Interspeech 2022
18-22 September 2022, Incheon, Korea

# Robust Pitch Estimation Using Multi-Branch CNN-LSTM and 1-Norm LP Residual

*Mudit Batra*[1], *M.K. Jayesh*[2], *C.S. Ramalingam*[1]

[1]Department of Electrical Engineering, IIT Madras, Chennai, India
[2]Samsung Research India, Bangalore

muditbatra2211@gmail.com, jayeshmkoroth@gmail.com, csr@ee.iitm.ac.in

## Abstract

Pitch and voicing determination are important in many speech and audio signal processing applications. Even in the clean signal case their estimation can pose problems, and more so when noise is present. In this paper we propose a Multi-Branch CNN-LSTM based Temporal Neural Network for pitch and voicing determination. In addition, rather than using the raw waveform, we use the $\ell_1$-norm based LP residual as the input signal. These changes have made the proposed method more robust to SNR degradation, i.e., even though there is a slight fall in accuracy in the clean signal case, there is a $2.9\%$ absolute increase in RPA for the 0 dB case when compared with the CREPE algorithm. More importantly, when the RPA tolerance is tightened, the fall in accuracy is smaller. This robustness has been achieved with only 1.79M parameters, which is an order of magnitude less than what is used in CREPE.

**Index Terms**: Pitch estimation, voicing determination, deep learning.

## 1. Introduction

Pitch estimation and tracking, along with its closely associated problem of voicing determination, is an important problem in audio and speech analysis. This task is difficult even when the signal is clean and monophonic. The factors that contribute to the difficulty are, changes in the vocal tract, pitch variations over the analysis window and its wide range, irregularity in the excitation signal, etc. [1]. From the signal processing point of view, depending on the context, the term *pitch* could mean either the fundamental *frequency* ($f_0$) or the fundamental *period* ($T_0 = 1/f_0$) of the (quasi-)periodic analysis segment. If this segment is part of a speech waveform, the reference points are usually placed at the glottal closure instants (GCIs).

Pitch estimation finds use in several applications such as text-to-speech synthesis, speaker verification, speech analysis, etc. Even though this has been well-studied, the problem is far from solved. The works by Hess [1–3] are comprehensive references on this topic. Pitch is typically estimated either in the time-domain [4–6], the frequency domain [7–9], or the cepstral domain [10,11]. The feature vectors used in conventional methods include zero-crossing rate (ZCR), summation of residual harmonics (SRH) [12], summation of speech harmonics (SSH), cepstral peak prominence (CPP) [13], etc. YIN [4] estimates the autocorrelation of the raw waveform and then uses the local minima of the Normalized Mean Difference Function (NMDF). PYin [14] is a probabilistic modification of YIN [4] and uses Hidden Markov Models (HMMs). Other time-domain methods are PRAAT [15] and RAPT [6], which use the normalized cross-correlation function (NCCF). SWIPE [9] is a frequency-domain method in which template matching is performed with the spectrum of a sawtooth waveform to estimate pitch. PEFAC [8] is also a frequency-domain method that is robust in the presence of noise. Hybrid methods use both time- and frequency-domain features, an example of which is YAAPT [16].

Currently, machine-learning-based methods are very popular [17–21]. Unlike conventional methods, using the raw signal directly is more common in deep learning methods, e.g., CREPE [19]. This method employs a CNN-based end-to-end architecture and $f_0$ in log scale as the target. Its accuracy is good for both the clean and noisy cases. SPICE [20] uses unsupervised model training and a constant-Q transform (CQT) for pitch estimation. Deep-F0 [22] and PIMOD [23] are other Deep Learning based methods that directly operate on the raw waveform for estimating pitch. The high model complexity is a notable disadvantage; moreover, if a tighter threshold is specified on the RPA accuracy, the accuracy drops noticeably, especially for noisy speech. Therefore, despite the existence of several algorithms, developing a method that is both accurate as well as robust to noise continues to attract attention. The primary objective of this paper is to improve the accuracy for the noisy case, while keeping the model complexity low.

