMM super-vectors are successfully combined with a support vector machine (SVM) [26]. Meanwhile, some studies [16]–[18] of voice recognition show that voice impersonation leads to performance degradation. Specifically, intensive experiments in [18] show the impersonation increases false acceptance rates from close to 0% to between 10% and 60%. This result implies that distinguishing an imitated voice, especially by a skilled person, is more laborious than conventional voice recognition.

However, the singer classification problem is quite different from voice recognition for the following reasons. Singer classification first segments the audio signal into a singer's voice and a musical instrument. Then, it performs classification based on the singer's voice. Moreover, vocals have different features, such as frequency variation and singing techniques, compared to speech although they are made by the same person. Thus, different features should be dealt with in different ways.

Music classification issue can be classified into several categories: genre classification, mood classification, artist identification, instrument recognition, and music annotation [27]. Among those categories, artist identification is the most related to our work. Artist identification involves the recognition of an artist, a singer, or a composer. Only singer identification research is represented here. Studies have mostly focused on feature extraction methods, which extract a singer's voice from an audio signal. The extracted features include Mel-frequency cepstral coefficient (MFCC) features [3]–[5], linear frequency cepstral coefficient features [6], [7], harmonic features [6], cepstrum-based features [8], GMM super-vectors [9], and i-vectors [10], [12]. Segmenting a singer's voice and musical instrument is itself a classification problem and is outside the scope of this study.

Regarding extracted features, K-nearest neighbor (KNN) [28], SVM [29], and GMM [14] are the most popular classifiers used for artist identification. KNN is a multi-class classifier and uses training data directly for the classification of testing data. It classifies a testing instance by majority voting on the labels of the nearest instances in the training data set. SVM is a 2-class classifier based on the large margin principle. With training data, SVM finds the optimal separating hyperplane which maximizes the distance to the nearest training data points in both of two classes. A testing instance is classified by the hyperplane. Both KNN and SVM are applicable to singer identification, in which singers are distinguished. However, unlike conventional methods of singer identification, the training data is limited to only the original singer's song. Multi-class classifiers and supervised learning methods such as KNN and SVM cannot be utilized because the songs of imitators and the labels are not provided during the training phase. One-class and unsupervised learning algorithms such as GMM, OCSVM, and autoencoder can be applied to this problem. These approaches are used to build a model of the original singer and can be used for distinguishing the imitation singers. By using the model, GMM calculates the distance between original song and candidates and selects the candidate having the minimum distance. OCSVM, on the basis of the trained model, classifies each frame of test data into the model's class and out-of-class. If a frame is significantly different from the model, the frame is labeled as out-of-class. OCSVM selects

a candidate singer whose number of out-of-class frames is the least.

Note that these previous works [3]–[12] do not take into account imitating singers who are highly trained to imitate an original song. We provide the performances of possible 1-class learning algorithms, GMM and OCSVM including autoencoder in respect of the Hidden Singer problem. To the best of our knowledge, two studies have focused on the imitated song [30], [31]. The first one [30] studies songbirds imitating the songs of their parents as a tractable model to improve neural mechanisms. The other [31] classifies factors influencing vocal imitations and quantifies them via analysis of the pitch trajectories to help the future music education. In short, their purpose is not to distinguish imitating singers.

3. Proposed Method

In this section, we present our method for distinguishing an original singer among imitators.

3.1 System Overview

The overall system architecture is shown in Fig. 1. In the training phase, the original song is segmented manually. The features of the original singer's voice are then extracted. After normalization processing, the data is used as training data and an original singer's model is made. As there is only the original singer's song data in training phase, an unsupervised learning technique is used by autoencoder. In other words, there is no data for building others or a background model.

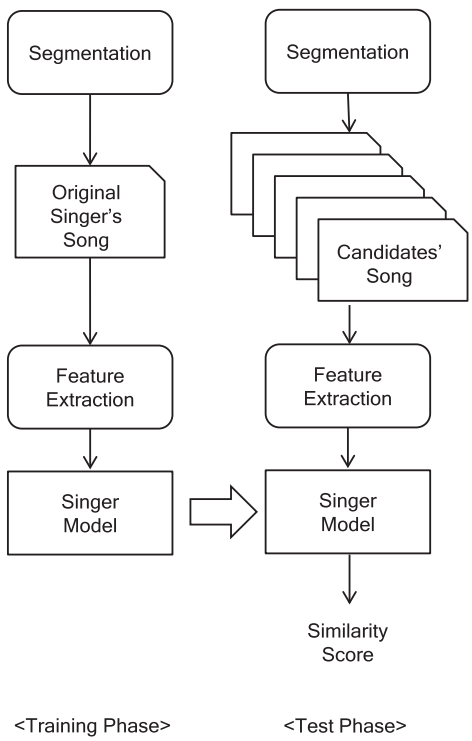


Fig. 1 Overall system architecture.

