Classification of Speaking and Singing Voices Using Bioimpedance Measurements and Deep Learning

# Classification of Speaking and Singing Voices Using Bioimpedance Measurements and Deep Learning

*,#Eugenio Donati, †Christos Chousidis, *Henrique De Melo Ribeiro, and *Nicola Russo, *London, and †Guildford, UK

**Summary:** The acts of speaking and singing are different phenomena displaying distinct characteristics. The classification and distinction of these voice acts is vastly approached utilizing voice audio recordings and microphones. The use of audio recordings, however, can become challenging and computationally expensive due to the complexity of the voice signal. The research presented in this paper seeks to address this issue by implementing a deep learning classifier of speaking and singing voices based on bioimpedance measurement in replacement of audio recordings. In addition, the proposed research aims to develop a real-time voice act classification for the integration with voice-to-MIDI conversion. For such purposes, a system was designed, implemented, and tested using electroglottographic signals, Mel Frequency Cepstral Coefficients, and a deep neural network. The lack of datasets for the training of the model was tackled by creating a dedicated dataset 7200 bioimpedance measurement of both singing and speaking. The use of bioimpedance measurements allows to deliver high classification accuracy whilst keeping low computational needs for both preprocessing and classification. These characteristics, in turn, allows a fast deployment of the system for near-real-time applications. After the training, the system was broadly tested achieving a testing accuracy of 92% to 94%.

**Key Words:** Speech classification−Singing detection−Bioimpedance measurements−Electroglottography −EGG-to-MIDI−Voice-to-MIDI−Voice information retrieval−Real-time voice classification.

## INTRODUCTION

Amongst the different means of human interaction, voice represents the primary form of communication. Along with speech, voice is also used in singing, which is employed as a form of communication and as an instrument in music. However, despite being the most basic musical instrument, voice lacks implementation in modern music technology compared to most other instruments. Many applications have been developed throughout the years to emulate musical instruments' physical and mechanical characteristics and deliver digital control signals, mainly in the form of musical instruments digital interface (MIDI). Due to its nature, however, voice poses a challenge for its conversion into MIDI, especially in a real-time environment. This is due to the processing of voice being bound to the use of microphones and sound recordings, which complicates voice information retrieval. A sound recording presents itself as a complex signal, the processing of which requires lengthy and computationally expensive procedures. Moreover, using microphones can add environmental noises and interferences that, in turn, make singing voice information retrieval even more complex.

Our previous research proposed in[1,2] and[3] tackles the challenges of singing voice information retrieval by replacing sound recordings with bioimpedance measurements of the vocal folds. Based on this concept, we proposed in[4] and[5] a method for efficient, real-time extraction of singing voice information based on bioimpedance measurements. The technology employed to evaluate the bioimpedance variations of phonation is Electroglottography (EGG) which will be discussed in section three. Using bioimpedance measurements instead of recorded sound delivers a much simpler signal. This simpler signal consequently allows a fast and resource-efficient extraction of voice information. In addition, as bioimpedance is measured directly from the larynx, the system is not affected by external noise and environmental sounds. Any move of the vocal folds, however, causes a change in bioimpedance and generates a signal. This, in turn, will produce a MIDI conversion even for nonphonatory instances. Implementing a classifier capable of distinguishing singing voice from other phonation acts would allow the system to discard unwanted signals and perform the MIDI conversion only for singing acts.

This project proposes using the information obtained from bioimpedance measurements to train a neural network for the real-time classification between speaking voice and singing voice. This approach offers an advantage in voice act classification due to the simplicity of the EGG signal. Voice depends on the oscillation of vocal folds, and the tension of the folds is employed in singing to control pitch and duration.[6,7] Because such tension is higher in singing, the tone differs between the speaking and singing voice. Consequently, the fundamental frequency of voice tends to be more stable in singing.[8,9] As a bioimpedance evaluation solely reflects the vocal folds' behavior, such measurement simplifies the estimation of the voice's fundamental frequency which represents a significant advantage over audio recordings. Alongside the stability of the fundamental

Accepted for publication March 29, 2023.

From the *School of Computing and Engineering, University of West London, London, UK; and the †Department of Music and Media, Institute of Sound Recording, University of Surrey, Guildford, UK.

#The author conducted the research as part of a PhD at the University of West London under the Vice-Chancellor Scholarship Scheme.

Address correspondence and reprint requests to Eugenio Donati, School of Computing and Engineering, University of West London, St Mary's Road Ealing, London W55RF, UK. E-mail: eugenio.donati@uwl.ac.uk

Journal of Voice, Vol. ■■, No. ■■, pp. ■■−■■
0892-1997



https://doi.org/10.1016/j.jvoice.2023.03.018

frequency, vocal intensity and spectral content also contribute to the differentiation between speaking and singing. However, while it excels at evaluating the vocal folds' frequency, EGG is not reliable in the evaluation of the voice amplitude intensity.[1,2] In addition, as the bioimpedance measurement is performed at the larynx, the spectral content added by the vocal tract if fully bypassed.[4,5] EGG, moreover, is not agnostic to gender or age as these can affect the size and elasticity of the vocal folds. As shown in[10] and,[11] however, gender and age tend to mainly affect the mean value of fundamental frequency in an individual vocal pitch range and not its stability. This project therefore bases the distinction of singing and speaking voices on the overtime stability of the fundamental frequency.

Another element that can differ between singing and speech is the duration of the vocal folds' contact time, with singing often requiring lengthier times to sustain longer notes or phrases. The alteration of contact time, however, can vary with the style of singing or speaking as well as specific phrases and phonemes succession making. However, the evaluation of contact times in EGG results to be misleading and unreliable in the evaluation of each individual vibratory cycle.[12]

The successful implementation of this research, in combination with the work presented in[4] and,[5] could allow the development of a voice-to-MIDI converter capable of applying the conversion solely to singing voice acts avoiding errors or interferences caused by speech.