In this paper we propose to use the $\ell_1$-norm-based linear prediction (LP) residual as the input, unlike the raw waveform used in CREPE. From the architecture viewpoint, we propose to use a Multi-Branch CNN-LSTM based model, inspired from InceptionNet [24]. The $\ell_1$-norm is preferred because this criterion makes the filter estimate independent of the fundamental frequency of the speech signal. The hyper-parameters were tuned to get the best performance on the validation dataset. To generate noisy speech we have added white noise from the NOISEX-92 database [25]. We compare our results with CREPE [19] (Full and Tiny versions) and SWIPE [9].

## 2. Proposed Method

We begin by considering alternatives to the raw waveform for the input feature vector, followed by changes to the architecture.

### 2.1. Input Feature Vectors

In Deep Learning methods the features used have, not surprisingly, a direct bearing on the accuracy. Using the raw waveform as the input feature vector has been shown to give good results [19,22,23]. In this paper we study the following three additional feature representations (frame-size 40 ms at 16 kHz sampling rate, frame-shift of 10 ms): (a) **Magnitude Spectrum** (linear scale) obtained using 1024-point DFT and retaining only the first half; (b) **$\ell_2$-norm based LP Residual** obtained using filters computed using the $\ell_2$-norm minimization criterion (orders 14 and 16 were considered); and (c) **$\ell_1$-norm based LP Residual** obtained using filters computed using the $\ell_1$-norm minimization criterion (orders 14 and 16 were considered). When comput-



10.21437/Interspeech.2022-10704

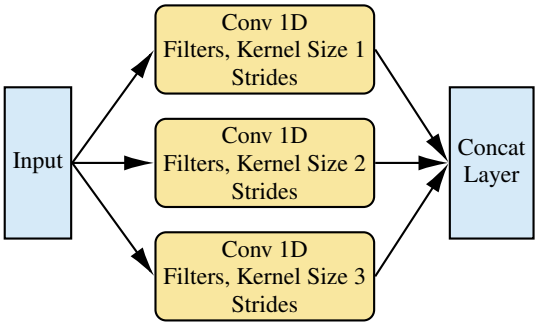


Figure 1: *Multi-Branch Conv 1D Block Diagram*

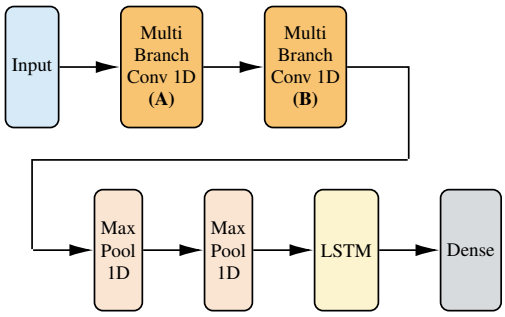


Figure 2: *Flowchart of the the proposed architecture.*

ing the LP residual, the beginning and ending transient portions were removed from the inverse filtered signal before further processing. All feature vectors were normalized in the range from $(-1, 1)$.

It has been observed that providing "context" significantly improves the performance of Deep Learning models [26]. *Context* is defined as appending $K$ past and $K$ future feature vectors, and treating this stack as the input to the neural network. We have chosen $K = 2$ to balance accuracy versus computational complexity.

### 2.2. Network Architecture

From the architecture point of view, increasing the number of layers leads to a better latent representation. The disadvantage is that such networks are not only prone to over-fitting but also suffer from vanishing or exploding gradients. Szegedy, et al. [24] proposed the idea of having multiple kernels with different sizes operating at the same level, which has the distinct advantage of extracting multilevel features while reducing the overall model complexity. Also, LSTMs, a form of recurrent neural networks, have been proven successful in modeling temporal data. Since the information in speech is quintessentially in the temporal domain, we propose to incorporate an LSTM layer in our architecture.