In the test phase, songs by five candidates consisting of an original singer and four imitating singers are extracted. Here, the song is same as the training song. However, the original singer again sings the song. Therefore, song data of the original singer is now different from the song that is trained. Feature extraction and normalization processes are the same as in the training phase. Five candidates' data are input to the singer model. Then, the singer model produces the similarity scores of each of the candidate's song. The best scored song is selected as the original singer.

3.2 Preprocessing

To make input data for the model, the song is segmented, features are extracted from the segmented sections, and features are normalized.

3.2.1 Song Segmentation

In the Hidden Singer TV show, five candidate singers sing a song in a relay fashion. Therefore, vocal sections of candidate singers must be segmented, and each singer's section is used as test data. The preludes and interludes also need to be trimmed-out for authentication accuracy. These vocal detection and trimming preludes and interludes are other research issues altogether. There have been some previous studies on this problem [4], [5], [8]. However, we do not apply the results of these studies to evaluate the system's ability to distinguish voices. Moreover, segmenting song sections sung by different imitating singers is actually a problem of classifying the imitating singers. Therefore, we manually segment a song into five sections. In the TV show, number indicators show who is singing a given section among the candidates. It is easy for a human to segment the song while watching the TV show. We also do not separate vocal and accompaniment because the accompaniment of test data is the same as the original recorded music. In the show, only existing instrumental versions are played back. Separating vocals from the instrumental part of the song is a sensitive process and it may be better to avoid the risk that the process damages the vocal data in this case.

3.2.2 Feature Extraction

Each section of songs is stored in a .wav file. To obtain the numeric features of candidates' voices, the MFCC is extracted from each song section. MFCC is the short-term cepstral representation of speech signal in the Mel scale [32]. It has been largely used to extract unique characteristics of a speaker's voice during speaker identification [24] and to address speaker verification problems [25], [26]. We carry out MFCC extraction by following the procedures in [6]. The overall procedure is shown in Fig. 2 and is described as follows.

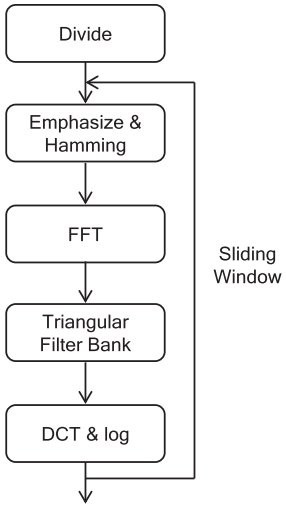


Fig. 2 Feature extraction procedure.

1. Divide .wav data into a time window of 5 ms. For example, with wave file of 44,000-Hz sampling rate, the time window includes 256 frames.
2. Pre-emphasize the signal.
3. Apply a Hamming window to correct the discontinuity at the start and ending samples of the frames.
4. Using a Fast Fourier transform to compute the spectrum amplitude of each window.
5. Filter the signal in the spectral domain with a triangular filter-bank with 40 filters which are approximately linearly spaced on the Mel scale and have equal bandwidth in the Mel scale.
6. Compute the discrete cosine transform of the log-spectrum.
7. Slide the window by 2 ms and repeat steps 2–6.

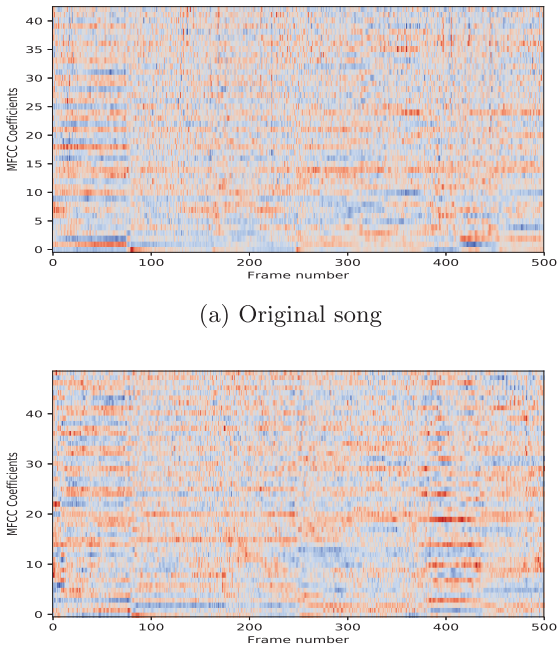
The output will be several vectors, x, where x_i represents the i^{th} MFCC for each time window. After calculating MFCC of all input data, the cepstral mean is subtracted from each vector. This helps remove channel bias and intra-singer variability, as mentioned in [4]. Figure 3 shows the extracted MFCCs of original song and the imitating singer. The two MFCCs are only small parts; however, we can deduce that they show similar patterns.

3.3 Building Singer Model with Autoencoder

Using the extracted features of the original singer's song, a singer model is made. We propose a voice model using an autoencoder [2], which is the kind of neural network that reproduces input data as its output, where the output is similar to input data as the input is similar to the trained data. An autoencoder consists of two parts: the encoder and the decoder, as shown in Fig. 4.

