INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

LACOPE: Latency-Constrained Pitch Estimation for Speech Enhancement

Hendrik Schröter[1], *Tobias Rosenkranz*[2], *Alberto N. Escalante-B.*[2], *Andreas Maier*[1]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Pattern Recognition Lab, Germany
[2]Sivantos GmbH, Research and Development, Erlangen, Germany

hendrik.m.schroeter@fau.de

Abstract

Fundamental frequency (f_0) estimation, also known as pitch tracking, has been a long-standing research topic in the speech and signal processing community. Many pitch estimation algorithms, however, fail in noisy conditions or introduce large delays due to their frame size or Viterbi decoding.

In this study, we propose a deep learning-based pitch estimation algorithm, LACOPE, which was trained in a joint pitch estimation and speech enhancement framework. In contrast to previous work, this algorithm allows for a configurable latency down to an algorithmic delay of 0. This is achieved by exploiting the smoothness properties of the pitch trajectory. That is, a recurrent neural network compensates delay introduced by the feature computation by predicting the pitch for a desired point, allowing a trade-off between pitch accuracy and latency.

We integrate the pitch estimation in a speech enhancement framework for hearing aids. For this application, we allow a delay on the analysis side of approx. $5\,\mathrm{ms}$. The pitch estimate is then used for constructing a comb filter in frequency domain as post-processing step to remove intra-harmonic noise.

Our pitch estimation performance is on par with SOTA algorithms like PYIN or CREPE for spoken speech in all noise conditions while introducing minimal latency.

Index Terms: pitch estimation, speech enhancement, convolutional recurrent neural network

1. Introduction

Removing unwanted environmental noise is a common feature of modern hearing aids. An important property of hearing aid processing is a low overall latency, which includes analysis, filtering steps like noise reduction as well as synthesis. Especially for hearing aids with open coupling, a strong component of the original signal is reaching the ear drum. Thus, delays greater than $10\,\mathrm{ms}$ are generally undesirable [1] since they introduce unwanted comb filter effects[1]. These latency requirements result in fairly short processing windows of approx. $6\,\mathrm{ms}$ and frequency bandwidth of $500\,\mathrm{Hz}$. Due to this frequency resolution, it is not possible to reduce intra-harmonic noise resulting in a rougher sounding signal compared to clean speech. To be able to attenuate the noise between the speech harmonics a comb filter was recently proposed [2, 3]. Valin et al. [2] estimate pitch similarly to the OPUS codec [4] with an autocorrelation based approach. However, these methods use at least $20\,\mathrm{ms}$ frames for pitch analysis and are thus not feasible for our latency constraints.

Other pitch estimation algorithms require a similar or often even higher look-ahead. RAPT [5] also uses normalized cross correlation (NCC) features combined with a maximum search and dynamic programming for selecting the best f_0 candidate. Dynamic programming improves robustness by exploiting smoothness properties of pitch and is thus adopted by many approaches [5, 6, 7, 8]. However, its full potential is only utilized if at least a few steps of the Viterbi backward algorithm are computed which results in an additional delay of appox. $100\,\mathrm{ms}$ [5]. YIN, as well as its probabilistic successor PYIN [9, 6] use cumulative mean normalized difference function (CMN DF) instead of NCC since this is supposed to help with octave errors. Typically, both require frame sizes of at least 20 to $100\,\mathrm{ms}$. PYIN needs an additional look-ahead for dynamic programming. CREPE [7] is a deep learning approach based on time domain convolutions with a frame size of $64\,\mathrm{ms}$, slightly outperforming PYIN. Zhang et al. [10] also present a joint pitch estimation and speech enhancement framework. However, they only use pitch features as input of a denoising network.

In this work, we propose LACOPE, a latency-constrained pitch estimation approach based on deep learning with several advantages over SOTA pitch estimation algorithms. First, our algorithm allows for a configurable latency between 0 to $20\,\mathrm{ms}$, while larger latencies result in a higher pitch accuracy. Second, our algorithm produces robust pitch estimations also for low SNRs as well as periodic noises. This is accomplished through extensive data augmentation with a huge variety of noise signals during training as well as a multi-target loss for pitch estimation and noise reduction. Finally, the complexity is tremendously lower compared CREPE.