A block diagram of the proposed architecture is shown in Fig. 2. It consists of a stack of Multi-Branch convolutional blocks (Fig. 1), followed by an LSTM layer. The output from the LSTM is a 512-dimensional feature representation, which is passed on to a densely connected layer with Softmax activation that generates the output vector $\hat{\mathbf{y}}$. The dimension of $\hat{\mathbf{y}}$ is $300 \times 1$, with each entry corresponding to the pitch frequency in cents. *Cent* is a unit that represents frequency in logarithmic scale, with a specified reference frequency $f_{\text{ref}}$. We have chosen $f_{\text{ref}} = 10$. The range of pitch frequencies considered was 31.7–1014.1 Hz, which, in cents, corresponds to 1997.37–7996.87 cents. This range was divided into 300 uniform bins (20 cents bin width). A one-hot vector $\mathbf{y}$ is encoded with the bin corresponding to the ground-truth pitch value. The model was trained to minimize the categorical cross entropy loss between the predicted vector $\hat{\mathbf{y}}$ and the one-hot encoded ground truth vector $\mathbf{y}$; the Adam optimizer [27] was used in our experiments. For unvoiced frames first bin of target vector was set as one. The pitch estimate from $\hat{\mathbf{y}}$ was calculated by taking local weighted average over 9-bins.

We carried out a hyper-parameter search for the proposed network, which resulted in the following best configuration:

- Multi-Branch Conv 1D (A) - Filters: 64, Strides: 2, Kernel Size1: 5, Kernel Size2: 9, Kernel Size3: 15
- Multi-Branch Conv 1D (B) - Filters: 64, Strides: 2, Kernel Size1: 3, Kernel Size2: 5, Kernel Size3: 7
- Early stopping with tolerance value of 0.001 and patience value set to 20 epochs.
- Each Convolution Layer is followed by a Dropout Layer [28] with a dropout probability of 0.25
- ReLU activation for all layers, except the last Dense Layer where Softmax is used.
- Learning Rate $= 0.0005$ and Batch Size $= 64$.

## 3. Experimental Setup

### 3.1. Dataset

The publicly available Pitch Tracking Database (PTDB-TUG) [29] was used in our experiments. It consists of laryngograph signals, audio files, and pitch trajectories of 20 English speakers (10 Male and 10 Female). For generating noisy speech at specified SNRs, we used white noise from the Noisex-92 [25] database. The SNRs considered were $\infty$ (i.e., clean speech), 20, 10, 5, and 0 dB.

### 3.2. Methodology

The data were split into 60/20/20 (train/validate/test) to maintain fairness in the model's training and testing generalizability. For testing purposes, data corresponding to two male and two female speakers were kept aside (not part of the training set). The evaluation metrics used were the Raw Pitch Accuracy (RPA) and Voicing Recall Rate (VRR). RPA is defined as the percentage of the voiced frames for which pitch error is within 50 cents (or 0.5 semitones). VRR is defined as the percentage of voiced frames correctly estimated by the algorithm. Apart from robustness to noise, robustness to RPA threshold was also studied by varying it in the range 50–30 cents in steps of 10 cents.

As stated earlier, the model architecture and the input feature vector jointly influence the overall performance of any deep learning model. We examine the performance of feature vectors described in Sec. 2.1, while keeping the model architecture unchanged. Similarly, we use the $\ell_1$-norm based LP Residual with LP order 14 as the input to the network architecture and examine the importance of each building block of the model architecture.