The rest of this paper will be organized as follows. Section two presents a state-of-the-art analysis considering the work being carried across the literature in singing voice information extraction and voice classification. Section three briefly overviews the theoretical concepts behind the human phonation system and EGG. Section four describes the implementation of the classifier from the data processing to the development and training of the neural network. Section five analyses the results of the testing and evaluates the system. Finally, in section six, the conclusions are presented, the overall development is analyzed, and possible further implementations are considered.

## STATE OF THE ART

In the field of singing voice classification, the recent literature greatly revolves around singing voice detection (SVD) within audio segments and music tracks.[7] Usually, such a task is approached by extracting one or more audio features from an audio segment which are then paired with a classifier for SVD. Therefore, given the powerful features of neural networks in classification problems, deep learning approaches are increasingly employed in SVD.[13]

Schlüter et al.[14] proposed a method for SVD based on Convolutional Neural Networks where mel-spectrograms are used to train and evaluate the network. Here the authors focus on data augmentation for the improvement of the model by applying pitch-shifting and time-stretching to the data samples. The system reaches a maximum accuracy of about 91%. However, using CNNs and data augmentation results lengthy and computationally expensive. Moreover, the system is trained and tested on sound recordings and was not tested on real-time inputs.

You et al., in,[15] conducted a comparative study of different techniques for training and testing a CNN. One approach employs Mel Frequency Cepstrum Coefficients (MFCCs), resulting in an accuracy of about 88%. The other employs short time fourier transform (STFT) spectrograms reaching an accuracy of about 92%. Despite the system's high effectiveness and accuracy, image recognition and CNN result are inefficient for real-time applications. Similar to what was presented in,[14] this system also employed sound recordings and was not tested on live inputs.

Huang et al, in[16] also proposed a CNN methodology for SVD. Here the authors trained and tested the network with three different features: MFCCs, discrete fourier transform (DFT) coefficients and raw audio samples. The research shows how DFT coefficients yielded the most performing network with an accuracy of 92%. Once again, however, the use of CNNs with image recognition results time consuming and requires significant data preprocessing.

The use of CNN and image recognition techniques on audio recordings, as presented in,[14-16] results too lengthy and computationally inefficient for a real-time application. When considering the implementation with voice-to-MIDI conversion, for the system to be true real-time, a latency below 20 ms is needed.[4,17] In addition, the mentioned methods are designed for the detection of singing voice within music recordings or lengthy audio segments and not for the discrimination between speaking voice and singing voice.

An example of speech-singing classification is proposed in.[18] The authors present a method to discriminate between a singing voice and a speaking voice by applying maximum likelihood principles to both MFCCs and voice fundamental frequency information. The models reached an accuracy of 65% for MFCCs and 80% on fundamental frequency measurements. Nevertheless, such accuracy was obtained using as input 300 ms audio samples for MFCCs and 2 second recordings for fundamental frequency, which makes the method inefficient for true real-time.

Another approach to speech-singing classification is proposed in.[9] This study focuses on the difference in fundamental frequency between singing and speaking. In this case, a deep neural network (DNN) was trained using the fundamental frequency values in speech and singing recordings extracted through statistical analysis. The statistical analysis is carried out by applying a log-linear regression to the fast fourier transform (FFT) coefficients obtained from an audio sample. Using DNN in this application, as opposed to CNN, allows faster training and reduces the time needed for signal preprocessing. As this system is based on fundamental frequency readings, the use of recorded sound and microphones could pose a limitation. The system accuracy would rely significantly on the fundamental frequency readings, which could be compromised by microphone limitations,

such as sensitivity and bandwidth, and surrounding sound sources.

Therefore, a common point across the literature for both voice acts classification and SVD is using sound recordings and microphone inputs. The use of sound data for the evaluation of the fundamental frequency can become highly inefficient given degraded acoustic conditions, external noises, or surrounding sound sources. The susceptibility of microphones and their usage in acoustically noncontrolled environments generate a series of artefacts in the recording that could cause errors in evaluating the fundamental frequency. As the fundamental frequency is considered the primary element of differentiation between speech and singing, an erroneous evaluation would cause a significant drop in accuracy and efficiency.

This project tries to tackle the differentiation between a singing voice and a speaking voice using bioimpedance measurements. The proposed approach employs EGG to generate a signal mirroring the behavior of vocal folds, representing the fundamental frequency of voice. The system employs a DNN using MFCCs as input features to create a training dataset and perform real-time prediction on a live EGG input.

## OVERVIEW OF PHONATION AND EGG

This paper discusses voice information extraction through bioimpedance measurements. This is achieved using EGG which analyses vocal folds' behavior during a phonation act. Such a characteristic allows an EGG to produce a simpler signal than a microphone audio recording. The core of such difference is linked to the physiology of voice production.

### Phonation

In human phonation, a steady airflow is generated from the lungs and pushed through the trachea until it reaches the vocal folds within the larynx. Stimulated by the airflow, the folds move repetitively from a contact to a noncontact position, converting the kinetic energy of the airflow into acoustic energy.[6] The folds' vibration is periodic by nature; thus, the number of cycles per second is the frequency of oscillation of voice. This frequency represents the fundamental frequency of the produced voice.[1] This acoustic signal then, reaches the vocal tract; the physiological elements of the vocal tract generate multiple resonances that in turn add harmonics to the original sound. This latest stage of phonation defines the characteristics of a distinguishable voice sound.

### EGG

EGG is a known medical technology that evaluates the behavior of vocal folds by applying an electrical current across the larynx. A pair of electrodes are placed across the larynx cartilage, and an alternating current with low voltage and high frequency is applied across the vocal folds. The cyclic change in position of the folds causes a change in the distance between them. This changes the bioelectrical conductance (and, in turn, impedance) across the larynx.[19] Such behavior performs an amplitude modulation (AM) on the signal applied initially. The demodulation of said AM signal determines the modulating frequency.[20,21] The result is a sinusoidal-like signal representing the vocal fold movement cycles and, thus, the voice oscillation frequency.[22] Figure 1 shows a generic EGG waveform.

