A Deep Learning Method for Pathological Voice Detection using Convolutional Deep Belief Network

Huiyi Wu[1], *John Soraghan*[1], *Anja Lowit*[2], *Gaetano Di Caterina*[1]

[1]Centre for Signal and Image Processing, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, Scotland
[2]Speech and Language Therapy, School of Psychological Sciences and Health, University of Strathclyde, Glasgow, Scotland

huiyi.wu@strath.ac.uk, j.soraghan@strath.ac.uk, a.lowit@strath.ac.uk, gaetano.di-caterina@strath.ac.uk

Abstract

Automatically detecting pathological voice disorders such as vocal cord paralysis or Reinke's edema is a challenging and important medical classification problem. While deep learning techniques have achieved significant progress in the speech recognition field there has been less research work in the area of pathological voice disorders detection. A novel system for pathological voice detection using convolutional neural network (CNN) as the basic architecture is presented in this work. The novel system uses spectrograms of normal and pathological speech recordings as the input to the network. Initially Convolutional deep belief network (CDBN) are used to pre-train the weights of CNN system. This acts as a generative model to explore the structure of the input data using statistical methods. Then a CNN is trained using supervised back-propagation learning algorithm to fine tune the weights. It will be shown that a small amount of data can be used to achieve good results in classification with this deep learning approach. A performance analysis of the novel method is provided using real data from the Saarbrucken Voice database.

Index Terms: pathological voice detection, convolutional neural network (CNN), Convolutional deep belief network (CDBN), deep learning

1. Introduction

Voice pathologies affect the larynx and result in irregular vibrations of the vocal folds. This leads to psychological and physiological problems to individuals, and it has a significant impact on economy considering the costs of medical diagnosis and treatment[1].

Traditional diagnostic method of voice pathologies relies on clinician's experiences and on expensive devices such as laryngoscope, endoscope etc. However, computer-aided medical systems for diagnosis of voice pathologies have been popular due to major advance in signal processing techniques are introduced. These complementary tools are usually non-invasive and non-subjective, which generally are an advantage in medical field.

Many research works related to automatic detection of voice pathologies have been carried out in the past few decades. In this context, features are extracted from the speech recordings and they are then processed by classifiers to distinguish normal voice instances from pathological voice recordings. These features are mainly derived from two research fields. One is from speech recognition applications, with signal processing tools used to automatically detect features such as Mel-Frequency cepstral coefficients (MFCC), linear prediction cepstral coefficients (LPCC) and energy and entropy of discrete wavelet packets[2-4]. Other features come from voice quality measurement according to physiological and etiological research. While pitch, jitter and shimmer are used to detect the roughness of the speech, other characteristics such as harmonic-to-noise ratio (HNR), normalized noise energy (NNE), glottal-to-noise ratio (GNR) and cepstral peak prominence (CPP) represent the breathiness of the speech[5].

Most of the research works use the Massachusetts Eye and Ear Infirmary (MEEI) database. However, healthy voice recordings and pathological voice recordings in this database are recorded in two different environments[6], which make it hard to distinguish whether it is discriminating environments or voice features. Saarbruecken Voice Database is a downloadable database with all recordings sampled at 50 kHz and with 16-bit resolution. This database is relatively new so that little research has been done with this database. However, the recordings are recorded in the same environment so that it is an ideal database for this work.

It is shown that state-of-the-art signal processing techniques, which applied in speech recognition field before, also achieved significant progress in automatic pathological voice detection field. For example, Martínez et al. in [7] use Gaussian mixture model (GMM) on Saarbruecken Voice Database, and achieved 67% classification accuracy with neutral sustained vowel /a/. However, with enhanced computational abilities of hardware and improvement of machine learning algorithms, deep neural network (DNN)-hidden Markov model (HMM) is gradually replacing the traditional GMM-HMM[8] to become the popular method for speech recognition. To date deep learning methods have not commonly been used in pathological voice detection field mainly because of the lack of data available in this research field is limited, as a DNN requires large amount of data to be trained.