We denote a set of x of a singer as $\mathbf{X}$. When the autoencoder has one hidden layer, the encoder is defined as follows.

$$\mathbf{y} = \sigma_1(W\mathbf{X} + b), \tag{1}$$



(a) Original song

(b) Imitating singer

Fig. 3 MFCCs in frames.

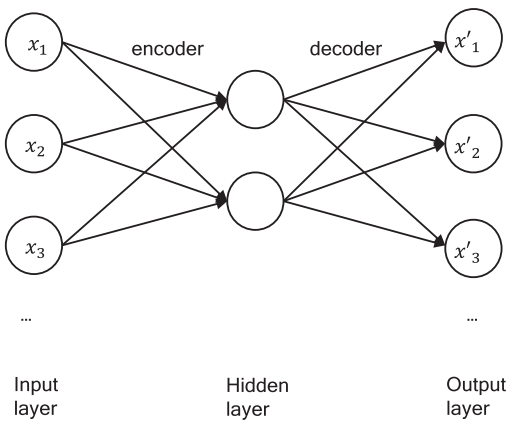


Fig. 4 Autoencoder construction.

where σ is an element-wise activation function, such as a *sigmoid* function, W is a weight matrix, and b is a bias vector. After encoding, y is mapped onto $\mathbf{X}'$ by the following decoder function.

$$\mathbf{X}' = \sigma_2(W'\mathbf{y} + b') \tag{2}$$

The loss function, $\mathcal{L}$ is defined as follows.

$$\begin{aligned}\mathcal{L}(\mathbf{X}, \mathbf{X}') &= \|\mathbf{X} - \mathbf{X}'\|^2 \\ &= \|\mathbf{X} - \sigma_2(\mathbf{W}'(\sigma_1(\mathbf{W}\mathbf{X} + \mathbf{b})) + \mathbf{b}'\|^2\end{aligned} \tag{3}$$

Training the autoencoder is a procedure of finding a W that minimizes $\mathcal{L}$ to build a better reconstruction network.

3.4 Evaluating Difference Using a Trained Autoencoder

In this section, we describe a difference measure, *Diff*,

for distinguishing an original singer from imitating singers. *Diff* is defined as the summation of the differences between the input song frame of a candidate singer, $\mathbf{x}_j$, and the output of autoencoder, $\mathbf{x'}_j$, for all frames, as follows.

$$Diff = \frac{1}{m}\sum_{j=1}^{m}(\mathbf{x}'_j - \mathbf{x}_j)^2 \quad (4)$$

where m is the number of frames of test data. The candidate singer whose song's *Diff* is smallest (i.e., the best case) is selected as the original singer's.

4. Experiments

We conduct two kinds of experiments: a machine contest and a human contest. The results are respectively shown in Sects. 4.3 and 4.4. In the machine contest, we compare three 1-class learning algorithms: autoencoder, GMM [14], and OCSVM [15]. Human perception is also compared with the learning algorithms by the human contest.

4.1 Experiment Data

We select 30 contest songs as a data set from 30 episodes. Original singers consist of 19 males and 11 females. Table 1 lists the 30 contest songs. The data set is composed of two types: training data and test data. The training data refers to the original recorded music, and the test data refers to the songs of candidates on the show. We gather the original song files in an online music market and use the files as training data. For each contest song, the original singer's model is built by using the training data. The test data set is manually extracted from the 30 episodes. Each test data includes the five candidates' songs, and we tag the original singer among them.

Table 1 The list of contest songs.

	Singer	Title
1	Lena Park	I hope it would be that way now
2	Sikyung Sung	you made me impressed
3	Kwan-woo Jo	Swamp
4	Sooyoung Lee	Grace
5	Yoon-jung Jang	Flower
6	Sangmin Park	Sunflower
7	Jongkuk Kim	Standstill
8	Vibe	Drinking
9	Gunmo Kim	Love is gone
10	Eunmi Lee	Into the memories
11	Seongmo Jo	To Heaven
12	Beomsu Kim	Appearing
13	Hyeonmi Ju	Crush
14	IU	Good day
15	Jinyoung Park	Honey
16	Kangsuk Kim	My song
17	Sunhee Lee	Destiny
18	Fly to the sky	Even though my heart hurts
19	JinAh Tae	Love is, not for everyone
20	Sooni In	Dream of goose
21	Jongshin Yoon	Rebirth
22	Taewoo Kim	High high
23	BoA	No.1
24	Buzz	Thorn
25	Chanwhee So	A wise choice
26	Jungmin Kim	Sad promise
27	Yeonwoo Kim	Is it still beautiful
28	Gummy	We should've been friends
29	Jinseop Byeon	Being alone
30	Jiyoung Baek	Don't forget

Before feature extraction, we perform low frequency filtering for all song data to get more dynamics for the essential frequencies. An effective cut-off frequency depends on the singer's gender, because singing voices have different frequency ranges, per the gender. Thus, we get rid of frequencies lower than 75 Hz for male singers, and those below 150 Hz for female singers. The cut-off frequency is configured by a performance experiment, shown in Fig. 5. The performance is represented by percentages, unlike other graphs, because the number of male singers is different from that of female singers.