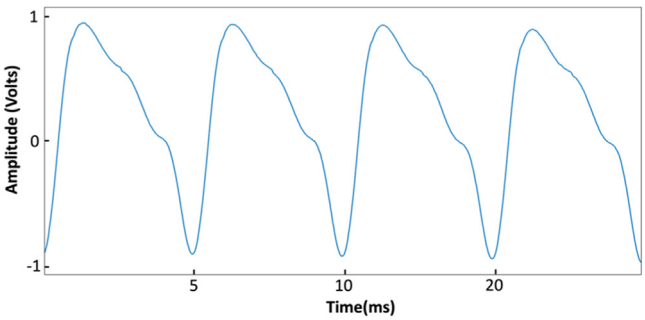



**FIGURE 1.** Generic EGG waveform.

Because of its characteristics, EGG can be particularly efficient in evaluating the fundamental frequency of voice. By acting directly at the vocal folds' level, it performs its reading before any resonance occurs across the vocal tract and, therefore, before any harmonic is added to the voice by the vocal tract.[23] The bypassing of the vocal tract, thus, allows to analyze a much simpler signal than an audio recording which in turn allows more efficient feature extraction and faster computation. As the amplitude of the voice is based on the pressure level of the airflow, the EGG is not as effective in measuring the amplitude of the voice sound as it is not dependent on the vocal folds' vibration. Figure 2 shows how the resultant signal is much simpler in comparison to that of an audio recording. On the left-hand side, the time-domain characteristics of the two signals are shown, while the right-hand side shows the spectral content of each signal. Both were recorded during the same phonation.

## METHODOLOGY

The proposed project employs a DNN to classify the human voice between singing and speaking. The technique chosen for this implementation is based on a fully connected DNN fed with numerical data obtained by processing the MFCCs of the EGG signals. The existing literature shows how DNNs are found to outperform traditional machine learning techniques in the processing of MFCC voice features due to the nonlinear characteristics of the input data.[24,25] The training dataset was constructed using 2400 samples of EGG with a 50% ratio of singing and speaking. The samples were recorded from 12 participants, each performing 100 sung notes at different frequencies and 100 spoken words with different intonations. For the singing, to minimize the effect of styles at this initial stage, the participants were asked to perform a sustained sung vowel at several pitches that would comfortably suit their vocal range. Similarly, for the spoken words, the participants were required to pronounce 100 separate words with their natural speech

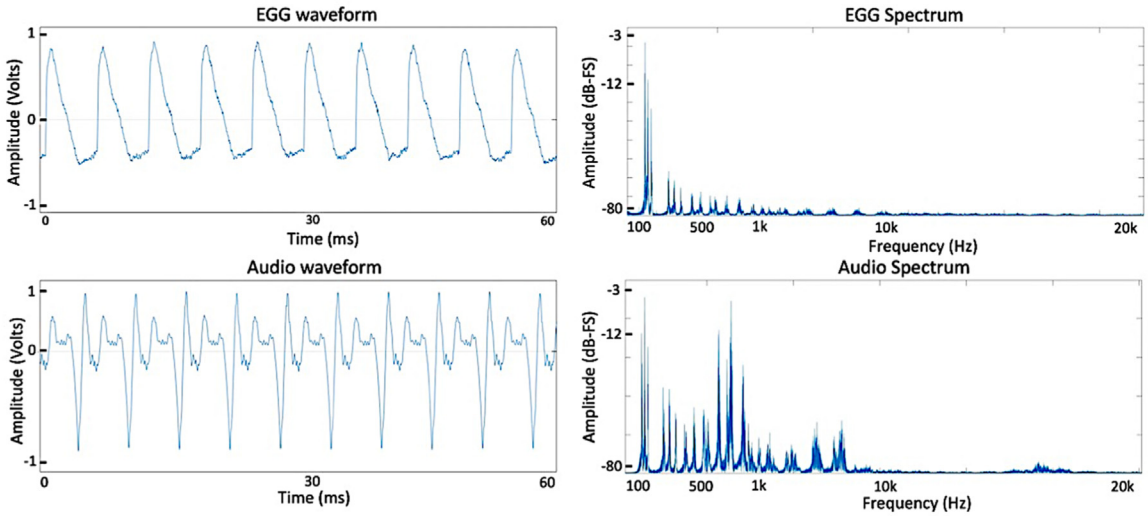



**FIGURE 2.** Comparison of EGG and audio for the same phonation in time and frequency domain.

pitches. The selected words were chosen to include sustained vocalic content so to perform a comparison with the sustained vowels of the singing samples.

Finally, to increase nonlinearity and the size of the dataset, the samples were subjected to a data augmentation process based on pitch-shifting. Each sample was pitch-shifted up and down by three semitones generating three versions of each. Data augmentation, thus, brought the dataset to a total of 7200 samples. Pitch-shifting is proven to be highly effective to both increase the data and obtain more generalized models.[14]

## Data preprocessing

In recent years, MFCCs have become widely used for voice information retrieval due to their ability to accurately represent spectral information at lower frequencies[26,27,28] and are considered to be the most appropriate for voice classification tasks.[29] MFCCs are the cepstrum coefficients of a signal with the frequency zones mapped to the mel scale by applying a series of triangular windows. As such a scale is intended to mimic the logarithmic perception of human hearing, MFCCs offer higher resolution than conventional FFT regarding the human voice spectrum.[30] These characteristics allow MFCCs to capture more detailed spectral and temporal characteristics than other methods commonly employed in speech processing such as entropy-based pitch estimation.[31] As the distinction between singing and speaking is here based mostly on the variability of the fundamental frequency, moreover, MFCCs result more suitable for the proposed application when compared to signal energy and amplitude-based methods.[32,33]

MFCCs are derived as follows:

A DFT is performed on a given signal.

1. The powers of the resulting spectrum are mapped to the mel scale. This is achieved by applying a series of overlapping triangular windows, typically 20 or 40. The number of windows per frequency band decreases with an increase in frequency resulting in higher precision in the lower end of the spectrum. Figure 3 shows a typical mel frequency window bank featuring 20 windows.
2. At each of the mel frequencies, the logarithms of the powers are calculated and then processed through a discrete cosine transform (DCT).
3. The MFCCs are the amplitudes of the DCT spectrum.