Hinton et al. in [9] proposed Restricted Boltzmann Machine (RBM) as an unsupervised method for pre-training DNN to achieve global minima precisely. As a generative model, it will improve the deep learning performance even on small dataset. Convolutional Deep Belief Networks (CDBN) were proposed by Lee et al.[10] as an advanced specific structure for pre-training CNN.

In this paper, we propose a novel deep learning method to automatically discriminate pathological voice and healthy

voice. A convolutional neural network (CNN) structure is utilized in this work to analyze spectrograms of speech recordings automatically. CDBN is used for pre-training the weights and avoid overfitting problems.

The rest of the paper is organized as follows. Section 2 describes the methodology in detail. In section 3, results are presented and further discussed. Finally, conclusions are drawn in section 4.

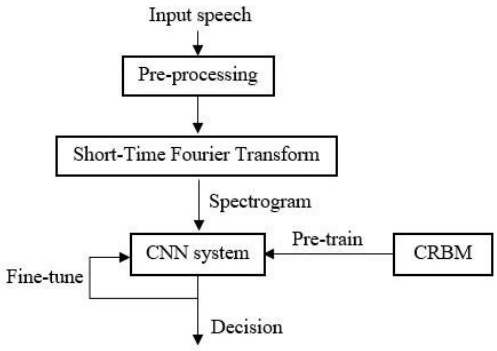

Figure 1: *Block diagram of proposed pathological voice detection system*

2. Methodology

Figure 1 shows the block diagram of proposed pathological voice detection system. First, pre-processing steps, such as resampling, reshaping techniques, are applied to the speech recordings. Short-time Fourier transform (STFT) technique is then applied to get the spectrograms of the speech recordings as the input to the CNN system. Weights in the CNN system are pre-trained using CDBN and fine-tuned with back-propagation method. The trained CNN system is capable of extracting features automatically and classifying audio samples.

2.1. Input Data to CNN system

CNN contains "feature extractors" which is commonly applied to feature maps. Therefore, speech recordings are transformed from one-dimensional signals to two-dimensional spectrograms.

2.1.1. Database

This work uses the Saarbruecken voice database which was recorded by Institute of Phonetics of Saarland University in Germany. This database contains 71 different pathologies with speech recordings from over 2000 individuals. Each participant file contains recordings of sustained vowels /a/, /i/ and /u/ in neutral, low, high and low-high-low intonations and a continuous speech sentence "Guten Morgen, wie geht as Ihnen?" ("Good morning, how are you?"). Sustained vowel is applied in this work because it is stationary throughout time and easier to see the changes.

6 pathologies are selected as the pathological group, including laryngitis, leukoplakia, Reinke's edema, recurrent laryngeal nerve paralysis, vocal fold carcinoma and vocal fold polyps. These pathologies are all organic dysphonia which are caused by structural changes in the vocal cord. We use sustained vowel /a/ at neutral pitch of each individual, of which 482 are healthy and 482 are diagnosed with pathologies (140 laryngitis, 41 leukoplakia, 68 Reinke's edema, 213 recurrent laryngeal nerve paralysis, 22 vocal fold carcinoma and 45 vocal fold polyps). The data is divided into training set and testing set, containing 75% and 25% samples respectively.

2.1.2. Pre-processing and organization of input data

First, the original speech is resampled at 25 kHz in pre-processing step. The aim of this step is to reduce the amount of data in feature map to boost the training process. Furthermore, STFT is applied to transform the time-domain signal into spectral-domain signal. In this step, each file is divided into 10 ms Hamming window segments, with 50% overlap between consecutive windows. Finally, the spectrogram is reshaped to same size of 60*155 points to get rid of the useless part which contains no information. In this case, useless noise is dismissed and essential features are presented. The comparison of input feature maps between normal voice and pathological voice is shown in figure 2.

