ROBUST PITCH TRACKING IN NOISY SPEECH USING SPEAKER-DEPENDENT DEEP NEURAL NETWORKS

Yuzhou Liu[1] *and DeLiang Wang*[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{liuyuz,dwang}@cse.ohio-state.edu

ABSTRACT

A reliable estimate of pitch in noisy speech is crucial for many speech applications. In this paper, we propose to use speaker-dependent (SD) deep neural networks (DNNs) to model the harmonic patterns of each speaker. Specifically, SD-DNNs take spectral features as input and estimate probabilistic pitch states at each time frame. We investigate two methods for SD-DNN training. The first one is direct training when speaker-dependent data is sufficient. The second one is speaker adaptation of a speaker-independent (SI) DNN with limited data. The Viterbi algorithm is then used to track pitch through time. Experiments show that both training methods of SD-DNNs outperform an SI-DNN based system as well as a state-of-the-art pitch tracking algorithm in all SNR conditions.

Index Terms— Pitch estimation, deep neural network, hidden Markov model, speaker-dependent modeling

1. INTRODUCTION

Pitch, or fundamental frequency (F0) of human speech can be used as an important cue for automatic speech recognition [3], speaker identification [23] and speech separation [20]. Many algorithms have been designed for pitch tracking [2] [4] [18], and they all achieved excellent performance on clean speech. However, in situations where speech is severely corrupted by noise, the performance of pitch trackers degrades drastically, which makes the estimated pitch uninformative for speech applications. Although a lot of recent studies tried to address the noise-robustness issue for pitch tracking, it is still challenging to extract pitch at negative signal to noise ratios (SNRs).

We can broadly group robust pitch tracking algorithms into three categories: spectral approach, temporal approach and spectrotemporal approach [20]. Spectral approaches analyze harmonic structure of speech in the spectral domain. For example, PEFAC [7] selected pitch candidates from an amplitude-compressed and comb-filtered spectrogram. Han and Wang [8] used the processed spectrogram in PEFAC [7] as the input feature, and tracked pitch using a deep neural network–hidden Markov model (DNN–HMM) based system. The second category, i. e., temporal approaches, examines the periodicity in the time domain by using autocorrelation functions (ACFs), e. g., RAPT [18] captured peaks in normalized ACFs and used dynamic programming for pitch selection. Lastly, spectrotemporal approaches decompose the signal using a bank of filters, and then apply time domain analysis on each subband signal. For instance, Jin and Wang [13] used periodicity information on reliable channels to model the pitch distribution and estimated continuous pitch tracks with an HMM. Lee and Ellis [14] applied principle component analysis on subband autocorrelation functions, and fed the derived features into a multilayer perceptron for pitch score estimation. Among these algorithms, Han and Wang [8] reported the best performance at negative SNRs.

On the other hand, a generic model for pitch estimation may be suboptimal for a given speaker, as a speaker's vocal tract and harmonic patterns can be unique. Speaker-dependent modeling is used recently in multipitch tracking tasks [16] [21] and has shown large improvement over generic models. However, none of the abovementioned single pitch tracking algorithms utilize speaker-dependent information for pitch estimation.

In this study, we extend Han and Wang's pitch tracking framework [8], and investigate the impact of individual speaker characteristics on robust pitch tracking. First, we use DNNs to model the posterior probability that a frequency bin (pitch state) is pitched given the frame-level observation. Instead of training a generic DNN speaker-independently, we train one DNN for each enrolled speaker using speaker-dependent training data. Specifically, two training techniques, i. e., direct training and speaker adaptation, are explored. We then investigate related issues affecting the training process, including input-layer/output-layer/all-layer adaptation and the training size. Lastly, an HMM with the Viterbi algorithm [5] is used to connect all frame-level probabilities and generate continuous pitch tracks.

The rest of the paper is organized as follows. The next section describes the details of the proposed pitch tracking

This research was supported in part by an AFOSR grant (FA9550-12-1-0130) and the Ohio Supercomputer Center.

system. Experimental results and comparisons are presented in Section 3. Section 4 concludes the paper.

2. SYSTEM DESCRIPTION

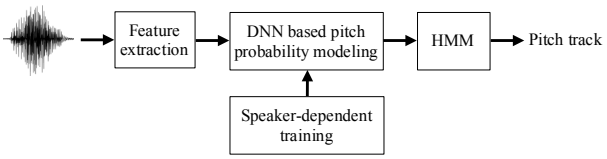

Fig. 1: Diagram of the proposed system.