To train the network, the whole dataset was processed to extract the MFCCs which were then organized into a set of characterizing signal features referred to as a feature matrix.

For each sample, the coefficients were calculated using a DFT frame size of 512 samples and 20 triangular windows for the mel frequency mapping. The MFCCs processing delivers an array of coefficients the dimensions of which are dictated by the number of triangular windows and the length of the sample. The recorded samples in the dataset all feature different lengths depending both on the type of phonation and on the speaker, hence, the resulting arrays will feature variable lengths. For a duration of 500 ms, for example, a sampling frequency of

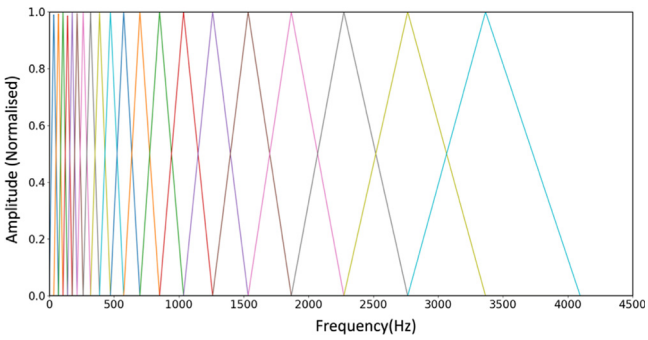



**FIGURE 3.** Mel frequency window bank.

| | Frame 1 | Frame 2 | Frame 3 | Frame 4 | Frame | Frame 39 | Frame 40 | Frame 41 | Frame 42 | Frame 43 |
|---|---|---|---|---|---|---|---|---|---|---|
| Window 1 | -367.88583 | -337.5296 | -336.75711 | -348.77524 | -340 | 318.01709 | -318.15436 | -315.11646 | -315.8645 | -285.83612 |
| Window 2 | 157.947083 | 182.359558 | 180.812668 | 164.632568 | 164 | 16.514053 | 186.280182 | 191.163071 | 189.476654 | 220.954834 |
| Window 3 | 94.161728 | 89.248337 | 85.644356 | 76.391388 | 70. | 68.379486 | 66.641785 | 72.366959 | 71.584946 | 80.715149 |
| Window 4 | 47.890198 | 34.516998 | 35.219231 | 37.708443 | 32. | 31.488604 | 26.919468 | 30.946009 | 32.26767 | 27.719481 |
| Window 5 | 14.177546 | 7.183388 | 13.671328 | 23.087357 | 20. | 25.582468 | 17.725622 | 19.225758 | 22.7164 | 17.511232 |
| Window 6 | -1.664939 | -3.726184 | 1.953016 | 9.930851 | 7. | 5.406825 | -4.499628 | -1.892562 | 4.148892 | 0.468826 |
| Window 7 | 1.175078 | -1.19842 | -2.060004 | 0.235697 | -1. | -1.843798 | -8.858826 | -5.698651 | -2.47386 | -5.917892 |
| Window 8 | -0.103343 | -4.385953 | -7.501308 | -7.358344 | -81. | 4.107086 | 0.259773 | 0.803679 | -1.184511 | -4.625482 |
| Window 9 | -7.124045 | -10.436687 | -8.236124 | -6.103273 | -61. | 2.830511 | -1.836334 | -1.180306 | -2.159634 | -7.177766 |
| Window 10 | -7.339389 | -7.803612 | -2.454901 | 2.426772 | 0. | 4.138357 | -8.467128 | -5.31683 | -2.777406 | -8.590819 |
| Window 11 | -4.78974 | -5.54912 | -3.947456 | 2.645816 | -1. | 5.354015 | -9.237675 | -5.298379 | -2.066489 | -8.191411 |
| Window 12 | -5.51976 | -7.003093 | -6.427798 | -2.574097 | -4. | 1.162338 | -4.201743 | -0.090454 | 1.73759 | -5.69737 |
| Window 13 | -5.983764 | -4.615072 | -0.940605 | 0.394096 | (29 | 4.161293 | -0.380226 | 3.011677 | 2.875058 | -2.691964 |
| Window 14 | -6.09188 | -3.854546 | -1.020143 | 1.357471 | -0. | -1.416429 | -5.269978 | -3.267189 | -4.303728 | -6.071244 |
| Window 15 | -8.619304 | -7.905156 | -6.279934 | -4.586061 | -83. | 5.657391 | -10.553171 | -7.967607 | -8.280605 | -8.935244 |
| Window 16 | -9.224706 | -8.642377 | -5.57129 | -3.271861 | -5. | 5.360586 | -9.077008 | -5.995215 | -5.442661 | -6.055198 |
| Window 17 | -8.159183 | -9.001717 | -5.495242 | -0.527855 | -0. | 7.647768 | -9.644127 | -7.811342 | -7.635093 | -8.817122 |
| Window 18 | -9.389755 | -11.133488 | -8.937087 | -6.58245 | -9. | 9.684253 | -11.914372 | -10.859359 | -10.734005 | -12.926013 |
| Window 19 | -7.864786 | -7.908289 | -7.92396 | -10.497719 | -11. | 4.591782 | -8.6637 | -8.71846 | -7.015992 | -7.499305 |
| Window 20 | -7.449003 | -7.049423 | -5.870617 | -8.459927 | -7. | 1.850064 | -7.154682 | -8.932299 | -5.663432 | -2.78934 |

**FIGURE 4.** MFCC data array for a sample with a 500 ms duration.

44.1 kHz, and 20 triangular windows, the resulting array will present a size of 20 × 43. This can be confirmed through equation 1.

For: sample rate = 44100, frame size = 512, samples duration 0.5seconds

$$duration\ in\ samples = 44100 \cdot 0.5 = 22050$$

$$number\ of\ frames = \frac{duration\ in\ samples}{frame\ size} = \frac{22050}{512} = 43 \quad (1)$$

The resulting coefficients are then organized into a CSV file. Figure 4 shows the output array for a single sample of 500 ms.

