INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Direct F0 Estimation with Neural-Network-based Regression

Shuzhuang Xu[1], *Hiroshi Shimodaira*[2]

[1]School of Informatics, University of Edinburgh
[2]Centre for Speech Technology Research, University of Edinburgh
sz.xu@outlook.com, h.shimodaira@ed.ac.uk

Abstract

Pitch tracking, or the continuous extraction of fundamental frequency from speech waveforms, is of vital importance to many applications in speech analysis and synthesis. Many existing trackers, including conventional ones such as Praat, RAPT and YIN, and newly proposed neural-network-based ones such as DNN-CLS, CREPE and RNN-REG, have conducted an extensive investigation into speech pitch tracking. This work developed a different end-to-end regression model based on neural networks, where a voice detector and a newly proposed value estimator work jointly to highlight the trajectory of fundamental frequency. Experiments on the PTDB-TUG corpus showed that the system surpasses canonical neural networks in terms of gross error rate. It further outperformed conventional trackers under clean condition and neural-network classifiers under noisy condition by the NOISEX-92 corpus.

Index Terms: fundamental frequency, pitch tracking, neural network

1. Introduction

Fundamental frequency (F0) is an important acoustical feature determining the audible pitch level. A host of applications, such as speaker determination, audio processing, speech recognition and synthesis, can rely on a reliable F0 estimation heavily. How to accurately and automatically estimate F0 from speech waveforms then becomes a practical requirement. On the basis, *fundamental frequency determination algorithm* (F0DA), or pitch tracking algorithm, has been extensively investigated over past decades. Since F0 is a dynamic quantity over time, a common practice is dividing whole waveforms into many smaller and overlapped frames, where F0 is assumed to be static. Though the segmenting also introduces precision trade-off between time and frequency, most F0 tracking systems are basically working on this foundation.

Conventionally, F0DA is dominated by signal-processing-based and statistics-based algorithms [1]. Since the periodicity may be deduced from inspecting waveform or spectrogram, some common approaches include: time-domain based methods, e.g. YIN [2] and Praat [3], that typically make use of autocorrelation to find outstanding periods; frequency-domain based methods, e.g. SWPIE [4] and SHR [5], that work on some frequency features. Newer methods such as Bayesian-filter-based models also present a decent performance [6].

In the most recent decade, the feasibility of estimating F0 with neural networks has been validated through a number of models. The approach proposed by Han and Wang introduced both *deep neural network* (DNN) and *recurrent neural network* (RNN) to estimate a pitch state, still requiring a complex feature engineering as the pre-processing and Viterbi decoding as the backend [7, 8]. Following this framework, the work by Verma and Schafer removed feature extraction and Viterbi decoding, directly mapping waveforms to F0-corresponded states [9], which can be considered as an end-to-end classifier. As an alternative, the work by Kim et al. replaced DNN and RNN by *convolutional neural network* (CNN), known as the CREPE pitch tracker [10], which revealed the feature extraction capability within 1D-CNN. However, all these neural-network-based models transform the pitch tracking task into a classification task, which induces 2 constraints: the numerical output is quantised, leading to a certain systematic error; and the target vector in classification forces the probabilities of every pitch state except the correct one to be 0, which does not reflect the difference between making a small error (e.g. to an adjacent frequency state) and making a large error (e.g. to incorrect decision on F0 existence). By contrast, the most recent model investigated by Kato and Kinnunen presented an outstanding robustness under very noisy condition by RNN-based identity mapping, where the neural network is trained to be a noise filter [11]; however, their model still requires autocorrelation as its backend to determine F0 value as well as F0 existence, which includes an empirical boundary condition to be manually determined.

Our work insists on the end-to-end framework that directly interprets F0 from the speech waveforms, but further replaces quantised pitch states by a continuous numerical output without any complex pre-processing and backend utility. We first manage to identify the latent obstacle that constrains the performance under this end-to-end context, and propose to employ a dedicated voice detector and value estimator with neural networks. In our experiment, the voice detector is best achieved by a deep feedforward neural network with dropout and batch normalisation; while the value estimator is best achieved by our proposed value decoder, where the resultant value is composed of distributed representations from multiple output nodes.