The average length of the original singer's song, after segmentation, is about 182 s. Average length of each candidate's song is about 14 s. These are sampled in 5-ms windows of 2-ms sliding steps at a 44-KHz sampling rate. We could, therefore, obtain 110K frames for 1 s of song. Thus, the average size of training data is about 20M frames; the average size of each test data is 1.54M frames. The number of dimensions of extracted features is 49. The time window size is also configured by a performance experiment, as shown in Fig. 6. This result is different from voice recognition studies, which usually exploit a near-20-ms window. The remaining experimental details, for both experiments, follow parameters of the Sects. 4.2 and 4.3 in the best case.

4.2 Experiment Setup

Machine learning models are implemented using the Theano [33] library, which is a well-known open source machine learning library. Each subject model is built on the basis of the original singer's song. To find the best result, the experiment is repeated with different parameters. Tables 2, 3, and 4 show the major parameter ranges of autoencoder, GMM, and OCSVM respectively. In case of autoencoder, for example, a five-layer autoencoder can consist of an input

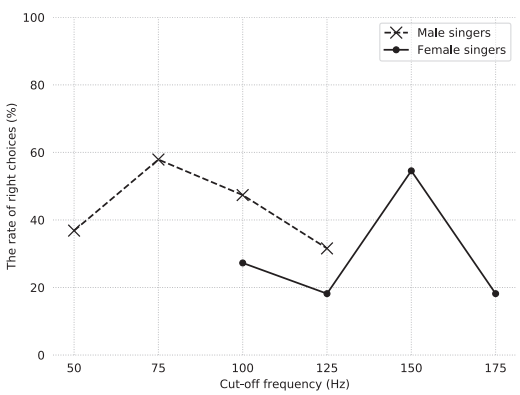


Fig. 5 Effect of low-frequency filtering.

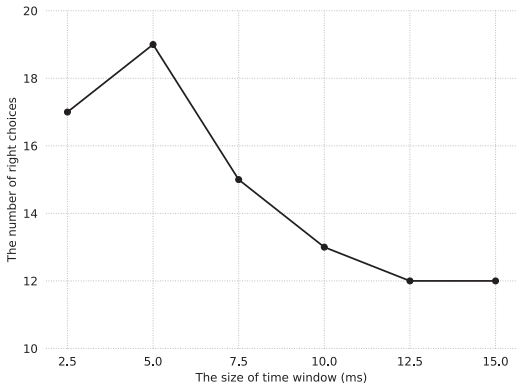


Fig. 6 Effect of time window size.

Table 2 Autoencoder parameters.

Parameters	Values
# of layers	3, 4, 5, 6
# of nodes (each hidden layer)	15, 20, 25, 30, 35, 40
Epoch	500, 600, 700, 800, 900

Table 3 GMM parameters.

Parameters	Values
n_components	1–10 (every 1)
tol	0.0005–0.005 (every 0.0005)

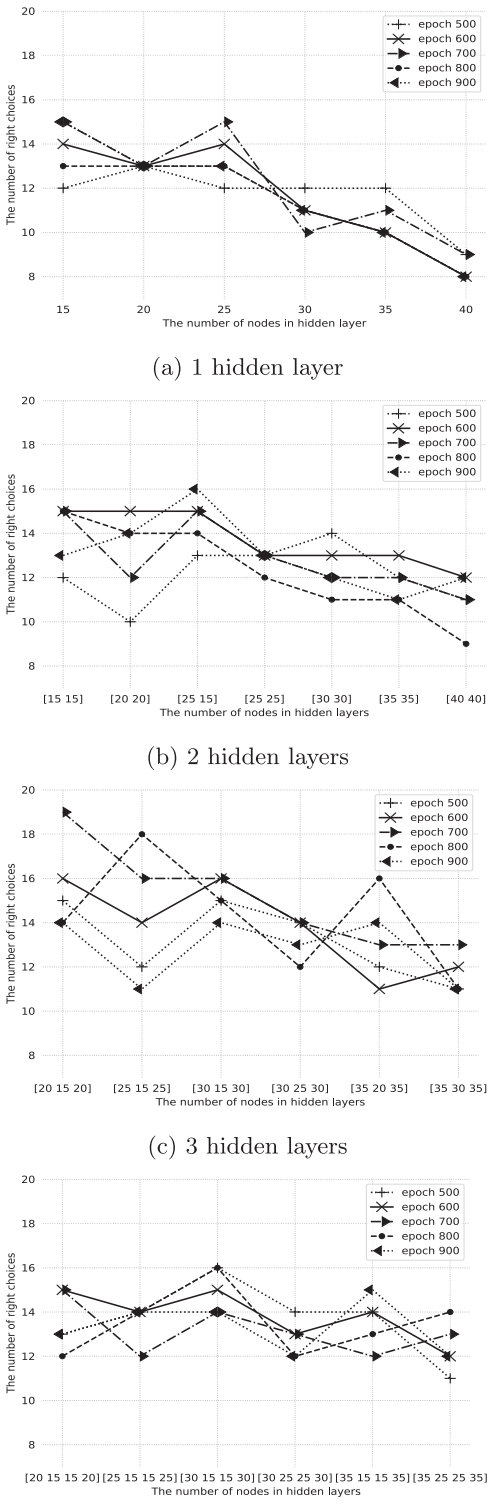
Table 4 OCSVM parameters.