A diagram of the proposed system is shown in Fig. 1. We first extract the frame-level feature vector $\mathbf{y}_m$ from a noisy utterance in the spectral domain, where m denotes the frame index. The frame length is 90 ms and the frame shift is 10 ms. In the next step, features are fed into DNNs to compute the posterior probability of pitch states at frame m, i. e., $p(x_m|\mathbf{y}_m)$, where x_m denotes the pitch state at frame m. x_m has 68 unique states $(s^1, s^2, s^3, \ldots, s^{68})$, where s^1 corresponds to an unvoiced or silent state, and s^2 to s^{68} refer to different frequency bins ranging from 60 to 404 Hz [8]. Specifically, the frequency range 60 to 404 Hz is divided into 67 bins using 24 bins per octave in a logarithmic scale. $p(x_m = s^i|\mathbf{y}_m)$ equals 1 if the groundtruth pitch falls in the frequency bin of s^i. To leverage speaker-dependent information, one DNN is trained for each speaker. More details of speaker-dependent DNN training can be found in Section 2.3 and Section 3. After the estimation of pitch probabilities, an HMM with the Viterbi algorithm is employed to track pitch through time.

2.1. Feature extraction

The feature used in study is introduced by Gonzalez and Brookes [7] and used by Han and Wang in [8].

To get the frame-level feature, a signal is first decomposed using short-time Fourier transform (STFT), where the power spectral density of STFT is denoted as $X_m(f)$. m is the frame index and f is the frequency bin index. We then interpolate $X_m(f)$ onto a log-spaced frequency resolution, and denote it as $X_m(q)$, where $q = \log(f)$.

The derived log-frequency power spectrogram is normalized through time to attenuate narrow-band noise: $X'_m(q) = X_m(q)\frac{L(q)}{\overline{X}_m(q)}$, where $L(q)$ is a long-term average speech spectrum, and $\overline{X}_m(q)$ is a smoothed spectrum using a moving average window.

Next, a comb-filter $h(q)$ is convolved with $X'_m(q)$ to enhance harmonic peaks: $\tilde{X}_m(q) = X'_m(q) * h(q)$, where:

$$h(q) = \begin{cases} \frac{1}{\gamma - cos(2\pi e^q)} - \beta, & \text{if } \log(0.5) < q < \log(10.5) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Here β is selected so that $\int h(q)dq = 0$. γ controls the width of harmonic peaks and is set to 1.8. $\tilde{X}_m(q_0)$ includes harmonic peaks at q_0 and peaks corresponding to multiples of q_0. Frequency components of $\tilde{X}_m(q)$ ranging from 60 to 404 Hz are selected as the feature.

To make use of temporal information, we splice a window of 5 frames of features as our final frame-level feature $\mathbf{y}_m$.

The feature in this study generalizes well to different noise types and also contains speaker-dependent information which can be leveraged in speaker-dependent DNN training.

2.2. DNN based pitch probability estimation

We adopt Han and Wang's idea [8] and use a DNN to estimate the posterior probability of pitch states given the frame-level feature vector, i. e., $p(x_m|\mathbf{y}_m)$. The input layer of the DNN corresponds to the frame-level feature vector of the mixture. There are three hidden layers, and each one has 1600 units with the ReLU activation function [6]. The reason why we choose ReLU instead of sigmoid is that it alleviates the vanishing gradient problem. Faster and effective training/adaptation can thus be performed. The output of the DNN is a softmax layer with 68 units, with each unit estimating the posterior probability of one pitch state. We use the cross-entropy cost function, standard backpropagation and dropout regularization [9] (dropout rate 0.2) to train the network. Mini-batch stochastic gradient descent along with a momentum term (0.9) is adopted for optimization. In Han and Wang's study [8], training data contains noisy utterances from 100 speakers, which can be denoted as a speaker-independent DNN or SI-DNN.

2.3. Speaker-dependent training of DNNs

Because everyone's speech has unique spectral patterns, an SI-DNN may not be optimal for all speakers. To model the characteristic of each enrolled speaker, we train speaker-dependent DNNs (SD-DNNs) for pitch-probability estimation. There are two ways to train SD-DNNs. If speaker-dependent training data is sufficient, we can train SD-DNNs using the same training recipe as for SI-DNNs. This method is denoted as SD-DNN-TRAIN. However, with limited training data, direct training of SD-DNNs may result in overfitting. To address this problem, we perform speaker adaptation for the SI-DNN, and denote it as SD-DNN-ADAPT.

