INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Robust pitch regression with voiced/unvoiced classification in nonstationary noise environments

Dung N. Tran, Uros Batricevic, Kazuhito Koishida

Microsoft Corporation
{dung.tran, urosb, kazukoi}@microsoft.com

Abstract

Accurate voiced/unvoiced information is crucial in estimating the pitch of a target speech signal in severe nonstationary noise environments. Nevertheless, state-of-the-art pitch estimators based on deep neural networks (DNN) lack a dedicated mechanism for robustly detecting voiced and unvoiced segments in the target speech in noisy conditions. In this work, we proposed an end-to-end deep learning-based pitch estimation framework which jointly detects voiced/unvoiced segments and predicts pitch values for the voiced regions of the ground-truth speech. We empirically showed that our proposed framework significantly more robust than state-of-the-art DNN based pitch detectors in nonstationary noise settings. Our results suggest that joint training of voiced/unvoiced detection and voiced pitch prediction can significantly improve pitch estimation performance.

Index Terms: pitch estimation, fundamental frequency, deep neural network, voiced/unvoiced classification, pitch regression

1. Introduction

Pitch estimation refers to estimating the fundamental frequency (F_0) of a quasi-periodic signal, typically a digital recording of speech or a musical tone. It is a fundamental problem in speech processing and serves as a crucial component in various speech applications such as text-to-speech, speech enhancement, speech recognition, and speaker identification.

Traditional pitch estimation methods typically comprise a simple signal processing algorithm or heuristic followed by a pitch smoothing step [1, 2, 3, 4, 5, 6]. Many reliable signal processing methods have been proposed in the past several decades which commonly exploit either harmonic structures in the frequency domain or periodic structures in the time domain of the input signal. pYIN [7] uses a probabilistic model to predict the pitch sequence from the cumulative mean normalized difference function of the time-domain input signal. SWIPE [8] utilizes a template matching heuristic operating in the spectrum domain.

Recent advances in deep learning and traditional machine learning have led to several data-driven pitch estimation frameworks [9, 10, 11, 12, 13, 14, 15]. State-of-the-art data-driven pitch detectors commonly train an end-to-end deep neural network (DNN) that predicts a pitch sequence from the corresponding time-domain audio excerpt. In [13], Kim et. al. introduced a deep convolutional neural network (CNN), namely CREPE, that formulates pitch estimation as a classification problem. It consists of a series of 1D convolutional layers followed by a fully-connected layer and is trained to match an audio segment to the corresponding pitch values represented by smoothed one-hot vectors on the cent scale. During inference time, each predicted cent vector is collapsed into a single cent value which is then converted into the predicted fundamental frequency. A similar but more computationally economical CNN architecture was proposed in [15] targeting low-latency applications.

Despite outperforming signal processing baselines, state-of-the-art DNN based pitch estimators, which naively predict a pitch value for every sample of the audio input, fail to demonstrate robust results in more challenging scenarios. In applications such as speech enhancement in which the target speech is contaminated by nonstationary noise, accurately estimating the pitch information of the target speaker becomes nontrivial, especially when the noise power is comparable or overwhelms the target speech power. The reason for the poor performance of these pitch estimators in this situation is twofold. On one hand, the noise information in the unvoiced segments of the target speech inevitably causes these pitch detectors to make false F_0 prediction in these regions. On another hand, during training, attempting to fit the network in the unvoiced regions distracts it from fitting the voiced pitch values, leading to poor prediction accuracy in the voiced regions of the target speech.

The insights above suggest that robustly detecting voiced/unvoiced regions of the target speech, which is currently lacking from advanced DNN based pitch detectors such as CREPE, is critical to handle speech data overwhelmed by nonstationary noise. Based on this intuition, we propose to train a CNN which jointly detects voiced/unvoiced segments and predicts pitch values for the voiced regions of the ground-truth speech. Specifically, we introduce a CNN architecture that predicts F_0 from $2D$ features of the input signal in the time-frequency domain. The network is trained end-to-end in a supervised manner to minimize a hybrid loss comprising a classification loss for voiced/unvoiced detection and a regression loss for voiced pitch prediction. Using a training and test sets from a public speech enhancement database, we empirically show that our framework produces significantly more robust pitch estimations than state-of-the-art DNN based pitch estimators.

The paper is organized as follows. Section 2 introduces our F_0 prediction framework, the proposed network architecture, and the training loss. In Section 3, we first describe the dataset and the pitch estimation metrics used in the experiments. Then, we evaluate and benchmark our proposed framework againts state-of-the-art DNN based pitch detectors. Finally, Section 4 summarizes our contributions and discuss future work.

Notation. We use bold letters to express vectors or matrices. For a vector $\boldsymbol{v}$, its l-th element is denoted by v_l or $[v]_l$. We use $(\cdot)^T$ to indicate the vector/matrix transpose operator and $\odot$ the element-wise multiplication operator. For a set $\mathcal{A}$, we denote $|\mathcal{A}|$ its cardinality representing its number of elements.