This paper is organised in the following order: chapter 2 presents the framework of the system, as well as how the voice detector and value estimator are built; chapter 3 presents the experiment configuration, result and discussion; finally, a conclusion summarises the work.

2. Methodology

Given waveform frames, the F0 estimation involves 2 sub-tasks: a classification determining whether the frame contains a voice or not; and a regression estimating the F0 value. A fundamental implementation is achieved by using a single model for both, targeting the reference F0 values for *voice frames* and 0 for *"unvoice" frames* (any frames not presenting a valid voice F0). However, this strategy may be performance-limited since: varying target values instead of probabilities are not appropriate for classification, causing an unstable boundary between voice and unvoice; and the zeros for unvoice denote an infinite large F0 period, which should not be included in the training data for numerical regression. As a consequence, the model may be trained

http://dx.doi.org/10.21437/Interspeech.2019-3267

to a balancing point between classification and regression, incapable of maximising its power on each sub-task. Some models transform the whole task into a pure classification, therefore avoid this conflict; however, it still brings new constraints as described in Introduction.

Based on the consideration, we supposed the shared hidden representation may prevent from better performance and proposed to use a pair of dedicated F0 detector and estimator. For each frame input, the detector determines the F0 existence: if voice F0 is present, the estimator then evaluates the F0 value; otherwise, a zero is produced to be compatible with conventional systems. Under this framework, the detector and estimator are only involved in one sub-task respectively and can be trained independently.

2.1. Voice detection

The voice detection is formalised as a binary classification. The probability of a frame being either voice or unvoice frame is represented in two states respectively. Therefore, the target value is 1 or 0 instead of F0. Accordingly, activation function e.g. *sigmoid* is applicable for the output layer and loss e.g. *binary cross-entropy loss* is compatible for training.

We implemented *fully-connected deep feedforward network* (FDNN) as well as its improvement by introducing dropout [12] and batch normalisation [13]. Dropout is a simple but effective technique that partially masks layer units as zeros in training time, attenuating the co-dependency between units and hence constraining overfit. Batch normalisation is a regularisation technique that re-scales tensors in a mini-batch by learnable parameters to centralise values and stabilise variance, reducing the network difficulty to learn new distributions of incoming tensors. Apart from this, we explored the possibility by the more recently innovated residual network on this classification. Residual network introduces the connection between non-adjacent layers and changes the learning objective of the vaulted layer to the difference between source and target. A pair of popular implementations including the earlier model [13] and the later model [14] are implemented.

In addition, simple pre-processing is used. Since adjacent frames may include supportive information to determine F0 existence, the length l of the input frame x is enlarged to p times by symmetrically enclosing sampling points from both sides. Apart from this, since the loudness varies among recordings, each whole recording X is normalised to $X' = X \cdot \frac{1.0}{M(|X|)}$ where $M(\cdot)$ computes the absolute average value. Note that the pre-processing above is applied for the voice detector only.

2.2. F0 estimator

The F0 estimation is formalised as a numerical regression. Though the networks for voice detection should be applicable as an F0 estimator by altering the output layer to $ReLU$ activation [15], we conduct a further investigation to seek better improvement. Inspired by "jump wire" techniques such as highway and residual networks that shorten the distance between shallow layers and deeper layers, as well as how an exponential series represents a natural number n: $n = \sum_{i=0}^{\infty}(c_i \times 2^i)$, where $c_i \in \{0, 1\}$, we consider every F0 value $\hat{f}$, instead of being derived from a single output layer, can be assembled from multiple output nodes:

$$\hat{f} = \sum_{i=1}^{n} k_i \cdot o_i + b \tag{1}$$

where $o_1, o_2 \cdots o_n \in [0, 1]$ are values from output nodes with *sigmoid* activation; k_i, b and n are pre-defined constants such that the equation's lower and upper bound $[b, b + \sum_{i=1}^{n} k_i]$ covers the practical F0 range $[f_{min}, f_{max}]$. We then proposed a network, *value decoder* (VD), as illustrated in Figure 1, where the outputs $o_1, o_2 \cdots o_n$ obtained in a forward propagation are given by:

$$o_i = O(H_i(\cdots H_2(H_1(F(x)))\cdots)) \tag{2}$$

where: x is the input frame from a whole recording X; $F(\cdot)$ is an arbitrary combination of layers including the input layer; $H_1(\cdot), H_2(\cdot), \cdots H_n(\cdot)$ are hidden layers of either regular ones e.g. fully-connected, or RNN cells e.g. LSTM [16] and GRU [17]; and $O(\cdot)$ is the output layer with shared parameters and sigmoid activation to derive $o_1, o_2 \cdots o_n$.

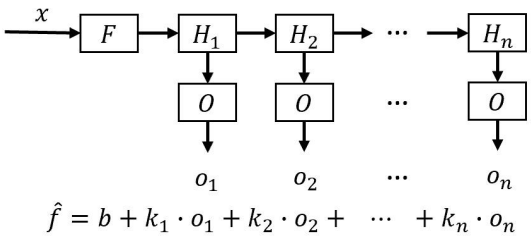

Figure 1: *The diagram of the proposed value decoder*

This model makes the final F0 no longer inferred from a single output layer but a computation from layers in different depths i.e. a distributed representation. It can be trained by a regular numerical loss such as *L1* and *mean squared error* (MSE) loss.

3. Experiment

The system performance is evaluated in terms of accuracy on voice decision and value estimation. Comparison is made vertically and horizontally: vertically, the candidate models for detector and estimator are compared; horizontally, the system assembled with the best detector and estimator is compared with external models.

3.1. Pitch tracking corpus

The latest pitch tracking corpus from the Graz University of Technology (PTDB-TUG) [18] is adopted for experiments, which surpasses previous corpora in terms of quality and abundance. This database consists of 236 sentences selected from the TIMIT corpus based on phonetic richness. Each sentence is spoken by 10 women and 10 men, giving a total of 4720 pieces of recordings. The reference F0 values are derived from the coupled laryngograph, labelled for every frame of 32 milliseconds (adjacent frame is overlapped by 22 milliseconds). The total number of training pairs is 3420k and 807k of which are of voice. F0 in the database covers a wide range from about 50Hz to 380Hz, with mean, median and standard deviation at around 151Hz, 148Hz and 52Hz respectively. The only pre-processing on the database is a down-sampling to 12kHz instead of using the original 48kHz sampling rate. For dataset division, 80% of the total are formed into a 4-fold cross validation set for training and validation, and 20% is held-out for test. In addition, the robustness is investigated under a simulated noise condition by mixing standard noise from the NOISEX-92 corpus [19] onto

Table 1: *Network specification for vertical comparison*

Models	Specification
FDNN-1	FDNN with 6 hidden layers of 1024, 896, 768, 640, 448 and 384 units respectively using ReLU activation
FDNN-2	Same as FDNN-1 but each hidden layer has 25% dropout and batch normalisation
FDNN-3	Same as FDNN-1 but used for voice detection and F0 estimation in a single model
ResNet-1	3 residual cell [13] of width 1024
ResNet-2	3 residual cell [14] of width 1024
VD-FNN	Value decoder using FDNN (width 1024)
VD-RNN	Value decoder using LSTM (width 1024)

the clean waveform. 5 different noises, babble, factory1, destroyerops, leopard and white, selected based on sound characteristic, are applied at 3 different levels of *signal-to-noise ratio* (SNR): -10dB, 0dB and 10dB.