2. Signal Model

Let $x(k)$ be a mixture signal recorded in a noisy room.

$$x(k) = s(k) \star h(k) + n(k) \tag{1}$$

where $s(k)$ is a dry clean speech signal, $\star$ denotes the convolution operator, $h(k)$ a room impulse response (RIR) from the speaker to the microphone and $n(k)$ additive noise. Including reverberant speech $s^{rev} = s(k) \star h(k)$ in the signal model is important for generalization on real world signals. Furthermore, the periodic component of reverberant speech is usually slightly degraded. A comb filter could then improve the perceptual quality, by improving the periodic parts.

Our noise reduction approach operates fully in frequency domain. Therefore, we use a standard uniform polyphase filter bank resulting in the following signal model:

$$X_b(l) = S_b(l) \cdot H_b(l) + N_b(l) \tag{2}$$

with $b \in \{0 \dots \mathrm{B} - 1\}$ being the frequency bands and l the frame index. Due to our real-time requirements for hearing aids, the analysis filter bank (AFB) operates on approx. $6\,\mathrm{ms}$ frames with a subsampling rate of 24. This results in $\mathrm{B} = 48$ bands with a bandwidth of $500\,\mathrm{Hz}$. Thus, typical hearing aid noise reduction algorithms are only able to attenuate the overall spectral envelope and not enhance the periodic parts of speech [11, 12].

[1]Not to be confused with the digitally applied comb filter for intra-harmonic noise reduction

http://dx.doi.org/10.21437/Interspeech.2021-633

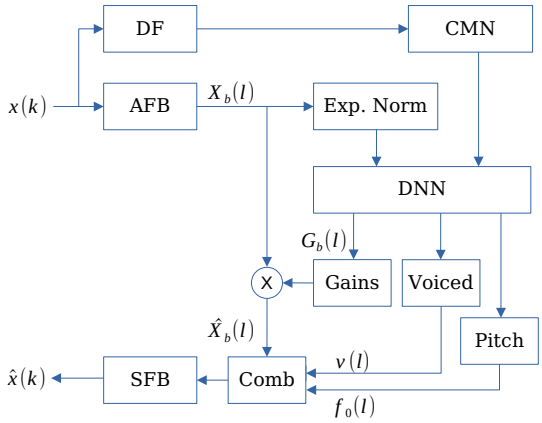

Figure 1: *Proposed joint pitch estimation and speech enhancement algorithm overview.*

Similar to [3], our noise reduction algorithm operates in two steps, as shown in Fig. 1. First, the overall spectral envelope is modeled by estimated band gains G_b resulting in an enhanced spectrogram $\hat{X}_b(l) = X_b(l) \cdot G_b(l)$. This also includes slight dereverberation by suppressing the late reflections. Next, a comb filter is applied in frequency domain and weighted given a voiced probability estimate to improve the periodic component in $\hat{X}_b$. We utilize a convolutional recurrent neural network to predict gain, pitch and voiced probability estimates.

2.1. Difference function

To provide good features for the pitch estimation task of the network, we compute the cumulative mean normalized difference function as described by [9]

$$d(\tau) = \sum_{j=\tau_{\min}}^{\tau_{\max}} (x_j - x_{j-\tau})^2 \,, \tag{3}$$

where $\tau_{\max}$ and $\tau_{\min}$ correspond to the lags of 60 Hz and 500 Hz, which mark the minimum and maximum pitch frequencies that our algorithm searches for. The difference function is then normalized using the cumulative mean

$$d'(\tau) = \frac{d(\tau)}{\frac{1}{\tau}\sum_{j=\tau_{\min}}^{\tau} d(j)} \,. \tag{4}$$

We chose a frame size of 20 ms and aligned the features in time as shown in Fig. 3. While for our application, a maximum look-ahead of $L = 5$ ms is acceptable, we tested several look-aheads from 0 to 20 ms.