Speaker adaptation of DNNs has been studied in automatic speech recognition for years. Two typical approaches include feature transformation [15] [22] and regularized retraining [1] [17]. Because of the limited training size in our study, we use regularized retraining for adaptation. Specifically, for each new speaker, we retrain the weights of the SI-DNN with a relatively small learning rate and a regularization term of 0.01 (L_2 regularization). We also examine two factors which may affect the adaptation process, i. e., which layer to

retrain and the size of adaptation data. In the end, adaptation is compared with direct training to show their advantages and shortages. Detailed experiments can be found in Section 3.

2.4. Hidden Markov model

After the estimation of posterior probabilities, we use an HMM to infer the most likely pitch track. The hidden variable of the HMM is the pitch state x_m, and the observation variable is the feature vector $\mathbf{y}_m$. Prior probabilities $p(x_m = s_i)$ and transition matrices are computed from training data directly. Because our training data is insufficient for building speaker-dependent HMMs, we use a speaker-independent HMM for all speakers. Emission probabilities are computed using estimated posterior probabilities and the Bayes rule.

In the next step, we apply the Viterbe algorithm [5] to connect all derived probabilities and generate the most likely pitch-state sequence. We then convert pitch states to mean frequencies of corresponding frequency bins, and use a moving average window of three frames to smooth the pitch track.

3. EVALUATION RESULTS AND COMPARISONS

We use the TIMIT [25] and the IEEE database [12] for experiments. The training set of the SI-DNN contains 1000 TIMIT utterances from 100 speakers. Three noises from NoiseX [19] are used during training: babble noise, factory noise, and high frequency radio noise. We mix the training utterances with the first half of all training noises at -5, 0 and 5 dB, resulting in a total of 9000 mixtures for the SI-DNN. We then use the IEEE database recorded by a male and a female speaker to train SD-DNNs. For each speaker, 10, 40, 160 and 640 training utterances are mixed with the first half of all training noises at -5, 0 and 5 dB, therefore four training sets with 90, 360, 1440 and 5760 mixtures are created. The test set contains 20 unseen utterances of each speaker in the IEEE database. The latter half of three training noises and three new types of noise, i. e., cocktail-party, crowd playground and crowd music noise [10], are used during test. Each test utterance is mixed with all six test noises at -10, -5, 0, 5 and 10 dB, thus 600 test mixtures are created for each speaker. Groundtruth pitches are extracted from clean utterances using Praat [2].

To evaluate pitch tracking results, we use two metrics: detection rate (DR) and voicing decision error (VDE) [14]. DR represents the percentage of voiced frames where estimated pitch deviates less than 5% from groundtruth pitch. VDE computes the percentage of frames where the pitched/unpitched decision is incorrect:

$$\mathbf{DR} = \frac{N_{0.05}}{N_p}, \quad \mathbf{VDE} = \frac{N_{n\to p} + N_{p\to n}}{N} \tag{2}$$

Here $N_{0.05}$ is the number of frames whose estimated pitch is within $\pm 5\%$ of groundtruth pitch. $N_{n\to p}$ and $N_{p\to n}$ are the number of frames mislabeled as pitched and unpitched respectively. N and N_p are the total number of frames and pitched frames respectively.

Table 1: Comparison of Han and Wang's system and our SI-DNN based method. Each value in the table is averaged across two speakers, three noise types and five SNR conditions. Boldface indicates the best result.

	Seen noises		Unseen noises	
	DR	VDE	DR	VDE
Han & Wang	0.646	0.216	0.677	0.212
Proposed SI-DNN	**0.738**	**0.193**	**0.748**	**0.195**

Table 2: Comparison of retrained layers during adaptation.

	Seen noises		Unseen noises	
	DR	VDE	DR	VDE
Input layer only	0.777	**0.194**	0.790	0.186
Output layer only	0.773	0.204	0.773	0.208
All layers	**0.787**	**0.194**	**0.800**	**0.179**

Table 3: Comparison of different training sizes for SD-DNN.

		Seen noises		Unseen noises	
		DR	VDE	DR	VDE
Direct training	90	0.737	0.207	0.759	0.194
	360	0.779	0.201	0.795	0.186
	1440	0.799	**0.193**	0.811	0.185
	5760	**0.803**	0.195	**0.814**	**0.183**
Adaptation	90	0.787	**0.194**	0.800	0.179
	360	0.792	0.195	0.805	**0.178**
	1440	**0.795**	0.202	**0.808**	0.194
	5760	0.794	0.207	0.805	0.202

We first present our baseline system: the SI-DNN based method. Table 1 compares Han and Wang's system with our SI-DNN based method. Although the two systems share the same framework and training corpus/noise, due to the larger training size and better training recipe of our work, we produce much better results in terms of both DR and VDE.