Parameters	Values
nu	0.05–0.95 (every 0.05)
gamma	0.00005–0.0001 (every 0.00001)

layer, three hidden layers of 30, 25, and 30 nodes each, and an output layer. The number of nodes for an input layer and an output layer is fixed to 49 because the extracted features of the training data have 49 dimensions. The training epoch depends on the size, amount, and type of training data. In case of GMM, *n_components* refers to the number of mixture components and *tol* refers to the convergence threshold. We use the Kullback-Leibler distance [14] for calculating the difference between the original song and candidate songs. In case of OCSVM, *nu* refers to the upper bound on the fraction of training errors and should be in the interval (0, 1], and *gamma* refers to the kernel coefficient.

4.3 Results of Machine Contest

In this subsection, we describe the performances of the three learning algorithms: auto-encoder, GMM, and OCSVM. As mentioned in previous sections, in this Hidden Singer problem, we cannot utilize multi-class classifiers and supervised learning unlike previous studies used for artist identification. We therefore apply well-known, 1-class, and unsupervised learning algorithms. We describe the results of each algorithm according to its parameters, and then summarize the best results at the end of this sub-section. All results are represented by the number of right choices from among 30 tests (i.e., 30 contest songs). For readability, we ignore some unimportant results.



(a) 1 hidden layer

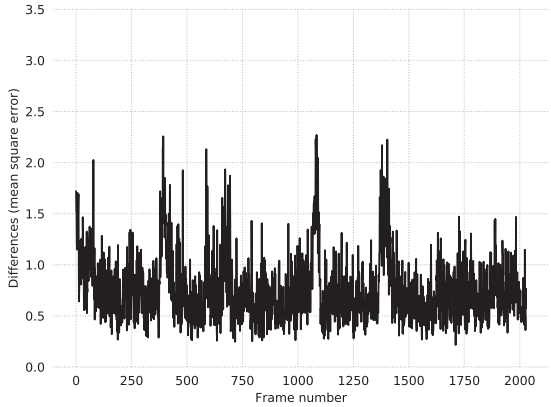
(b) 2 hidden layers

(c) 3 hidden layers

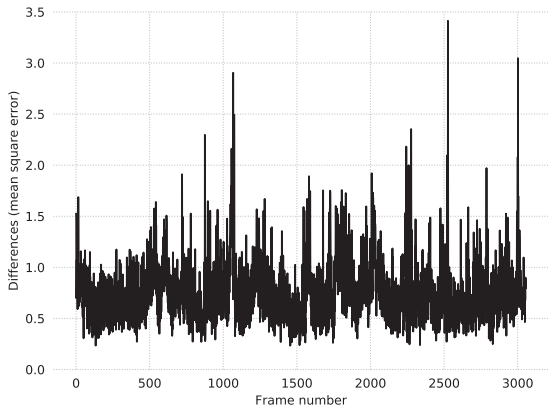
(d) 4 hidden layers

Fig. 7 Results of autoencoder according to the parameters.

Figure 7 shows the number of correct choices from autoencoder based method, with respect to the autoencoder pa-



(a) Original singer



(b) Imitating singer

Fig. 8 Differences between input and output of autoencoder.

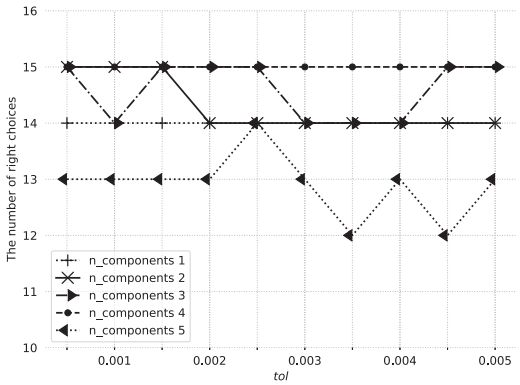


Fig. 9 Results of GMM according to the parameters.