2.2. Spectrogram normalization

Normalization is generally an important component for making a deep neural network (DNN) robust for unseen input data. Therefore, we transform the spectrogram into decibel scale and perform exponential normalization [13] while only ensuring zero mean. We found that unit variance does not provide any performance or generalization improvement.

$$X_{b,\text{norm}}[l] = X_{b,\text{dB}}[l] - \hat{\mu}_b[l] \tag{5}$$

The mean estimate $\hat{\mu}$ is given by

$$\hat{\mu}_b[l] = \alpha\hat{\mu}_b[l-1] + (1-\alpha)X_{b,\text{dB}}[l] \,, \tag{6}$$

where α corresponds to a normalization window of 3 s.

3. Comb filter

A comb filter uses constructive and destructive interference resulting in a comb-like frequency response by adding a delayed version of the input signal. Comb filters have recently proven to enhance the overall perceptual quality by reducing inter-harmonic noise [2, 3]. Typically, the comb filter is computed in time domain (TD) and defined as

$$y[k] = \frac{x[k] + x[k - T[k]]}{2} \tag{7}$$

where $T[k] \in \mathbb{N}^+ = \text{round}(f_s/f_0[k])$ is the pitch period corresponding to pitch $f_0[k]$ at time step k. In speech applications, the sampling frequency f_s is high enough such that sampling errors w.r.t. pitch period are negligible.

However, instead of applying the comb filter in TD, we operate fully in frequency (filter bank) domain FD. This has several advantages. First, it allows prior preprocessing in the hearing aid like beamforming. In this case, only the FD signal is available and an additional transformation into TD would introduce additional latency. In addition, we only have the pitch estimate after DNN processing, where the filter bank delay had already been introduced. Most importantly, we can apply the comb filter on the already enhanced spectrogram $\hat{X}_b$ instead of the unprocessed TD signal. This facilitates the decision to which the comb filter should be applied.

Since the analysis window is only appox. 6 ms long, the comb filter in FD needs to be applied in the same way as in TD.

$$Y_b[l] = \frac{X_b[l] + X_b[l - T'[l]] \cdot e^{-j\omega_k \tau}}{2} \,, \tag{8}$$

where $T'[l] = \text{round}(T^*[l]) = \text{round}(sr/f_0[l]/R)$ is reduced by the subsampling factor R. To compensate for the lower sampling rate in filter bank domain, we need a phase correction factor $e^{-j\omega_f \tau}$. It shifts the FB representation depending on the center frequency of band b and the residual delay $\tau = T^*[l] - T'[l]$.

Since the comb filter can only provide a benefit for periodic components of speech, we need an estimate for the periodicity for frame l. Therefore, we estimate the voiced probability that allows to define a weighting between stochastic and periodic components, where the comb filter should only be applied on periodic frames

$$\hat{X}'_b = \hat{X}_b \cdot (1 - v) + \text{comb}(\hat{X}_b, T) \cdot v \,. \tag{9}$$

Additionally, v could be locally reduced depending on the local SNR estimate via G_b. This ensures that the comb filter does not attenuate clean speech.

We also tried higher order comb filters like [3]. However, since we cannot use future taps due to the latency requirements, the resulting group delay is not feasible anymore.

4. DNN Model

We utilize a convolutional recurrent network (CRN) with two encoders as well as separate output heads for gains G, pitch f_0 and voiced probability v. The overall structure is shown in Fig. 2. We use time aligned convolutions which do not introduce any delay as shown in Fig. 3 in both spectrogram and difference function (DF) encoder. In contrast to the DF encoder, the spectrogram encoder does not include the last convolution and maxpool layer to not reduce frequency information too early.