3.2. Experiment configuration

For the voice detector, it is trained on all training pairs including both voice and unvoice frames. For the F0 estimator, it is trained solely on all voice pairs as the hypothesis suggests unvoice frames are not suitable for estimator. L1 loss is adopted in order to better constrain the overall inaccuracy, instead of MSE loss which will better constrain pairs with larger errors. The window enlarging ratio q is 3 for detector and 1 for estimator (i.e. not changed).

In the vertical benchmark, experiment is first conducted to find the optimal hyperparameters k, b and n for the proposed value decoder, where $b = 0$ and $k, n = \{100, 4; 133, 3; 200, 2; 400, 1\}$ that all match a range between 0Hz and 400Hz are tested. Then, the value decoder with the best hyperparameters is compared with other neural networks with key specifications listed in Table 1. As a note, the model FDNN-3 is trained to work on both voice and unvoice frames. Therefore, it has only one unit for the output layer, where the output is the F0 value for voice frames and 0 for unvoice frames by $ReLU$) (any output lower than 30Hz is also regarded as 0Hz). The purpose of this particular model is to validate the hypothesis that using separate models will improve the performance.

For the horizontal benchmark, the assembled model with the best detector and estimator is compared with conventional F0 trackers including Praat [3], RAPT [20] and YIN [2] under clean condition as well as the representative classifier DNN-CLS [9] under simulated noisy condition. The DNN-CLS here is modified to have the same structure as FDNN-2 (the width of the input layer is 1024) and a frequency state for every integer rounded from reference F0 (plus one state for unvoice). Anything else is implemented as the original configuration as best as possible, except that the input waveform is adjusted to use the original width and interval from the PTDB-TUG corpus.

3.3. Evaluation metrics

Pitch tracking performance is evaluated on standard metrics:

- **Voice decision error (VDE)**: This includes all frames which are incorrectly classified, either voice frames classified as unvoice or unvoice as voice.
- **Gross pitch error (GPE)**: This includes all voice frames with a relative error larger than 20% of its reference F0. A narrow definition is used here that gross pitch error excludes any frames that already trigger VDE.
- **F0 frame error (FFE)**: This includes all frames that trigger either VDE or GPE, which can be used as a combined measure of the system performance. Note that the rate of FFE equals to the sum of VDE and GPE.
- **Fine pitch error (FPE)**: This includes all remaining voice frames that triggers neither VDE nor GPE. The average μ_{FPE} and standard deviation σ_{FPE} of the absolute difference between the derived and referencing F0 describe the level of value accuracy on F0 estimation.

3.4. Result

Table 2 gives the performance of the proposed value decode on different sets of hyperparameters on FDNN. It is shown that using the set $k = 100$, $b = 0$ and $n = 4$ presents a lower GPE. A discussion for it is placed at the end of the chapter.

Table 2: *Value decoders with different hyperparameters*

(b=0)	n=1 k=400	n=2 k=200	n=3 k=133	n=4 k=100
GPE	2.81	2.33	2.21	**2.14**

Using the best configuration above, two variants of value decoder are further implemented in vertical comparison: using FDNN and LSTM as the cell respectively. The result of the vertical benchmark for all candidates as voice detector or F0 estimator is presented in Figure 3. The comparison between FDNN-2 and FDNN-3 shows that all performance metrics degrades in the latter, indicating an accurate regression from raw waveform to numerical F0 might not be achievable by the shared model, which proves the feasibility by using dedicated models for F0 detection and estimation. In other words, our hypothesis, that the shared hidden representation for F0 detection and estimation is an obstacle for better performance, holds in the context of direct numerical regression.