Next, different layers in the SI-DNN are retrained with 90 adaptation mixtures per speaker. As shown in Table 2, all three methods outperform the SI-DNN based method. Adaptation on all layers achieves the best performance for both seen and unseen noises. As a result, all future experiments of SD-DNN-ADAPT use all-layer adaptation.

Our next experiment investigates the relation between the training size and pitch tracking results. As shown in Table 3, the performance of SD-DNN-TRAIN improves with the increase of the training size. The improvement becomes small when the training size reaches 1440. For SD-DNN-ADAPT, the system produces pretty good results with 90 training mixtures. Further increasing the training size does not boost the performance significantly. The VDE of SD-DNN-ADAPT

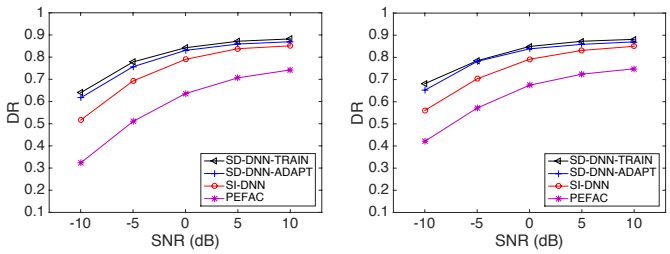

Fig. 2: (a) DR for seen noises. (b) DR for unseen noises.

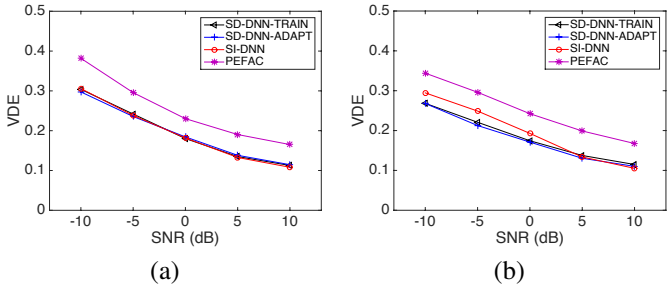

Fig. 3: (a) VDE for seen noises. (b) VDE for unseen noises.

starts to increase from 1440 mixtures. One possible explanation is that too many iterations of training lead SD-DNN-ADAPT to some local minima. Due to the small learning rate and the regularization term, the network lacks the ability to jump out of the local minima. The best performing system in terms of detection rate is SD-DNN-TRAIN trained with 5760 mixtures, and its corresponding VDE is also competitive. However, one should use SD-DNN-ADAPT when limited data is available for speaker-dependent training.

As Han and Wang [8] have shown substantial improvement over several other pitch tracking algorithms [7] [11] [13] [14], we choose to compare our methods, i. e., SI-DNN, SD-DNN-ADAPT trained with 90 mixtures and SD-DNN-TRAIN trained with 5760 mixtures, with only one of them: PEFAC [7], as a representative unsupervised method. As shown in Fig. 2, all proposed methods have much higher DR than PEFAC in all SNR and noise conditions. Both SD-DNN based methods substantially outperform SI-DNN. The improvement is higher when the SNR becomes low, which reflects the noise-robustness of speaker-dependent training. VDE results of our methods in Fig. 3 are also a lot better than PEFAC. SD-DNN-TRAIN and SD-DNN-ADAPT match the performance of SI-DNN for seen noises, and outperforms SI-DNN at negative SNRs of unseen noises.

Lastly, we compare pitch tracking results on a test sample of the female speaker. The speech in Fig. 4 is severely corrupted by noise, which makes the pitch tracking task almost impossible to accomplish. As shown in the figure, pitch extracted by SI-DNN deviates a lot from the groundtruth pitch. SD-DNN-ADAPT and SD-DNN-TRAIN significantly improve the pitch tracking performance by correctly capturing