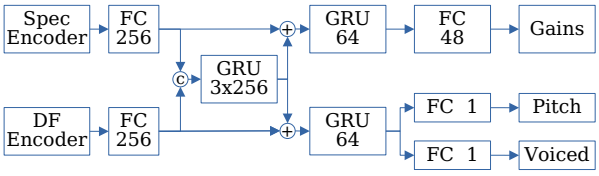

Figure 2: *DNN overview.* ⓒ *denotes the concatenation operator and* ⊕ *addition. The embedding GRU consists of 3 layers, the numbers below the layers represent the output hidden units.*

4.1. Loss function

We use a combined loss of 3 different loss functions for multi-target optimization.

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_p + \mathcal{L}_{ft}\,, \tag{10}$$

where $\mathcal{L}_g$ is a loss for the gains, $\mathcal{L}_p$ penalizes pitch $\hat{f}_0$ and voiced $\hat{v}$ estimates errors and $\mathcal{L}_{ft}$ is a loss based on the filtered time-domain signal given the ground truth and pitch and voiced probability. We use ideal amplitude mask gains as target gains [14] and adopt the gain loss from [3]. This loss combines a conventional L_2 with a L_4 term to penalize over attenuation and degradation of speech.

$$\mathcal{L}_g = \sum_b (g_b^\lambda - \hat{g}_b^\lambda)^2 + C_4 \sum_b (g_b^\lambda - \hat{g}_b^\lambda)^4\,, \tag{11}$$

where $\lambda = 0.5$ is a constant to match the perceived loudness and $C_4 = 10$ a balancing factor.

The pitch loss combines a weighted L_1 loss on the pitch and an L_2 loss on the voiced probability. We found that the L_1 loss on the pitch is a lot more robust than an L_2 loss due to prediction or ground truth outliers.

$$\mathcal{L}_p = C_p |\text{¢}(f_0) - \text{¢}(\hat{f}_0)| \cdot v + (v - \hat{v})^2\,, \tag{12}$$

where C_p is balancing factor and $\text{¢}(f) = 1200 \log_2(f/f_{ref})$ is the pitch measured in cent compared to $f_{ref} = 10\,\text{Hz}$. The pitch term is weighted by $C_p = 10^{-3}$ and the target voiced probability v to emphasize frames where the target is "sure" and disregard unvoiced frames.

Additionally, we utilize a time domain loss to indirectly improve on both tasks. The motivation for this is as follows. The comb filter can only improve the perceptual quality in periodic speech parts and should not be applied to other frames. On the other hand, a comb filter introduces a group delay of $T/2$ which makes it not feasible to use the original clean speech as target. This would force the voiced probability estimate $\hat{v}$ to 0 and the comb filter would not be applied at all. Thus, we filter the original clean speech spectrogram given the target pitch and voiced estimates like in Eq. 9 and then compute an L_1 loss in TD as suggested by [15]. Both target and speech estimate are transformed into time domain via the synthesis filter bank. The filtered time-domain loss,

$$\mathcal{L}_{ft} = |x' - \hat{x}'| \tag{13}$$

results in two effects. First, a wrong pitch estimate would result in a different frequency response from the comb filter, resulting in degraded speech where the effective error increases linearly with the frequency. In this cast, the network can improve by either estimating a better pitch, or, if this is not possible, by reducing $\hat{v}$ to at least not degrade the speech. The other effect is a frequency depending penalization of $\hat{v}$. For low fundamental frequencies, the effect of a comb filter decreases. The harmonics are much closer together resulting in a better frequency-local SNR. Thus, the network will reduce the effect of the comb filter the lower the pitch estimate is.

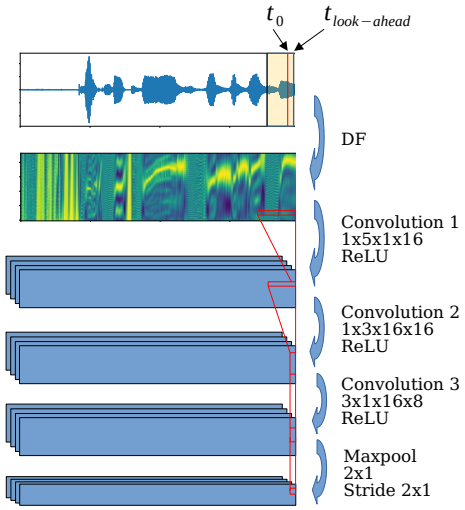

Figure 3: *Convolutional encoder for DF features. An audio frame (top, indicated in yellow) is aligned such that it corresponds to time position* t_0*. The look-ahead for this frame is* $L = t_{look\text{-}ahead} - t_0$*. Further convolutions are also aligned in time such that they don't introduce additional latency. Convolution weights (second row) represent kernel size in frequency and time axis as well as input and output channels.*