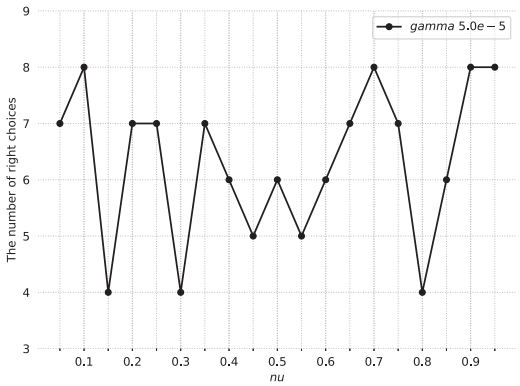


Fig. 10 Results of OCSVM according to the parameters.

rameters: number of layers, number of nodes for each hidden layer, and epoch. The highest number of correct answers is 19 (63.33%) when the parameters are five layers; three hidden layers of 20, 15, and 20 nodes each; and 700 epochs.

Figure 8 shows examples of the differences between input and output of autoencoder frame by frame. Input in Fig. 8 (a) is a song of the original singer, and input in Fig. 8 (b) is a song of one of the imitating singers. It is difficult to compare them with the naked eye; however, we can see that the big differences of the imitating singer are more than the original singer. In comparison with the music score, we also infer that significant differences usually appear in parts where the vocals are emphasized. In Fig. 8 (b), the three parts in which the differences exceed 2.5 are all high-pitched or stressed vocal sections.

Figure 9 shows the number of correct choices obtained from the GMM-based method, according to the parameters: *n_components* and *tol*. We exclude the results when *n_components* is more than 6, because they lead to worse performances. Similar to autoencoder, GMM builds a singer model with the original data and uses the model for calculating the difference from the test data. However, unlike autoencoder, GMM does not regenerate test data and just compares it with the original model. GMM predicts a candidate singer as the original singer if the difference (also called error or distance) is the smallest. The highest number of correct answers is 15 (50%), mostly when the *n_components* is 2, 3, 4. The parameter *tol* does not significantly influence the performance compared to the *n_components*.

Figure 10 shows the number of correct choices obtained from the OCSVM-based method, according to the parameter *nu*. The parameter *gamma* is fixed as 5.0e−5 because it has negligible influence on the results of the experiment. Unlike autoencoder and GMM, OCSVM does not directly calculate the difference between the model and test data. OCSVM, on the basis of the trained model, classifies each frame of test data into the model's class and out-of-class. If a frame is significantly different from the model, the frame is labeled as out-of-class. Consequently, OCSVM selects a candidate singer who's the number of out-of-class frames is the least as the original singer. Therefore, OCSVM is appropriate for finding the different portions among test data, but fundamentally offers lower precision than the other two algorithms in our experiment. The best case is 8 (26.67%).

Table 5 shows the best result of each 1-class machine learning algorithm. We can see that autoencoder demonstrates the best performance in machine contest.

Table 5 The best results of 1-clsss learning algorithms.

Algorithm	# of right choices	Percentage
Autoencoder	19	63.33%
GMM	15	50%
OCSVM	8	26.67%

Fig. 11 Human contest of Hidden Singer.

Table 6 Results of human contest.

	# of right choices	Percentage
The best	17	56.67%
The worst	4	13.33%
Average	10.05	33.48%

4.4 Results of Human Contest

To compare the performance of the proposed method with human perception, we hold a hidden singer contest. Figure 11 shows the human contest scene. Twenty-four people try to determine the original singer under the same conditions as the proposed system. In other words, they listen to an original song and then listen to five candidates sung by one original singer and four imitators. They must select the one singer they believe is the original. They repeat this test for 30 songs.

Table 6 shows the results of the human contest. The average number of correct answers is 10.05 (33.48%). Only the best result by a man is 19 (63.33%), the same level as the proposed method. It is not easy for a human to distinguish highly trained, very skilled imitation singers. Note that it is not difficult for a human to distinguish one singer from another, in most cases. The results show that a well-trained AI can surpass human perception in certain problems.

5. Discussion

We found efficient parameters such as cut-off frequency, time window size, the number of layers, the number of nodes in each hidden layer, epoch, and training data configuration, via repeated experiments. However, these parameters are not the global optimum, because numerous tries are needed to find it. Therefore, we found a local optimum for a parameter, fixed it, and then found another local optimum for another parameter. To approximate the global optimum as much as possible, if an unusual result is found, we return to the related parameters and repeated the experiment.

The accuracy of a male singer is higher than that of female singer for both human and machine contests. We cannot identify the precise cause for this result. It might be caused by the different frequency ranges, the characteristics of the sample singers, or something else. It will be interesting to pose a hypothesis about this result.

The current song of an original singer is not the same as the original song on the album. The original song is recorded in the best of cases a couple of years prior (20 years, worst case). Moreover, imitating singers are trained and skilled based on the original song. They observe characteristics that people sensitively distinguish the original singer. Additionally, they imitate their characteristics and even stressed them. People, therefore, are prone to feel that imitators are the original singers. Machines, however extract features of the original voice, itself, as well as the characteristics mentioned above. This could be the most different aspect from human perception, leading to the results in which machines win the contest. Machines also can emphasis the proper features and minimize indecent features during the learning phase.