Next, to obtain a more suitable format for the DNN input layer, the mean value for each of the triangular windows is calculated yielding a 20 × 1 array which then transposed into 1 × 20. This process overcomes the different durations of the samples by removing the "time variable" and provides an equal size feature matrix for each sample.[31] Finally, feature scaling is applied, and all the values are standardized to zero mean and unit variance. Figure 5 shows the final 1 × 20 array for a single sample.

To train the network, the resulting arrays were organized in a single CSV file and labelled according to their typology as either "speech" or "sing." Figure 6 shows a section of the training dataset. The first column from the left shows the labelled assigned to each sample while the following columns present the MFCC computed per each triangular window.

## DNN architecture

The design of the DNN architecture was based on the underlying principle of implementing real-time voice act classification for the development of a real-time EGG-to-MIDI converter.[4,5] To achieve a true real-time processing for voice-to-MIDI conversion, the latency between a phonation act and the delivery of the resulting message cannot exceed 20 ms.[17] For this reason, the design of the DNN was approached so to achieve the simplest possible architecture. Based on the training data, the input layer of the DNN was constructed of 20 neurons to match the 1 × 20 array representing each sample and, in turn, the number of windows employed for the extraction of the MFCCs. Next, the network was implemented with a single hidden layer of 40 neurons featuring the *ReLU* activation function. The *ReLU* does not activate all neurons in a layer at the same time as only neurons receiving a positive input value are activated. This characteristic makes the *ReLU* computationally efficient as only some of the neurons are activated at once. Finally, the output layer was implemented with a single neuron. Given the binary nature of the classification, the output layer features a *Sigmoid* activation function.

The described architecture was selected through an experimental manipulation and testing of the network configuration by varying the number of neurons in the hidden layer. Whilst maintaining the dimensions of both the input and the output layers as 20 and one respectively, the network was thus tested and evaluated for several hidden layer configurations. This process will be analyzed in detail in the results section. Figure 7 shows the final architecture of the DNN.

| | Window 1 | Window 2 | Window 3 | Window | Window 18 | Window 19 | Window 20 |
|---|---|---|---|---|---|---|---|
| Scaled Mean | -4.158338 | 1.307359 | 0.662026 | 0.6032 | 0.056726 | 0.052628 | 0.059096 |

**FIGURE 5.** 1 × 20 scaled mean array for one recorded sample.

| LABEL | Win. 1 | Win. 2 | Win. 3 | Win. 4 | Win. 5 | Win. 6 | Win. 7 | Win. 8 | Win. 9 | Win. 10 | Win. 11 | Win. 12 | Win. 13 | Win. 14 | Win. 15 | Win. 16 | Win. 17 | Win. 18 | Win. 19 | Win. 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sing | 0.902 | 0.5931 | 0.9655 | -0.2559 | -1.0749 | -0.0338 | 1.4642 | -0.2449 | -0.2604 | 0.1409 | 0.7766 | -0.295 | 0.7683 | -0.0542 | 0.6214 | -0.2704 | 0.1672 | -0.168 | 0.256 | 0.0625 |
| speech | -1.7943 | 1.0106 | 1.0583 | -0.2852 | 0.5045 | -0.0424 | 0.7571 | 0.3254 | 1.3802 | 0.5691 | 1.5586 | 0.2965 | 1.136 | 0.3648 | 1.4408 | 0.8921 | 1.7792 | 0.6493 | 0.6887 | 0.1368 |
| speech | -1.6835 | -0.9455 | 1.3014 | 0.3863 | 0.1875 | -0.2307 | 0.3076 | 0.6897 | 1.4262 | 0.9286 | 1.6802 | 0.9265 | 1.902 | 1.4163 | 2.1236 | 1.2683 | 1.4877 | 0.8053 | 0.4978 | 0.4572 |
| speech | -0.2125 | 0.1294 | 0.1245 | 0.0476 | 0.3095 | 0.0078 | 0.4841 | 0.6345 | 0.7084 | 0.5871 | 0.8938 | 0.2221 | 1.1666 | 0.1776 | 1.2529 | 0.3368 | 0.9629 | 0.4296 | 0.7044 | 0.3506 |
| sing | -1.1402 | 0.743 | 0.0005 | 1.2385 | 0.9392 | 1.0954 | 0.5206 | 1.3531 | 0.455 | 1.6081 | 0.038 | 1.226 | -0.0642 | 0.5491 | -0.6482 | -0.8571 | -0.6044 | -1.0467 | -0.2047 | -1.0314 |
| speech | -1.0078 | -0.1747 | -0.6167 | 0.7164 | -0.7423 | 0.8234 | -0.6475 | 0.7567 | -0.2825 | 0.5401 | 0.2396 | 0.3122 | 0.6663 | 0.4161 | 0.8082 | 0.8521 | 1.1954 | 0.7194 | 0.2122 | 0.05 |
| sing | -0.1767 | 1.7797 | -1.8689 | -0.3387 | -1.9749 | 1.1197 | -1.3325 | 0.6131 | -1.5842 | -0.1552 | -1.6996 | -0.3514 | -0.9446 | -0.0044 | -1.4844 | -0.6635 | -0.8767 | -0.2272 | -0.3149 | -0.7351 |
| sing | -0.1999 | -0.4403 | -0.5556 | -0.3797 | 0.7738 | -0.5802 | 0.9893 | -0.5723 | 0.9818 | -0.2077 | 0.7575 | -0.205 | 0.1914 | -0.3828 | -0.3251 | 0.0979 | -0.238 | 0.3342 | -0.3054 | 0.115 |
| sing | -1.265 | 0.646 | -0.4902 | 1.1906 | 0.2517 | 1.0395 | 0.2679 | 1.234 | -0.2852 | 1.2011 | -0.4066 | 1.0782 | -0.5601 | 0.4668 | -0.6896 | -0.2166 | -0.1998 | -0.3953 | 0.3934 | -0.3014 |
| sing | -0.2244 | -0.2758 | -0.5918 | 0.0008 | 0.5529 | -0.3685 | 0.8879 | -0.3674 | 0.9064 | -0.1603 | 1.1186 | -0.0312 | 0.9503 | -0.1956 | 0.2461 | -0.4442 | -0.1298 | -0.1716 | -0.3335 | -0.1156 |
| speech | -0.8577 | 0.7003 | 0.8601 | -0.1418 | 0.4198 | -0.7614 | -0.1294 | 0.0725 | 1.0687 | 0.0426 | 1.2709 | 0.0998 | 1.0387 | 0.3108 | 1.3785 | 0.7156 | 1.3047 | 0.3371 | 0.3077 | -0.4658 |
| speech | -1.2032 | -0.4496 | 0.7279 | 0.2586 | 0.1674 | -0.2535 | 0.3307 | 0.5773 | 0.8422 | 0.8432 | 1.0132 | 0.7118 | 0.9977 | 0.8593 | 1.3346 | 0.7053 | 0.9494 | 0.8593 | 0.9558 | -0.0061 |
| speech | -0.2352 | 0.1949 | 0.7021 | 0.3538 | 0.0425 | -0.0571 | 0.2251 | 0.5382 | 1.0128 | 0.5419 | 1.3006 | 0.5023 | 1.4732 | 0.4797 | 1.6777 | 0.5304 | 1.4227 | 0.3221 | 0.5416 | 0.2596 |
| sing | -0.0428 | 1.5504 | -2.3055 | -0.7297 | -1.9072 | 0.2308 | -2.4135 | 0.1281 | -1.8987 | -1.0021 | -2.7002 | -1.1263 | -1.1252 | -0.3768 | -2.1611 | -2.1376 | -1.6205 | -0.2713 | -0.1027 | -1.155 |
| speech | -0.8399 | 0.5966 | -0.8079 | 0.5434 | -1.0137 | 1.7442 | -1.3446 | 1.1387 | -0.8172 | 1.0003 | -0.3521 | 0.5334 | -0.3162 | 0.3507 | 0.0085 | 0.2579 | -0.1803 | 0.107 | 0.1311 | 0.3463 |
| sing | 1.3304 | 0.6094 | -0.3628 | -0.5647 | -0.0043 | 0.4421 | 0.4545 | -0.0938 | 0.5283 | -0.0379 | 0.5861 | 0.0645 | 0.9466 | 0.0909 | 0.6768 | -0.0389 | 0.5038 | -0.0014 | 0.3655 | -0.0201 |