4.2. Training Data

We train our model on an extensive amount of speech and noise data to ensure good generalization. As speech datasets, EUROM [17], VCTK [18] and LJ speech [19] were used. Noise was included from DEMAND [20], RNNoise dataset [2] as well as the MUSAN corpus [21]. The later also contains music, which we consider as noise type. Including harmonic noise types like engine noise and music during training makes the pitch estimation more robust for real world signals. Also, we augment 30 % of the speech samples with a room impulse response (RIR) from either the Aachen RIR dataset [22] or a randomly simulated RIR via the image source model using [23]. All these datasets are split into train, validation and test sets with a 70/15/15 split. We mix all speech recordings randomly with up to 4 noises at SNR levels of [-5, 0, 10, 20, 100].

The ground truth pitch for training was estimated with PYIN [6] based on the clean speech. While this pitch estimate is not perfect, it provides a sufficiently accurate target pitch for our application. Interestingly, the DNN ended up being more robust on the noisy training data for instance w.r.t. octave errors or in reverberant conditions compared to the target pitch from the clean signal.

We evaluate our method using the pitch tracking data base PTDB-TUG [24] which contains over 4600 samples from unseen speakers. The ground truth pitch provided by PTDB was derived using RAPT [5] applied to a laryngograph recording that only captures the periodic parts of speech. We use noise from our test set at the same SNR levels. RIR augmentation was disabled to keep the frames aligned.

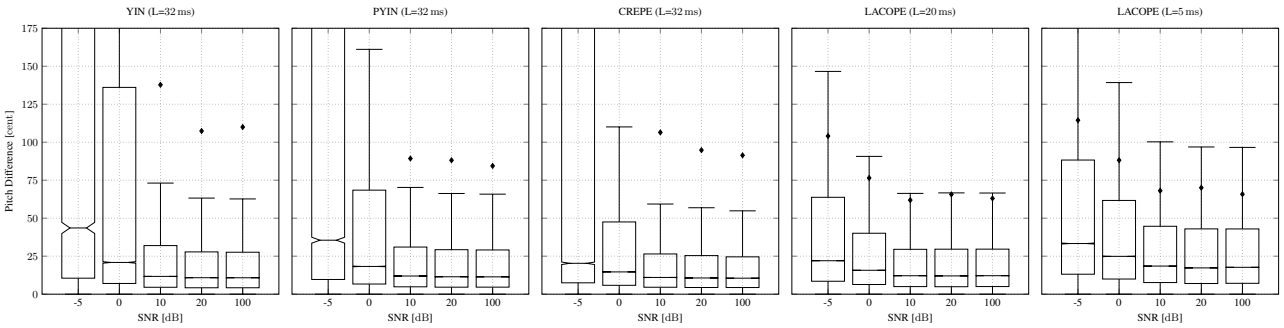

Figure 4: *Pitch difference based on PTDB dataset at different SNR conditions. Look-ahead L in brackets, diamonds represent arithmetic mean as a measure of the amount and strength of outliers. Note, that the official PYIN Vamp plugin implementation performed significantly worse for low SNRs. Instead, we compare to the better results of the librosa implementation [16].*

5. Experiments and Results

Our model was trained for 20 epochs with a batch size of 32, a learning rate of $5e{-}4$ using AdamW [25] optimizer with a weight decay of $1e{-}4$. Fig. 4 shows pitch difference on the PTDB test set. We compare our performance to YIN [9], PYIN [6] and CREPE [7] which all use much larger frame sizes and look-aheads. LACOPE is robust within all SNR conditions due to the speech enhancement multi-target training and the extensive noise augmentation. While the median difference is slightly worse compared to CREPE, the overall amount and strength of outliers is lower, as indicated by the IQR and mean.