We compare our proposed model with CREPE and SWIPE. Since the PTDB-TUG dataset was not part of the training data in [19], that model was retrained on this data set. Two variants

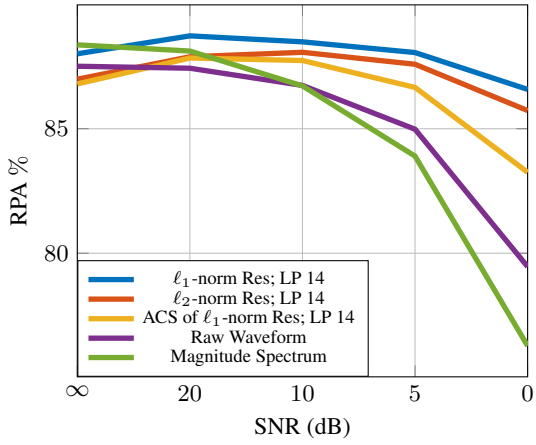


Figure 3: *The proposed method's RPA values for different feature vectors for a threshold of* 50 *cents. Clearly,* $\ell_1$*-norm residual with LP order* 14 *provides the most robustness against varying noise levels.*

of CREPE were compared, viz., *F(ull)* model having 22.5M parameters and *T(iny)* model having 487K parameters.

# 4. Results

We first present results of how changing the input feature affects RPA percentage. We have studied this from two angles, viz., how it varies with (i) SNR, and (ii) RPA threshold. For the best performing case, we have examined the effect of (i) removing different layers, and (ii) changing the activation function (Snake, Sinusoid, and ReLU).

## 4.1. Feature Vector Analysis

CREPE uses the raw waveform as the input feature vector. For speech signals, other natural candidates are the magnitude spectrum, the LP residual, and its autocorrelation sequence (ACS). Furthermore, the error norm criterion for estimating the LP inverse filter coefficients can be either the $\ell_1$-norm or the $\ell_2$-norm.

Fig. 3 shows how the proposed method's RPA varies as a function of SNR for the various input features. The plots show that the best overall results are obtained with the $\ell_1$-norm residual estimated using a 14-th order filter. A 16-th order filter was found to be slightly less effective than a 14-th order filter in terms of RPA and VRR accuracy. For clean speech, the magnitude spectrum input feature vector gave a slightly better accuracy compared to the $\ell_1$-norm residual (88.37% vs. 88.01%); however, except for this slight anomaly, the latter feature yielded the best overall performance (for an RPA threshold of 50 cents). Clearly, it is also more robust to SNR degradation: at 0 dB the RPA has fallen to only 86.58 as against 79.46 for the raw waveform. The $\ell_1$-norm criterion completely removes the influence of the vocal tract from the residual signal, which is a plausible explanation for its robustness. The behaviour of the $\ell_2$-norm residual is also similar; its relatively poorer performance can be attributed to its worsening ability to remove the vocal tract information with increasing $f_0$.

Henceforth we use the $\ell_1$-norm residual signal obtained with model order 14 as the input feature vector because that gave the best performance for this task among all the input feature vectors that were examined.

| Model | Params. | Metric | Clean | 20 dB | 10 dB | 5 dB | 0 dB |
|---|---|---|---|---|---|---|---|
| Swipe | | VRR | 87.92 | 85.65 | 64.89 | 48.52 | 34.01 |
| | | RPA | 85.33 | 83.01 | 62.29 | 46.81 | 32.05 |
| CREPE (T) | 487K | VRR | **92.04** | **91.99** | 91.51 | 90.37 | 86.79 |
| | | RPA | 88.10 | 88.06 | 87.59 | 86.48 | 83.04 |
| CREPE (F) | 22.5M | VRR | 91.52 | 91.47 | 91.02 | 89.85 | 86.46 |
| | | RPA | **88.55** | 88.52 | 88.08 | 86.95 | 83.69 |
| Proposed | 1.79M | VRR | 91.09 | 91.74 | **91.54** | **91.18** | **89.72** |
| | | RPA | 88.01 | **88.73** | **88.49** | **88.06** | **86.58** |

Table 1: *RPA and VRR comparison (in %) on PTDB-TUG data set with added white noise for different SNRs and RPA threshold of* 50 *cents.*

## 4.2. Model Robustness

We next compare our proposed method with SWIPE, CREPE (F) (22.5 M parameters), and CREPE (T) (487 k parameters). The results for white noise are given in Table 1. For the clean speech case, the highest RPA value was obtained using CREPE (F), whereas CREPE (T) gave the highest VRR accuracy. For all other SNR conditions the RPA estimates of the proposed method is better. As the SNR degrades, the proposed method becomes noticeably better than CREPE: e.g., it shows an absolute improvement of 3.26% in VRR and 2.89% in RPA for the 0 dB SNR case. This is particularly noteworthy since the proposed method has only 1.79 M parameters, which is 92 % less than what is used by CREPE (F).