**FIGURE 6.** Labelled dataset for DNN training.

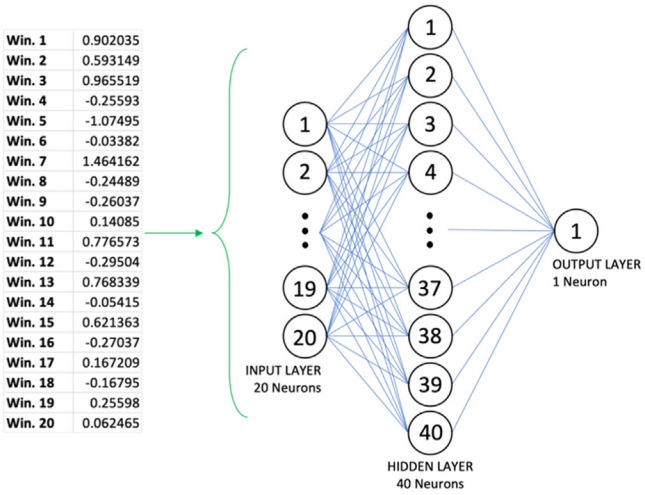



**FIGURE 7.** DNN architecture with relative input data.

## RESULTS

The performance and behavior of the classification system were tested using the created dataset, evaluating its accuracy in training and with "unseen" samples. Finally, the system was tested for real-time performances using a live input stream from the EGG.

### DNN training and evaluation

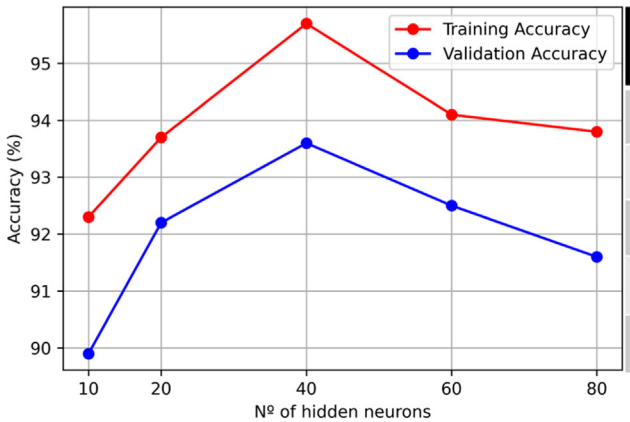
The first element for the analysis of the DNN was the evaluation of its accuracy. For the training of the NN, the dataset was randomly split into training data and validation data with a percentage of 70% and 30%, respectively. The training was performed on several configurations to evaluate the most suitable architecture, each featuring a different number of neurons in the hidden layer. In all tested cases, the training took between 20 and 25 seconds and delivered an accuracy above 90%. The training was performed over 50 epochs. Figure 8 shows the accuracy score for the various tested configurations.

Out of the tested architectures, configuring the neural network with 40 hidden neurons delivered the best performance, with a training accuracy of almost 96% and a validation accuracy of about 94%. The architecture was also tested with extra added hidden layers; this showed a negligible change in accuracy whilst increasing the computational needs of the network and it was therefore chosen to maintain the three-layer configuration. Figure 9 shows the accuracy through the training epochs for the final 20-40-1 configuration.

Once the model was trained, a performance test was conducted using 250 "unseen" samples recorded from the same participant as the training dataset. The DNN showed an accuracy of 92% in classifying "unseen" EGG samples. Figure 10 shows the confusion matrix for the testing process.