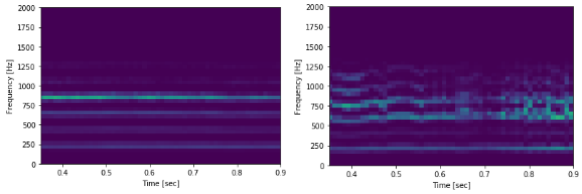

Figure 2: *Comparison of input feature maps (a).spectrogram of one normal voice; (b).spectrogram of one pathological voice*

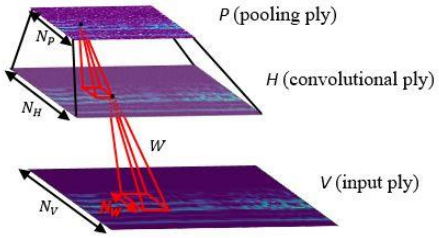

Figure 3: *CNN structure in one layer*

2.2. CNN architecture

CNN is built by an input layer and several hidden layers. Each individual layer consists of convolutional ply H and pooling ply V, which is shown in figure 3. For intonations, input feature map is set as $V_l(l = 1, \dots, L)$, and convolutional feature map is set as $H_k(k = 1, \dots, K)$. Weights (filters) are shared among all the units on convolutional ply, on which each unit is computed as,

$$h_m^k = \sigma\left(\sum_{l=1}^{I}\sum_{n=1}^{N_W} v_{l,n+m-1} w_{l,n}^k + w_0^k\right) \quad (1)$$

where $v_{l,m}$ means the m-th unit of the l-th input ply V, and $h_{k,m}$ means the m-th unit of the k-th convolutional ply H. N_w is set as the size of filters (weights), $w_{l,n}^k$ is the n-th unit of the weight. In this procedure, features are detected locally and automatically by shared-weights throughout the feature map.

In order to reduce the resolution in convolutional ply and reduce the computational complexity, pooling from convolutional map is essential. Maximization or averaging

function are commonly applied to build pooling ply. We set G as the size of pooling window, using maximization function, and unit on pooling ply is defined as,

$$p_m^k = max_{n=1}^{G}\, h_{l,(m-1)\times s+n} \quad (2)$$

where s is stride when the pooling windows shifting among the convolutional ply.

The whole CNN architecture is shown in figure 4. There are a total of 10 hidden layers. In the first hidden layer, the size of filters are 8*3 and the stride is 1. The size of pooling window is 4*4 and stride is 1. After the first hidden layer, each layer was convolved with 8 filters with the shape 8*3*8 and stride of 1. Max-pooling windows are 4*4 and activation function is *RELU* throughout the neural network.

Finally, the feature map is formed into a *Dense Layer* (fully-connected layer), to train the model for classification. *L2*-regularization is applied to avoid overfitting problems.

Parameters such as stride, the size of the filters in each layer, and number of layers were selected after hundreds of experiments. We use rectangular filter window due to the characteristics of spectrograms.

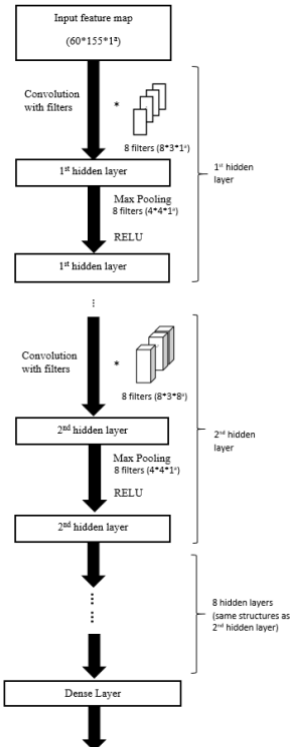

Figure 4: *CNN architecture*

2.3. Generative Pre-Training

Deep learning is a "black box" which requires a large amount of data and weight tuning processes. In contrast, Bayesian methods are robust and small-data interpretable which performs slightly poorer than deep learning techniques[11]. To combine the complimentary advantages of these two methods, generative models were developed to improve the deep learning performance on small data set and eliminate the over-fitting problems.