Table 3: *Vertical comparison across candidate models.*

Models	Detector	Estimator		
	VDE	GPE	μ_{FPE}	σ_{FPE}
FDNN-1	3.85	2.73	2.59	3.19
FDNN-2	**3.40**	2.51	2.15	3.11
FDNN-3	3.51	3.10	4.57	3.90
ResNet-1	3.82	2.36	1.97	3.10
ResNet-2	3.74	2.57	2.40	3.47
VD-FNN	-	**2.14**	1.47	2.66
VD-RNN	-	2.44	**1.42**	**2.63**

Among these dedicated models for detection and estimation, it is obvious that the FDNN-1 presents the overall worst performance, while other models all present some improvement. For F0 detection, it shows that both residual network structures have very little help for better determination on F0 existence with ignorable difference. On the contrary, dropout and batch normalisation efficiently improves voice detection by 0.45% dropping in VDE. For F0 estimation, it shows that the residual network proposed by [13] achieves all metrics better than DNN with dropout and batch normalisation, while the alternative residual structure proposed by [14] underperforms the former. However, all these models mentioned above does not

outperform the proposed value decoder on F0 estimation. It is found that using RNN cell for this structure obtains slightly better value accuracy on fine pitch frames, with the mean and standard derivation decreased by 0.05% and 0.03% respectively. Even though, on GPE, the RNN-based value decoder is inferior to the feedforward-network based value decoder, where the latter obtains a GPE at 2.15%, absolutely 0.29% better than the RNN-based one. Therefore, based on the performance scores, the final system is assembled with deep neural network with dropout and batch normalisation as the voice detector and feedforward-network-based value decoder as the value estimator.

Figure 2 presents the result between conventional models and our system under the clean PTDB-TUG corpus. Apparently, the result presents our model significantly outperforms other contrastive ones. The improvement is accumulated from better VDE which decreases by around 25% compared with RAPT, as well as better GPE which is almost 1/3 of that of YIN, to an overall FFE 30% better than other best performed model (Praat).

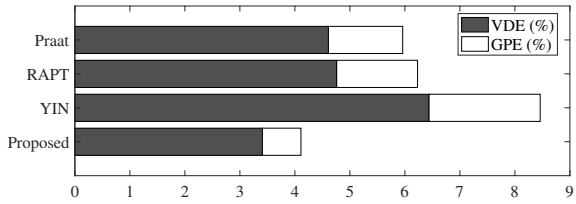

Figure 2: *The comparison between conventional models and the proposed model under clean condition*

Figure 3 demonstrates the result of the modified DNN-CLS model as well as the proposed model under simulated noisy condition. With respect to voice decision accuracy, it shows that the proposed system outperforms DNN-CLS in every experimented noise level, with an absolute improvement at 0.42%, 0.83% and 2.2% under 10dB, 0dB and -10dB SNR respectively. However, DNN-CLS constantly preserves a better gross value accuracy in F0 estimation, surpassing the proposed model by an absolute value of 0.87%, 1.68% and 1.69% respectively under these noise levels. The overall measure in terms of FFE is very close between the proposed model and the contrastive one, where the former is slightly better only in very noised condition (-10dB). This result indicates that the dedicated voice detection model in the proposed system is better than the pure classification network; while the proposed value decoder is not as good as the classification model in both model capability and noise robustness. Since the hidden structure of the modified DNN-CLS model is identical to the voice detector in the proposed system, it also proves that the shared hidden representation degrades the performance.

Table 4: *Value decoder behaviours*

F0 (Ref.)	o_1	o_2	o_3	o_4
66	0.65	0.02	0.00	0.00
151	0.97	0.53	0.01	0.00
234	1.00	0.96	0.42	0.00
309	1.00	0.99	0.75	0.32

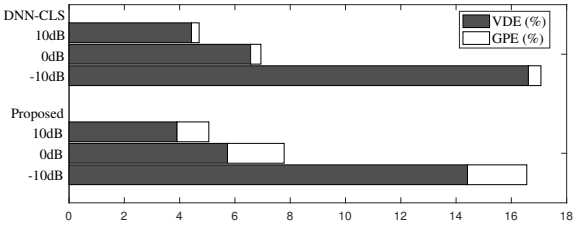

Figure 3: *The comparison between the modified DNN-CLS and the proposed model under 3 levels of simulated noisy condition*