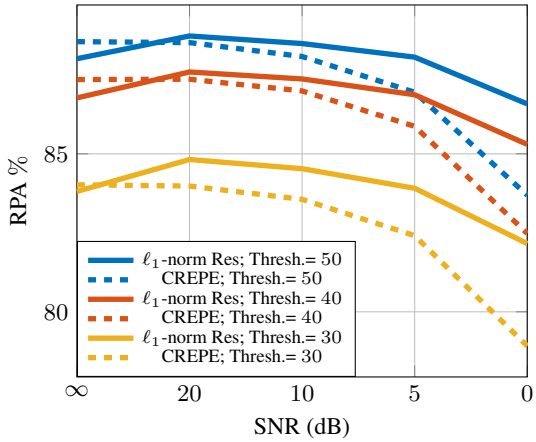


Figure 4: *Comparative performance of the proposed model versus CREPE as a function of RPA threshold. The proposed method outperforms CREPE when the RPA threshold is tightened.*

The robustness of the model can also be tested along another dimension, viz., the drop in RPA values as the threshold is tightened. To investigate this further, the default RPA threshold of 50 cents was decreased to 40 and 30 cents. We then compared the performance with CREPE (F). The results are shown in Fig.4. Clearly, the drop in accuracy of the proposed Model is much smaller, whereas CREPE (F) experiences a comparatively significant drop in performance. Although the performance of the proposed method is slightly poorer for clean speech, the drop in accuracy is smaller as the SNR degrades, with the maximum difference seen for the 0 dB case.

This robustness to tightening the RPA threshold is directly linked to the input feature. To see this, we compared the $\ell_1$-norm LP residual and raw waveform in the proposed method. Fig. 5 shows the results. Once again, it is clear that $\ell_1$-norm LP

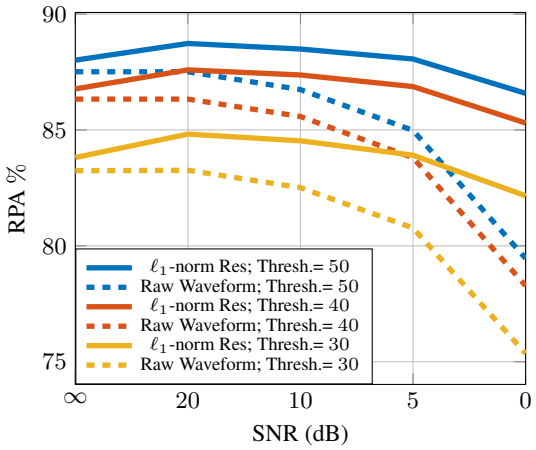


Figure 5: *Effect of tightening the RPA threshold on accuracy when using the $\ell_1$-norm residual versus the raw waveform in the proposed method. Clearly, the $\ell_1$-norm residual signal provides more robustness than the raw waveform.*

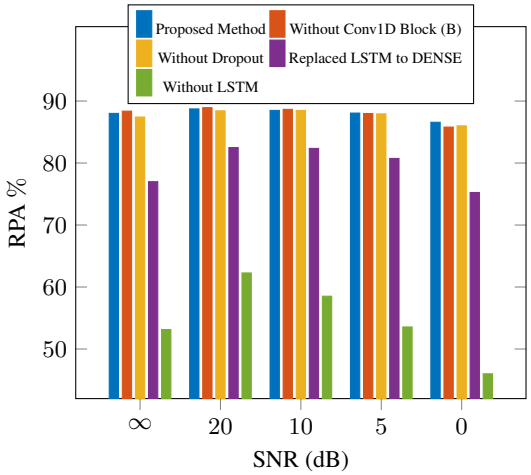


Figure 6: *Model ablation study with the input feature vector set as $\ell_1$-norm LP Residual; LP order* 14 *and RPA threshold set as* 50 *Cents.*