Our experiments have some limitations. As mentioned above, five candidate singers sing a song in a relay fashion. Therefore, each singer's part is not long enough to draw conclusions. Moreover, candidate singers do not sing the same parts. This limitation in the data may lead to negative effects on the performance. For example, a singer who sings the high-pitched or stressed vocal sections may be at a disadvantage. Note that significant differences usually appear in parts in which the vocals are emphasized in Fig. 8. We expect that the proposed method can compare singers more accurately if each singer sings the same parts. For the experiments, the best case is that each singer sings the whole song.

6. Conclusion and Future Works

In this study, we developed a computerized system for distinguishing an original singer from a group of well trained and highly skilled imitation singers. Autoencoder, a 1-class classifier, was trained with only songs by the original singer, and we sought appropriate training data configuration and parameters for the purpose. The experiment results show that autoencoder performs better (63.33%) than other 1-class learning algorithms: GMM (50%) and OCSVM (26.67%). We also conducted a human contest to compare the performance of the proposed method to human perception. The accuracy of finding an original singer was 63.33%: better than the average accuracy of humans (33.48%). Experimental results imply that the well-trained AI can surpass human perception in certain problems, such as danger recognition, forgery detection, and anomaly detection. We expect that our study will help people understand the ability and possible applications of AI in various areas.

Our future work will apply this distinguishing imitation method to other authentication problems, where skilled

forgery is an important issue. We are focusing on the authentication of masterpieces and other artworks. Distinguishing art forgeries is typically entrusted to a few specialists; people have no option but to trust them. If an AI attains a sufficient accuracy in this field, it will be a great help to the people and even to the specialists. On the other hand, studies [34], [35] recently introduced a security issue about evasion attacks that use forged data. We expect that our method might prevent evasion attacks by detecting the forged data beforehand. For this purpose, we will analyze the evasion attacks and modify our system, including the feature extraction and model configuration.

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (2016-0-00173, Security Technologies for Financial Fraud Prevention on Fintech) and the research program of Dongguk University, 2015.

References

[1] Hidden Singer, a television show of the JTBC in Korea, Available online: http://tv.jtbc.joins.com/hiddensinger4 (season 4).
[2] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," Proc. ICML Workshop on Unsupervised and Transfer Learning, Edinburgh, Scotland, pp.37–49, June 2012.
[3] B. Whitman, G. Flake, and S. Lawrence, "Artist detection in music with minnowmatch," Proc. IEEE Workshop Neural Networks for Signal Processing, Falmouth, MA, USA, pp.559–568, Sept. 2001.
[4] A.L. Berenzweig, D.P.W. Ellis, and S. Lawrence, "Using voice segments to improve artist classification of music," Proc. Int. Conf. Virtual, Synthetic, and Entertainment Audio, Espoo, Finland, June 2002.
[5] W.-H. Tsai and H.-M.Wang, "Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals," IEEE Trans. Audio, Speech, Language Process., vol.14, no.1, pp.330–341, Jan. 2006.
[6] Y.E. Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," Proc. Int. Conf. Music Information Retrieval (ISMIR), Paris, France, Oct. 2002.
[7] J. Shen, J. Shepherd, B. Cui, and K.-L. Tan, "A novel framework for efficient automated singer identification in large music databases," ACM Trans. Inf. Syst., vol.27, no.3, Article No.18, May 2009.
[8] T.L. Nwe and H. Li, "Exploring vibrato-motivated acoustic features for singer identification," IEEE Trans. Audio, Speech, Language Process., vol.15, no.2, pp.519–530, Jan. 2007.
[9] C. Charbuillet, D. Tardieu, G. Peeters, "GMM supervector for content based music similarity," Proc. Int. Conf. Digital Audio Effects, Paris, France, pp.425–428, Sept. 2011.
[10] H. Eghbal-Zadeh, M. Schedl, and G. Widmer, "Timbral modeling for music artist recognition using i-vectors," Proc. European Signal Processing Conf. (EUSIPCO), Nice, France, pp.1286–1290, Aug. 2015.
[11] H. Fujihara, M. Goto, T. Kitahara and H.G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre similarity-based music information retrieval," IEEE Trans. Audio, Speech, Language Process., vol.18, no.3, pp.638–648, Feb. 2010.
[12] H. Eghbal-Zadeh and W. Gerhard, "Noise robust music artist recognition using i-vector features," Proc. Int. Conf. Music Information Retrieval (ISMIR), New York City, USA, Aug. 2016.
[13] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershevam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature, vol.529, no.7587, pp.484–489, Jan. 2016.
[14] J.H. Jensen, D.P. Ellis, M.G. Christensen, and S.H. Jensen, "Evaluation of distance measures between Gaussian mixture models of MFCCs," Proc. Conf. Int. Society for Music Information Retrieval (ISMIR), Vienna, Austria, pp.107–108, Sept. 2007.
[15] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation, vol.13, no.7, pp.1443–1471, July 2001.
[16] N.W. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," Proc. Conf. Int. Speech Comm. Association (Interspeech), Lyon, France, pp.925–929, Aug. 2013.
[17] Z. Piotrowski and P. Gajewski, "Voice spoofing as an impersonation attack and the way of protection," Journal of Information Assurance and Security, vol.2, no.3, pp.223–225, 2007.
[18] Y.W. Lau, D. Tran, and M. Wagner, "Testing voice mimicry with the YOHO speaker verification corpus," Proc. Int. Conf. Knowledge-Based Intelligent Inf. and Eng. Systems (KES), Melbourne, Australia, pp15–21, Lecture Notes in Computer Science, vol.3684. Springer, Berlin, Heidelberg, 2005.
[19] L. Li, Y. Chen, D. Wang, and T.F. Zheng, "A study on replay attack and anti-spoofing for automatic speaker verification," Proc. Interspeech, Stockholm, Sweden, pp.92–96, Aug. 2017.
[20] Z.K. Anjum and R.K. Swamy, "Spoofing and countermeasures for speaker verification: A review," Proc. Int. Conf. Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, March 2017.
[21] A. Buriro, B. Crispo, F. Delfrari, and K. Wrona, "Hold and sign: A novel behavioral biometrics for smartphone user authentication," Proc. IEEE Symposium on Security and Privacy Workshops (SPW), San Jose, CA, USA, pp.276–285, May 2016.
[22] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor," Journal of Computers, vol.1, no.7, pp.51–59, Oct./Nov. 2006.
[23] N. Sae-Bae, N. Memon, K. Isbister, and K. Ahmed, "Multitouch gesture-based authentication," IEEE Trans. Inf. Forensics Security, vol.9, no.4, pp.568–582, Jan. 2014.
[24] D.A. Reynolds and R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Trans. Speech Audio Process., vol.3, no.1, pp.72–83, Jan. 1995.
[25] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," Digital Signal Processing, vol.10, no.1-3, pp.19–41, Jan. 2000.
[26] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," IEEE Signal Process. Lett., vol.13, no.5, pp.308–311, April 2006.
[27] Z. Fu, G. Lu, K.M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," IEEE Trans. Multimed., vol.13, no.2, pp.303–319, April 2011.
[28] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol.13, no.1, pp.21–27, Jan. 1967.
[29] B.E. Boser, I.M. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," Proc. 5th ACM Conf. Computational Learning Theory, Pittsburgh, PA, USA, pp.144–152, 1992.
[30] Y. Mandelblat-Cerf and M.S. Fee, "An automated procedure for evaluating song imitation," PLoS ONE, vol.9, no.5, e96484, May 2014.
[31] J. Dai and S. Dixon, "Analysis of vocal imitations of pitch trajectories," Proc. Int. Conf. Music Information Retrieval (ISMIR), New York City, USA, Aug. 2016.
[32] Z. Fang, Z. Guoliang, and S. Zhanjiang, "Comparison of different