As can be seen in Fig. 6, our model adapts well to speech, even if there is superimposed harmonic noise in the same frequency range. Note, that our network tends to classify frames after a voiced period still as voiced. This, however, is typically not an issue since the comb filter has no effect if gains are close to 0. CREPE and PYIN often adapt to the noise and classify most frames as voiced, while PYIN pitch estimates are often octave errors.

Fig. 5 shows the pitch accuracy measured in percentage of frames with a pitch difference below $50\,\text{cents}$. Here, the trade-off between look-ahead and pitch accuracy becomes clear. We achive comparable or better performance at lower look-aheads compared to related work.

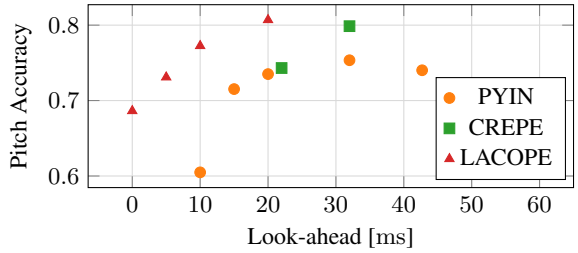

Figure 5: *Pitch Accuracy average over all SNRs. We compare to PYIN with different frame sizes, CREPE as well as 10 ms delayed CREPE, since the frame size is fixed. For PYIN and CREPE, the look-ahead corresponds to half of the frame size, hop size is not considered.*

We do not report metrics like recall and precision on the voiced estimation. While our model was trained on the target voiced probability estimation by PYIN, we argue that our model learns a slightly different representation. The last term in our loss does not result in the voiced probability estimation, but rather an estimate, to which degree the comb filter should be applied. Thus, the outputs are similar but not exactly the same.

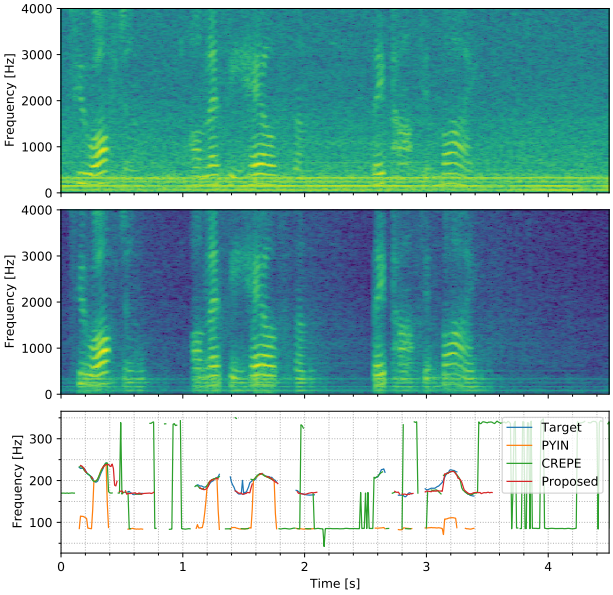

Figure 6: *Sample from the test set containing harmonic engine noise. Top: noisy spectrogram, middle: enhanced spectrogram, bottom: pitch estimates. Since other algorithms are sensitive to all kinds of periodic structures, they often adapt to the noise instead of the speech.*

6. Conclusion

In this paper, we presented LACOPE, a joint approach for pitch estimation and speech enhancement under low latency requirements. We showed, that we achieve on par performance compared to CREPE and better performance than PYIN especially in noisy conditions. While our model with approx. $2.4\,\text{M}$ parameters and $57\,\text{MFLOPs}$ per $10\,\text{ms}$ segment is still too large for running on embedded devices, our computational requirements including speech enhancement are a lot lower compared to CREPE. Here, we measured $28.2\,\text{BFLOPs}$ per $10\,\text{ms}$ segment which are about $2 \cdot 10^6$ more operations due to the amount of large convolutional layers.