### Real-time performance testing

Finally, the network was tested for real-time performance. The EGG signal was streamed through an audio interface and fed to the trained Neural Network input. This was




| Nº of hidden neurons | Training Accuracy | Validation Accuracy |
|---|---|---|
| 10 | 92.3% | 89.8% |
| 20 | 93.7% | 92.2% |
| **40** | **95.7%** | **93.6%** |
| 60 | 94.1% | 92.5% |
| 80 | 93.8% | 91.8% |

**FIGURE 8.** Training and validation accuracy per number of hidden neurons.

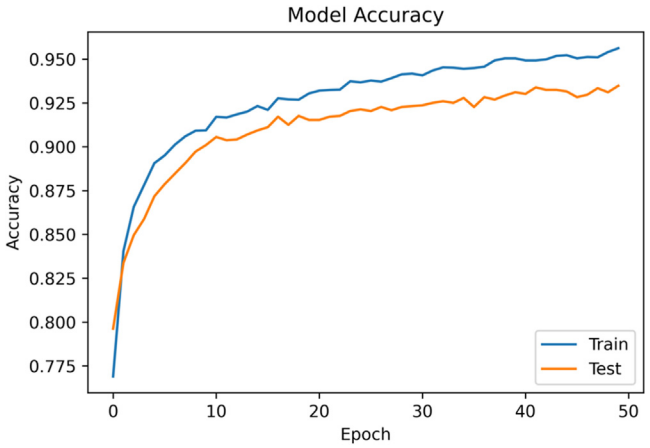



**FIGURE 9.** Model accuracy over 50 epochs for 20-40-1 configuration.

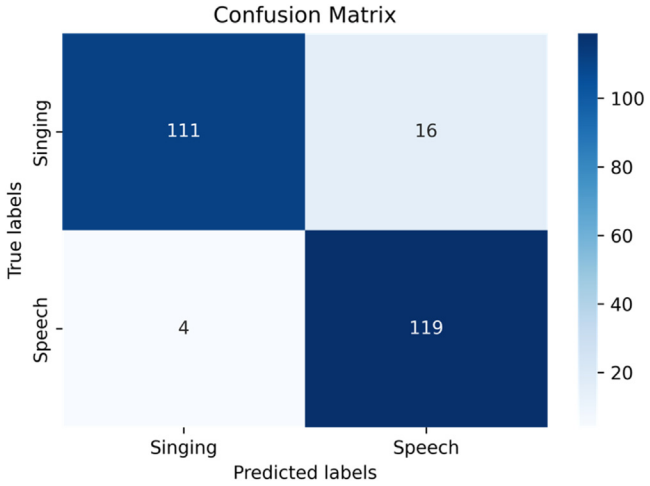



**FIGURE 10.** Confusion matrix for DNN performance with "unseen" samples.

implemented within a dedicated Python script. During each sampling frame, the processing of the EGG signal and the consequent DNN prediction were executed as follows:

The EGG input stream is sampled at 44.1 kHz, stored in a buffer with size 8820 samples, and the MFCCs are calculated.
The MFCCs' mean values are calculated and scaled to zero mean and unit variance.
The resulting array is reshaped into 1 × 20 and fed to the DNN input.
The DNN performs a prediction and classifies the signal between *Singing* and *Speech*.

For the analysis of the real-time performances of the DNN, a MIDI note number is outputted based on the prediction label. The MIDI note is then recorded together with the EGG input signal within a Digital Audio Workstation (DAW). This setup allows recording simultaneously both the EGG and the prediction labels hence comparing the

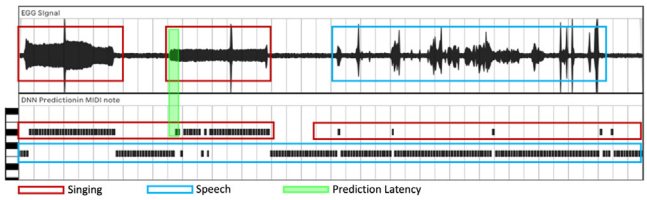



**FIGURE 11.** Real-time evaluation through audio and MIDI recordings.

effective phonation with the DNN classification. The outcome of the test showed an efficient behavior of the DNN with an overall prediction error of about 6% and a time delay between predictions of about 200 ms. Whether or not such latency can be considered real-time highly depends on the application. For voice-to-MIDI conversion, such latency is above the acceptable threshold. Figure 11 shows the results recorded in the DAW.

The time delay in between predictions is greatly dependent on the size input buffer. A buffer size of 8820 samples, in fact, is equivalent to 200 ms when using a sampling rate of 44.1 kHz. Despite such latency is not acceptable for voice conversion, it was observed through further testing that 200 ms of signal resulted as the minimum to achieve low error. This is mainly due to the distinction between speaking voice and singing voice. Such difference mainly relies on the steadiness of the fundamental frequency[8,9]; thus, the DNN requires sufficient signal periods to detect potential fluctuations.

## CONCLUSIONS

The presented research sought to the classification between speaking voice and singing voice using, as a signal source, electroglottography (EGG) instead of audio. Following our previous work in[4] and,[5] the classification system was designed to integrate with a real-time voice-to-MIDI conversion system. Such a system revolves around using EGG signal for its ease of processing and computational efficiency. 2400 EGG recordings were collected from 12 participants with a ratio of 50/50 between speaking and singing. In order to increase the available data and break linearity, the dataset was augmented through pitch shifting techniques, and a total of 7200 samples were obtained. The entire dataset was then processed to extract each entry's Mel Frequency Cepstral Coefficients (MFCCs). The resulting coefficients for each individual sample were then rearranged into a 1 × 20 array through mean and standard normalisation. The dataset obtained by this process was then used to train a fully connected deep neural network (DNN) featuring a single hidden layer and 20-40-1 architecture. The network delivered a validation accuracy of approximately 94%, with a training time of around 25s. The DNN was tested on 250 "unseen" samples and performed with a prediction accuracy of nearly 92%. Finally, the system was tested for real-time implementation, showing a total error of around 6%. The evaluation of the real-time performances

showed a time between predictions of about 200 ms. Such delay is primarily due to the use of 8820 samples sized input buffer. Whether such time between predictions can be considered real-time depends entirely on the application; for voice conversion, however, a latency not greater than 15 ms is required.[8,9] The need to employ such a big buffer is dictated by the distinction between speaking and singing, being mainly dependent on the fluctuations of the fundamental frequency. In order to allow the neural network to perform predictions properly, the input would require enough signal periods to detect changes in the fundamental frequency.