residual feature does not lead to as much a drop when the RPA tolerance is tightened. Note that the network architectures of the proposed method and that of CREPE are quite different. Nevertheless, the performance drop with decreasing SNRs are very similar in Figs. 4 and 5, which lends credence to the claim that the robustness is due to the $\ell_1$-norm LP residual input feature.

We used other metrics such as VUE, UVE, and GPE [21]. When averaged over all SNRs, the results of our method versus CREPE(F) were: UVE of $1.67\%$ vs $1.44\%$, VUE of $8.94\%$ vs $10.48\%$, and GPE of $1.79\%$ vs $1.88\%$. We also tested the RPA performance by adding babble noise from NoiseX-92 database [25]. CREPE(F) performed slightly better for all SNR levels for RPA thresholds of 50, 40 and 30 cents; however, for 20 and 10 cents, our method showed an average improvement of $2.6\%$ and $3.35\%$.

### 4.3. Effect of Various Network Layers

As a final point of comparison, we examine the effect of model architecture. We retained the input feature as the $\ell_1$-norm LP residual and carry out a block-wise analysis. The RPA threshold was kept at 50 cents throughout. The Multi-Branch Conv 1D blocks help in adding robustness to the model, especially at low SNR and tight RPA thresholds. For example, if only Block (B) is removed, for 0 dB, the RPA values dropped as follows: a drop of $0.78\,\%$ was observed for 50 cents , $0.76\,\%$ for 40 cents, $1.1\,\%$ for 30 cents, $2.31\,\%$ for 20 cents, and $4.59\,\%$ for 10 cents.

When the Dropout Layer was eliminated, the RPA results remained largely unchanged, but VRR decreased by roughly $2\%$ for all SNR levels. The decline in VRR, is conjectured to be directly related to the loss of generalisation capacity [28].

One of the most important layers in the model is LSTM. Two variants were tried, viz., replacing the LSTM layer by the Dense Layer and removing the LSTM layer completely. We can see from Fig. 6 that there is a significant drop in RPA in both cases. Moreover, the steepest drop happened when the LSTM layer was removed. This reveals the importance of this layer for pitch estimation. This is not surprising because the LSTM layer is very helpful in capturing the temporal dependencies of the input data. Finally, we carried out experiments by modifying the activation function. We have used ReLU in all the CNN layers of Multi-Branch Conv 1D block. Liu *et al.* [30] proposed a new activation function $x + \sin^2 x$ (called "snake"), which was claimed to be better than ReLU for periodic data. We have compared the proposed model's performance using the "snake" activation function as well as the sinusoidal activation function $\sin x$. The results are given in Table 2. ReLU gave the best results when compared to the other two choices. It should be noted that results from the other two activation functions are still better than CREPE.

| Activation | Metrics | Clean | 20dB | 10dB | 5dB | 0dB |
|---|---|---|---|---|---|---|
| Snake | VRR | 89.98 | 90.61 | 90.60 | 90.13 | 88.47 |
| | RPA | 86.35 | 86.97 | 86.93 | 86.40 | 84.67 |
| Sinusoid | VRR | 89.99 | 90.79 | 90.44 | 89.76 | 87.51 |
| | RPA | 87.04 | 87.84 | 87.51 | 86.87 | 84.68 |
| ReLU | VRR | **91.09** | **91.74** | **91.54** | **91.18** | **89.72** |
| | RPA | **88.01** | **88.73** | **88.49** | **88.06** | **86.58** |