In deep learning structures, a region of weight-space is found by generative model and helps the network to converge to global minimum rapidly. Convolutional Restricted Boltzmann Machine (CRBM) is a typical generative model, and is an extension to the RBM with visible ply and hidden ply as images, which is suitable for CNN settings. The model is trained to reach *thermal equilibrium* state, which is the deepest energy minimum state. In this state, hidden ply is able to model the structure of the input data.

CRBM consists of two plies, the visible (input) ply V, and a hidden (convolutional) ply H. Similar to CNN setting, weights W^k between input ply and convolutional ply are shared among all locations in the hidden ply. Hidden units are binary-valued while visible units can be real-valued or binary-valued.

Assume the size of visible ply is N_V, and the size of hidden ply is N_H. There are *K* filters (weights) and each weight W^k is convolved with visible ply, and there are bias b_k for each weights and bias c for visible ply. The energy function with binary input is defined as,

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{k=1}^{K}\sum_{j=1}^{N_H}\sum_{r=1}^{N_W} h_j^k\, W_r^k v_{j+r-1} - \sum_{k=1}^{K} b_k \sum_{j=1}^{N_H} h_j^k - c\sum_{i=1}^{N_V} v_i \quad (3)$$

The energy function with real-value data input is defined as,

$$E(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{2}\sum_{i}^{N_V} v_i^2 - \sum_{k=1}^{K}\sum_{j=1}^{N_H}\sum_{r=1}^{N_W} h_j^k\, W_r^k v_{j+r-1} - \sum_{k=1}^{K} b_k \sum_{j=1}^{N_H} h_j^k - c\sum_{i=1}^{N_V} v_i \quad (4)$$

The joint distribution is defined as,

$$P(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\exp(-E(\boldsymbol{v},\boldsymbol{h})) \quad (5)$$

Similarly, CRBM is trained using block Gibbs Sampling[10] as an extension to Gibbs Sampling in RBM, to maximize the similarity of distribution between construction visible ply and input visible ply, in which case reach the equilibrium state.

Stacks of CRBM constitutes convolutional deep belief network (CDBN). After the first layer of CRBM is trained, the activations are sent to the second layer as input and the weights are "frozen", and the rest layers can be done in the same manner. Since visible ply in the first layer is clamped with real-valued data, Gaussian visible units are applied for the first CRBM layer.

After pre-training the weights in each layers, the well-known back-propagation are applied for fine-tuning the weights for better classification result.

2.4. Experimental Setup

The framework for the training process was developed in Python using Tensorflow[12]. Training data are divided as 256 samples in each mini-batch, and is trained with GPU NVidia GTX1070 for higher speed. In order to make the training process more robust, an Adam optimizer[13] was applied as an adaptive optimizer for better performance. Delta value of L2 regularization is set to 0.0001 and the maximum epochs of training is 100.

CDBN sparsity is set as 0.6 and weights pre-trained in the first two CRBM layers are set as initialization of CNN.

3. Results

Table 1 and Table 2 present the confusion matrix of validation dataset and testing dataset. In Table 3, classification results in different metrics are listed. Sensitivity (*SN*) and Specificity (*SF*) are calculated in (6). Sensitivity reveals the performance on detecting the pathological voice files, and Specificity reveals the proportion of correctly detected healthy voice files. Precision (*P*) and F1-score (*F1*) are presented in (7), where Precision reveals the proportion of relevant pathological voice files.

$$SN = \frac{TP}{TP + FN}, SF = \frac{TN}{FP + TN} \tag{7}$$

$$P = \frac{TP}{TP + FP}, F1 = 2\frac{P \cdot SN}{P + SN} \tag{8}$$

True Negative (TN) means that healthy voice recordings are correctly detected and Ture Positive (TP) means that pathological voice recordings are correctly detected; False Negative (FN) represents that pathological voice recordings are detected wrong and False Positive (FP) represents that healthy voice recordings are detected wrong.