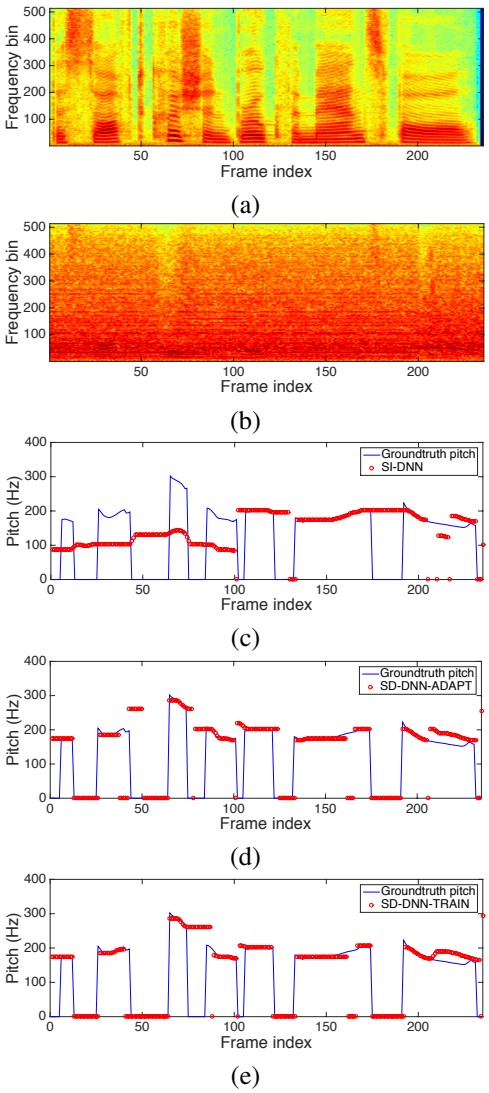

Fig. 4: Pitch tracking results of a female utterance, mixed with crowd music noise at -10 dB. (a) Spectrogram of clean speech. (b) Spectrogram of noisy speech. (c) SI-DNN based pitch contours. (d) SD-DNN-ADAPT based pitch contours. (e) SD-DNN-TRAIN based pitch contours.

pitch contours from frame 0 to 100. They also make right voicing decisions for frames 175 to 190.

4. CONCLUSION

We have proposed speaker-dependent DNNs for pitch probability estimation. When training SD-DNNs, speaker adaptation works well on small training sizes, and direct training performs better on large training sizes. They both outperform a speaker-independent DNN in all SNR conditions. To use our methods requires that the identity of the speaker be known beforehand. A noise-robust speaker identification algorithm proposed by Zhao *et al.* [24] can be used to help us select trained SD-DNNs for pitch tracking.

5. REFERENCES

[1] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proceedings of ICASSP*, 2013, pp. 7942–7946.

[2] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," in *Glot Int.*, vol. 5, 2001, pp. 341–345.

[3] C. Chen, R. Gopinath, M. Monkowski, M. Picheny, and K. Shen, "New methods in continuous mandarin speech recognition," in *Proceedings of Eurospeech*, 1997, pp. 1543–1546.

[4] A. D. Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Amer.*, vol. 111, pp. 1917–1930, 2002.

[5] G. D. Forney Jr, "The viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, 1973, pp. 268–278.

[6] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011, pp. 315–323.

[7] S. Gonzalez and M. Brookes, "PEFAC-A pitch estimation algorithm robust to high levels of noise," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, pp. 518–530, 2014.

[8] K. Han and D. L. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, pp. 2158–2168, 2014.

[9] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[10] G. Hu. 100 nonspeech sounds, 2006. [Online]. Available: http://www.cse. ohio-state.edu/pnl/corpus/HuCorpus.html

[11] F. Huang and T. Lee, "Pitch estimation in noisy speech using accumulated peak spectrum and sparse estimation technique," *IEEE Trans. Speech Audio Process.*, vol. 21, pp. 99–109, 2013.

[12] IEEE, "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust*, vol. 17, pp. 225–246, 1969.

[13] Z. Jin and D. L. Wang, "HMM-based multipitch tracking for noisy and reverberant speech," *IEEE Trans. Audio, Speech, Lang. Process.*, pp. 1091–1102, 2011.

[14] B. S. Lee and D. P. W. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proceedings of Interspeech*, 2012.

[15] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proceedings of ICASSP*, 2013, pp. 7947–7951.

[16] Y. Liu and D. L. Wang, "Speaker-dependent multipitch tracking using deep neural networks," in *Proceedings of Interspeech*, 2015, pp. 3279–3283.

[17] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, 2013, pp. 55–59.

[18] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding Synth.*, vol. 495, p. 518, 1995.

[19] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, pp. 247–251, 1993.

[20] D. L. Wang and G. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. Wiley-IEEE Press, 2006.

[21] M. Wohlmayr, M. Stark, and F. Pernkopf, "A probabilistic interaction model for multipitch tracking with factorial hidden Markov models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, pp. 799–810, 2011.

[22] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proceedings of ICASSP*, 2013, pp. 7893–7897.

[23] X. Zhao, Y. Shao, and D. L. Wang, "CASA-based robust speaker identification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, pp. 1608–1616, 2012.

[24] X. Zhao, Y. Wang, and D. L. Wang, "Robust speaker identification in noisy and reverberant conditions," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, pp. 836–845, 2014.

[25] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, pp. 351–356, 1990.