To further reduce the computational requirements, we plan to integrate methods from [13], like hierarchical RNNs and Bark scaling for the input spectrogram and output gains. Furthermore, techniques like pruning and quantization will be used for an additional complexity reduction [26].

7. References

[1] J. Agnew and J. M. Thornton, "Just noticeable and objectionable group delays in digital hearing aids," *Journal of the American Academy of Audiology*, vol. 11, no. 6, pp. 330–336, 2000.

[2] J.-M. Valin, "A hybrid DSP/deep learning approach to real-time full-band speech enhancement," in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2018, pp. 1–5.

[3] J.-M. Valin, U. Isik, N. Phansalkar, R. Giri, K. Helwani, and A. Krishnaswamy, "A Perceptually-Motivated Approach for Low-Complexity, Real-Time Enhancement of Fullband Speech," in *INTERSPEECH 2020*, 2020.

[4] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the opus codec," *arXiv preprint arXiv:1602.04845*, 2016.

[5] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[6] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 659–663.

[7] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A Convolutional Representation for Pitch Estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.

[8] 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.

[9] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[10] X. Zhang, H. Zhang, S. Nie, G. Gao, and W. Liu, "A Pairwise Algorithm Using the Deep StackingNetwork for Speech Separation and Pitch Estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 6, pp. 1066–1078, 2016.

[11] E. Hänsler and G. Schmidt, *Acoustic echo and noise control: a practical approach*. John Wiley & Sons, 2005, vol. 40.

[12] M. Aubreville, K. Ehrensperger, A. Maier, T. Rosenkranz, B. Graf, and H. Puder, "Deep denoising for hearing aid applications," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 361–365.

[13] H. Schröter, T. Rosenkranz, A. N. Escalante-B., P. Zobel, and A. Maier, "Lightweight Online Noise Reduction on Embedded Devices using Hierarchical Recurrent Neural Networks," in *INTERSPEECH 2020*, 2020. [Online]. Available: https://arxiv.org/abs/2006.13067

[14] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 708–712.

[15] U. Isik, R. Giri, N. Phansalkar, J.-M. Valin, K. Helwani, and A. Krishnaswamy, "PoCoNet: Better Speech Enhancement with Frequency-Positional Embeddings, Semi-Supervised Conversational Data, and Biased Loss," in *INTERSPEECH 2020*, 2020.

[16] B. McFee, V. Lostanlen, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, J. Mason, D. Ellis, E. Battenberg, S. Seyfarth, R. Yamamoto, K. Choi, viktorandreevichmorozov, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, D. Hereñú, F.-R. Stöter, P. Friesch, A. Weiss, M. Vollrath, and T. Kim, "librosa/librosa: 0.8.0," Jul. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3955228

[17] D. Chan, A. Fourcin, D. Gibbon, B. Granström, M. Huckvale, G. Kokkinakis, K. Kvale, L. Lamel, B. Lindberg, A. Moreno *et al.*, "EUROM-A spoken language resource for the EU-The SAM projects," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[18] C. Veaux, J. Yamagishi, and K. Macdonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," 2017.

[19] K. Ito and L. Johnson, "The LJ Speech Dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[20] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings," in *Proceedings of Meetings on Acoustics ICA2013*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 035081.

[21] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[22] M. Jeub, M. Schäfer, and P. Vary, "A Binaural Room Impulse Response Database for the Evaluation of Dereverberation Algorithms," in *Proceedings of International Conference on Digital Signal Processing (DSP)*, IEEE, IET, EURASIP. Santorini, Greece: IEEE, Jul. 2009, pp. 1–4.

[23] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[24] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "The pitch-tracking database from grazuniversity of technology," 2012.

[25] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019.

[26] J.-M. Valin, S. Tenneti, K. Helwani, U. Isik, and A. Krishnaswamy, "Low-Complexity, Real-Time Joint Neural Echo Control and Speech Enhancement Based On PercepNet," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.