It can be seen from Table 3 that validation set accuracy and testing set accuracy achieved 68% and 71% respectively. Compared to [7], it shows small progress when using small dataset. However, there is still large space for improvement if more experiments are conducted and several hyperparameters are tuned. In table 4, CNN system with and without CDBN pre-training reveals difference in classification result. It is shown that when using CDBN to initialize the weights, CNN tuning becomes more robust, with similar performance on validation dataset and testing dataset. In this case, it is proved that CDBN can avoid over-fitting problems to some extent. However, the testing set accuracy is less when using CDBN pre-trained weights, which reveals that accuracy might be affected when the system is more robust.

Table 1: *Confusion matrix of validation dataset*

	True: pathological	**True: healthy**
Prediction: pathological	53	30
Prediction: healthy	16	46

Table 2: *Confusion matrix of testing dataset*

	True: pathological	**True: healthy**
Prediction: pathological	55	23
Prediction: healthy	19	48

Table 3: *Metrics to evaluate classification result*

	SN	***SP***	***p***	***F1***	***ACC***
Validation dataset	0.77	0.60	0.64	0.70	0.68
Testing dataset	0.74	0.68	0.71	0.72	0.71

Table 4: *Classification accuracy with or without CDBN pre-training*

	CNN	**CNN + CDBN**
Validation dataset	0.66	0.68
Testing dataset	0.77	0.71

4. Conclusions

A novel algorithm for pathological voice detection is introduced in this work. Convolutional neural network is shown to effectively extract features from spectrograms of voice recordings and diagnose voice disorders. Convolutional deep belief network helps initialize the weights and make the system more robust. However, a tradeoff must be struck between robustness and accuracy. In the future work, more experiment will be conducted to balance them, and parameters will be tuned to achieve better performance.

5. Acknowledgements

The authors would like to acknowledge Capita plc and University of Strathclyde for their financial support with this study.

6. References

[1] K. Verdolini and L. O. Ramig, "Occupational risks for voice problems," *Logopedics Phoniatrics Vocology*, vol. 26, no. 1, pp. 37-46, 2001.

[2] A. A. Dibazar, S. Narayanan, and T. W. Berger, "Feature analysis for automatic detection of pathological speech," in *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] [Engineering in Medicine and Biology*, 2002, vol. 1, pp. 182-183 vol.1.

[3] M. K. Arjmandi and M. Pooyan, "An optimum algorithm in pathological voice quality assessment using wavelet-packet-based features, linear discriminant analysis and support vector machine," *Biomedical Signal Processing and Control*, vol. 7, no. 1, pp. 3-19, 2012/01/01/ 2012.

[4] M. Hariharan, K. Polat, and S. Yaacob, "A new feature constituting approach to detection of vocal fold pathology," *International Journal of Systems Science*, vol. 45, no. 8, pp. 1622-1634, 2014/08/03 2014.

[5] A. Al-nasheri *et al.*, "An Investigation of Multidimensional Voice Program Parameters in Three Different Databases for Voice Pathology Detection and Classification," *Journal of Voice*, vol. 31, no. 1, pp. 113.e9-113.e18.

[6] G. Muhammad *et al.*, "Voice pathology detection using interlaced derivative pattern on glottal source excitation," *Biomedical Signal Processing and Control*, vol. 31, pp. 156-164, 2017/01/01/ 2017.

[7] D. Martínez, E. Lleida, A. Ortega, A. Miguel, and J. Villalba, "Voice Pathology Detection on the Saarbrücken Voice Database with Calibration and Fusion of Scores Using MultiFocal Toolkit," in *Advances in Speech and Language Technologies for Iberian Languages: IberSPEECH 2012 Conference, Madrid, Spain, November 21-23, 2012. Proceedings*, D. Torre Toledano *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 99-109.

[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] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.

[10] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," presented at the Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada, 2009.

[11] J. Shi *et al.*, "ZhuSuan: A Library for Bayesian Deep Learning," *arXiv preprint arXiv:1709.05870*, 2017.
[12] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.