Table 2: *Activation function comparison for an RPA threshold of* 50 *cents*

## 5. Conclusion

In this paper we proposed a robust method for pitch and voicing determination based on a Multi-Branch CNN-LSTM network. Our experiments showed that using the $\ell_1$-norm LP residual as the input feature gave us the best results, making the method robust to SNR variations and tightening of RPA thresholds. We also presented results using the raw waveform, the magnitude spectrum (linear scale), and the LP residual based on the $\ell_2$-norm. Adding the LSTM layer helped in improving the performance because of its ability to model temporal variations. When white noise was added, our model outperformed CREPE and SWIPE, especially at lower SNRs, leading to its robustness. For the 0 dB SNR case we obtained an absolute improvement of $2.9\%$ in RPA with 50 cents as threshold. It also performed well when the RPA threshold was tightened to 30 cents and resulted in an absolute improvement of $3.24\%$ over CREPE. The model had only $1.79$ M parameters, compared to CREPE's $22.52$ M.

# 6. References


[1] W. Hess, "Pitch and voicing determination of speech with an extension toward music signals," in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. Berlin: Springer, 2008, pp. 181–227.

[2] ——, *Pitch Determination of Speech Signals*. New York, NY: Springer-Verlag, 1983.

[3] ——, "Pitch and voicing determination," in *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi, Eds. New York: Marcel Dekker Inc., 1992, pp. 3–48.

[4] 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.

[5] J. Markel, "The sift algorithm for fundamental frequency estimation," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 5, pp. 367–377, 1972.

[6] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding and Synthesis*, pp. 495–518, 1995.

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

[8] S. Gonzalez and M. Brookes, "PEFAC-a pitch estimation algorithm robust to high levels of noise," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 518–530, 2014.

[9] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[10] S. Ahmadi and A. S. Spanias, "Cepstrum-based pitch detection using a new statistical v/uv classification algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 333–338, 1999.

[11] T. Muhaseena and M. Lekshmi, "A model for pitch estimation using wavelet packet transform based cepstrum method," *Procedia Technology*, vol. 24, pp. 1061–1067, 2016.

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

[13] J. Hillenbrand, R. A. Cleveland, and R. L. Erickson, "Acoustic correlates of breathy vocal quality," *Journal of Speech, Language, and Hearing Research*, vol. 37, no. 4, pp. 769–778, 1994.

[14] 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.

[15] P. Boersma *et al.*, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings of the Institute of Phonetic Sciences*, vol. 17, no. 1193. Citeseer, 1993, pp. 97–110.

[16] K. Kasi and S. A. Zahorian, "Yet another algorithm for pitch tracking," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. I–361.

[17] T. Drugman, G. Huybrechts, V. Klimkov, and A. Moinet, "Traditional machine learning for pitch detection," *IEEE Signal Processing Letters*, vol. 25, no. 11, pp. 1745–1749, 2018.

[18] J. Zhang, J. Tang, and L.-R. Dai, "RNN-BLSTM based multi-pitch estimation," in *INTERSPEECH*, 2016.

[19] 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.

[20] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.

[21] D. N. Tran, U. Batricevic, and K. Koishida, "Robust Pitch Regression with Voiced/Unvoiced Classification in Nonstationary Noise Environments," in *Proc. Interspeech 2020*, 2020, pp. 175–179.

[22] S. Singh, R. Wang, and Y. Qiu, "DEEPF0: end-to-end fundamental frequency estimation for music and speech signals," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 61–65.

[23] Y. Segal, M. Arama-Chayoth, and J. Keshet, "Pitch estimation by multiple octave decoders," *IEEE Signal Processing Letters*, vol. 28, pp. 1610–1614, 2021.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[25] 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," *Speech Communication*, vol. 12, no. 3, pp. 247–251, jul 1993. [Online]. Available: https://doi.org/10.1016%2F0167-6393%2893%2990095-3

[26] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *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.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 3rd International Conference for Learning Representations (ICLR'15)*, 2015.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[30] L. Ziyin, T. Hartwig, and M. Ueda, "Neural networks fail to learn periodic functions and how to fix it," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1583–1594, 2020.