Last but not least, the mechanism of the proposed value decoder deserves a discussion. Some samples from a trained value decoder with $k = 100$, $b = 0$ and $n = 4$ is presented in Table 4. As expected, the inferred F0 is not derived from a single output layer, instead, from distributed output nodes. There are 2 functional parts in this structure: "subtractor" and "estimator". The hidden layers $H_1(\cdot), H_2(\cdot), \cdots H_n(\cdot)$ are trained to be subtractors. Through each subtractor, an abstract hidden representation for kHz is subtracted from the incoming tensor. The shared output layer $O(\cdot)$ is trained to be a numerical estimator. If the hidden vectors contain a frequency higher than kHz or "negative" frequency, the *sigmoid* activation can prevent the value from out-of-control. This structure has some interesting similarity with known models: it introduces shortcut connections from shallow layers to deep layers, as Highway network and residual network; and it visually looks like a time-flattened decoder in a canonical sequence-to-sequence model, while employing no loops from the output back to the input. By employing this design, shallow layers in this structure also have output layer closely connected, allowing back-propagation functions more effectively. The difficulty in estimating an accurate value in a time is decomposed into multiple output nodes and more instructive hints can be backpropagated to layers. They are considered as the key reasons of its success.

4. Conclusion

Our work conducts a successful investigation on how to directly estimate speech F0 from raw waveform with neural networks. It concludes that using dedicated models for voice detection and value estimation is a positive strategy to improve the overall performance. For voice detection, DNN with dropout and batch normalisation is shown to be very efficient. For F0 estimation, we proposed a good-performing value decoder structure, which outperforms both DNN and residual networks on all performance metrics. Constructed with the best components, the final system significantly outperforms traditional models in terms of both voice decision and pitch accuracy. Compared with representative neural-network classifier for pitch tracking, the system still preserves a competitive accuracy on voice decision under very noised condition; however, the value accuracy of the value decoder is not as robust as the contrastive model. By the decomposition in the pitch tracking task as well as the proposed value decoder, we demonstrated the effectivity of decomposing a complex task into simpler tasks, which is also applicable to other tasks. The feasibility of the proposed value decoder on other regression task is an open area of research.

5. References

[1] S. Strömbergsson, "Today's most frequently used f0 estimation methods, and their accuracy in estimating male and female pitch in clean speech," in *INTERSPEECH*, 2016.

[2] H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, 2002.

[3] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *the Institute of Phonetic Sciences*, 1993, pp. 97–110.

[4] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," in *The Journal of the Acoustical Society of America (JASA)*, vol. 124, no. 3, 2008, p. 16381652.

[5] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2002, pp. 333–336.

[6] H. Hajimolahoseini, R. Amirfattahi, S. Gazor, and H. Soltanian-Zadeh, "Robust estimation and tracking of pitch period using an efficient bayesian filter," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 7, pp. 1219–1229, 2016.

[7] K. Han and D. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 12, pp. 2158–2168, 2014.

[8] ——, "Neural networks for supervised pitch tracking in noise," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1488–1492.

[9] P. Verma and R. W. Schafer, "Frequency estimation from waveforms using multi-layered neural networks," in *INTERSPEECH*, 2016.

[10] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161–165, 2018.

[11] A. Kato and T. Kinnunen, "Waveform to single sinusoid regression to estimate the f0 contour from noisy speech using recurrent deep neural networks," in *INTERSPEECH*, 2018, pp. 327–331.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[14] H. Kawahara, Y. Agiomyrgiannakis, and H. Zen, "Using instantaneous frequency and aperiodicity detection to estimate F0 for high-quality speech synthesis," *CoRR*, 2016.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[17] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[18] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *Interspeech*, 2011, pp. 1509–1512.

[19] A. Varga and H. J. 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," in *Speech Communication*, vol. 12, no. 3, 1993, pp. 247–251.

[20] D. Talkin, "A robust algorithm for pitch tracking (rapt)," in *Speech Coding and Synthesis*, 1995, pp. 495–518.