implementations of MFCC," Journal of Computer Science and Technology, vol.16, no.6, pp.582–589, Nov. 2001.
[33] Theano 1.0 release, Available online: http://deeplearning.net/software/theano/ (accessed on 17 Dec. 2017).
[34] F. Marra, G. Poggi, F. Roli, C. Sansone, and L. Verdoliva, "Counter-forensics in machine learning based forgery detection," Proc. Conf. Media Watermarking, Security, and Forensics, San Francisco, CA, USA, Feb. 2015.
[35] K. Sunil, D. Jagan, and M. Shaktidev, "DCT-PCA based method for copy-move forgery detection," ICT and Critical Infrastructure: Proc. Annual Convention of Computer Society of India, vol.2, pp.577–583, 2014.

Hosung Park received the B.S., M.S., and Ph.D. degrees in computer engineering from Chungnam National University, Korea, in 2008, 2010, and 2014, respectively. He is currently a research associate at Department of Medical Information, Kongju National University, Korea. His research interests include information security and identity management.

Seungsoo Nam received the B.S. degree in applied mathematics from Kong-ju National University, Korea, 2012, the M.S., and Ph.D. degrees in convergence science from Kong-ju National University, Korea, in 2014 and 2018, respectively. He is currently a postdoctoral research associate at Department of Convergence Science, Kong-ju National University, Korea. His research interests include information security and biometric authentication.

Eun Man Choi received the B.S. degree in computer science from Dongguk University, Korea, in 1982, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1985, and the Ph.D. degree in computer science from Illinois Institute of Technology, Chicago, in 1993. He is currently a Professor at Department of Computer Science and Engineering, Dongguk University, Korea. His research interests include software design, software testing, and security programming.

Daeseon Choi received the B.S. degree in computer science from Dongguk University, Korea, in 1995, the M.S. degree in computer science from Pohang Institute of Science and Technology (POSTECH), Korea, in 1997, and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2009. He is currently a Professor at Department of Medical Information, Kongju National University, Korea. His research interests include information security and identity management.