The experiments showed how EGG could effectively classify singing and speaking voices given its ability to represent the fundamental frequency of voice. Using EGG in combination with MFCCs allows the employment of a light deep neural network (DNN) architecture to achieve high accuracy and fast predictions. Using a voice classifier paired with an EGG-to-MIDI converter could allow the development of a standalone device capable of converting singing voice into MIDI whilst automatically discarding unwanted signals, such as speaking voice. The current prediction frequency, however, suggests that this would be possible in a parallel manner by implementing the classifier and the EGG-to-MIDI converter on two separate threads.

## REFERENCES


1. Chousidis, C. and Lipan, L. (2016). The application of a novel voice-driven MIDI controller in music education and training ICICTE 2016 Proceedings.
2. Kehrakos K, Chousidis C, Kouzoupis S. A reliable singing voice-driven MIDI controller using electroglottographic signal. *Audio Engineering Society Convention 140*. New York, NY: Audio Engineering Society; 2016.
3. Kehrakos, K., Kouzoupis, S. and Chousidis, C. (2016). An efficient method of extracting singing voice information using electroglottographic signal. 23rd International congress on Sound and Vibration.
4. Donati E, Chousidis C. Electroglottography based real-time voice-to-MIDI controller. *Neuroscience Informatics*. 2022;2:100041.
5. Donati E, Chousidis C. Electroglottography based voice-to-MIDI real-time converter with AI voice act classification. *17th IEEE Medical Measurement & Application*. 2022.
6. Garcia M. Observations on the human voice. *Proc R Soc Lond*. 1854;7:399–410.
7. Monir R, Kostrzewa D, Mrozek D. Singing voice detection: a survey. *Entropy*. 2022;24:114.
8. Vijayan K, Li H, Toda T. Speech-to-singing voice conversion: the challenges and strategies for improving vocal conversion processes. *IEEE Signal Process Mag*. 2018;36:95–102.
9. de Medeiros BR, Cabral JP, Meireles AR, et al. A comparative study of fundamental frequency stability between speech and singing. *Speech Commun*. 2021;128. 15-2.
10. Herbst CT. Electroglottography–an update. *J Voice*. 2020;34:503–526 .
11. Ma EPM, Love AL. Electroglottographic evaluation of age and gender effects during sustained phonation and connected speech. *J Voice*. 2010;24:146–152.
12. Herbst C, Ternström S. A comparison of different methods to measure the EGG contact quotient. *Logoped Phoniatr Vocol*. 2006;31:126–138.
13. Zhang X, Yu Y, Gao Y, et al. Research on singing voice detection based on a long-term recurrent convolutional network with vocal separation and temporal smoothing. *Electronics*. 2020;9:1458.
14. Schlüter J, Grill T. *Exploring data augmentation for improved singing voice detection with neural networks*. Canada: The International Society for Music Information Retrieval (ISMIR; 2015:121–126.
15. You SD, Liu CH, Chen WK. Comparative study of singing voice detection based on deep neural networks and ensemble learning. *Hum.-Centric Comput. Inf. Sci.*. 2018;8:34.
16. Huang HM, Chen WK, Liu CH, et al. Singing voice detection based on convolutional neural networks. In: *Proceedings of the 2018 7th International Symposium on Next Generation Electronics (ISNE)*. 2018.
17. Stowell D, Plumbley MD. Delayed decision-making in real-time beatbox percussion classification. *J N Music Res*. 2010;39:203–213.
18. Ohishi Y, Goto M, Itou K, et al. Discrimination between singing and speaking voices. *Ninth European Conference on Speech Communication and Technology*. 2005.
19. Fabre P. La glottographie electrique en haute frequence, particularites de lappareillage. *C R Seances Soc Biol Fil*. 1959;153:1361–1364. [Publication in French].
20. Fourcin AJ. Laryngographic examination of vocal fold vibration. *Ventilatory and Phonatory Control Systems: An International Symposium*. Oxford University Press; 1974:315–333.
21. Titze I. Interpretation of the electroglottographic signal. *J Voice*. 1990;4:1–9.
22. Drugman T, Alku P, Alwan A, et al. Glottal source processing: from analysis to applications. *Comput Speech Lang*. 2014;28:1117–1138.
23. Drugman T, Bozkurt B, Dutoit T. A comparative study of glottal source estimation techniques. *Comput Speech Lang*. 2012;26:20.
24. Abiodun OI, Jantan A, Omolara AE, et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon*. 2018;4:e00938.
25. Liu W, Wang Z, Liu X, et al. A survey of deep neural network architectures and their applications. *Neurocomputing*. 2017;234:11–26.
26. Pishgar M, et al. *Pathological Classification Using Mel-Cepstrum Vectors and Support Vector Machine*. IEEE; 2018:5267.
27. Bae Hyan-Soo, Lee Ho-Jin, Lee Suk-Gyu Voice recognition based on adaptive MFCC and deep learning. IEEE. 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, China, 2016;1542.
28. Boles A, Rad P. *Voice biometrics: Deep learning-based voiceprint authentication system*. IEEE; 2017:1.
29. Rocamora, M. and Herrera, P. (2007) Comparing audio descriptors for singing voice detection in music audio files. pp. 27.
30. Lee, K., Choi, K. and Nam, J., (2018). Revisiting singing voice detection: a quantitative review and the future outlook.
31. Zakariah M, et al. An Analytical Study of Speech Pathology Detection Based on MFCC and Deep Neural Networks. *Comput Math Methods Med*. 2022;2022.
32. Guido RC. A tutorial review on entropy-based handcrafted feature extraction for information fusion. *Info Fusion*. 2018;41:161–175.
33. Guido RC. A tutorial on signal energy and its applications. *Neurocomputing*. 2016;179:264–282.