2. Method

Consider a target speech segment with unknown fundamental frequency $\boldsymbol{f}$. Assume that the target speech is contaminated by additive noise yielding a noisy speech input excerpt $\boldsymbol{x}$. We seek

http://dx.doi.org/10.21437/Interspeech.2020-3019

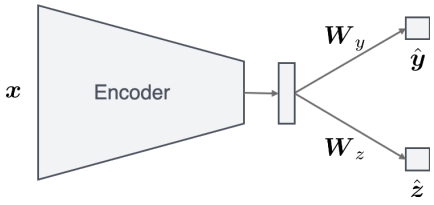

Figure 1: *Network diagram.*

Table 1: *Encoder architecture.*

Type	Kernel	Stride	Output	Activation
Input			$16 \times 256 \times 1$	
Conv2d	5×3	1×1	$16 \times 256 \times 64$	LReLU
Conv2d	3×3	1×2	$16 \times 128 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 64 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 32 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 16 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 8 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 4 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 2 \times 128$	LReLU
Conv2d	3×3	1×2	$16 \times 1 \times 128$	LReLU
Reshape			16×128	

an accurate estimate $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$ given $\boldsymbol{x}$. In our framework, $\boldsymbol{x}$ is a $L \times F$ log magnitude spectrum matrix corresponding to F frequency bins and L time frames. $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are L-long vectors representing the target and estimated fundamental frequencies of the input frames.

To obtain high-quality estimates of the ground-truth F_0, we build a DNN comprising a convolutional encoder followed by two fully-connected layers namely $\boldsymbol{W}_y$ and $\boldsymbol{W}_z$. The encoder consists of nine $2D$ convolutional layers gradually downsampling the input matrix along the frequency axis. This produces an encoding of size $L \times 128$. The first fully-connected layer $\boldsymbol{W}_y$ then transforms this encoding into a vector $\hat{\boldsymbol{y}} \in \mathbb{R}^L$, each element of which is a real number between 0 and 1 predicting the voiced/unvoiced probability of the corresponding frame in the input. Here, a voiced frame expects a large probability prediction, whereas a low probability prediction implies an unvoiced frame. The other fully-connected layer $\boldsymbol{W}_z$ decodes the encoding into a vector $\hat{\boldsymbol{z}} \in \mathbb{R}^L$ representing F_0 predictions corresponding to all input frames. These predictions are independent of the voiced/unvoiced decisions given by $\hat{\boldsymbol{y}}$. A high level diagram and detailed configuration of the proposed network is given in Fig. 1 and Table 1. The final fundamental frequency estimate $\hat{\boldsymbol{f}}$ is obtained by zeroing out the predicted F_0 values in the unvoiced frames detected by $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{z}} \odot \mathbb{1}_{\{\hat{\boldsymbol{y}} > \tau\}}. \tag{1}$$

In Eq. (1), τ is a threshold value between 0 and 1 deciding the voiced/unvoiced boundary and $\mathbb{1}_{\{\hat{\boldsymbol{y}} > \tau\}}$ is the indicator function of the set $\{\hat{\boldsymbol{y}} > \tau\}$ whose l-th element is defined by

$$[\mathbb{1}_{\{\hat{\boldsymbol{y}} > \tau\}}]_l = \begin{cases} 1 & \text{if } \hat{y}_l \geq \tau, \\ 0 & \text{if } \hat{y}_l < \tau. \end{cases} \tag{2}$$

We train the network using triplets $\left\{\left(\boldsymbol{x}^{(n)}, \boldsymbol{f}^{(n)}, \boldsymbol{y}^{(n)}\right)\right\}_{n=1}^{N}$ of log magnitude spectrum, the groundtruth fundamental frequencies, and the corresponding voiced/unvoiced labels, respectively. Here, N is the number of training samples. The voiced/unvoiced labels $\boldsymbol{y}^{(n)}, n = 1, ..., N$, are inferred from the fundamental frequencies:

$$y_l^{(n)} = \begin{cases} 1 & \text{if } f_l^{(n)} > 0, \\ 0 & \text{if } f_l^{(n)} = 0, \end{cases} \tag{3}$$

for $l = 1, ..., L$. In other words, $y_l^{(n)} = 1$ if frame l is voiced and $y_l^{(n)} = 0$ if the corresponding frame is unvoiced.

Our training loss consists of a regression loss for voiced pitch prediction and a classification loss for voiced/unvoiced detection:

$$\mathcal{L} = \mathcal{L}_{\text{pitch}} + \lambda \mathcal{L}_{\text{vu}}, \tag{4}$$

where $\lambda > 0$ is a scalar balancing the loss terms. In Eq. (4), the regression loss $\mathcal{L}_{\text{pitch}}$ forces the pitch estimate $\hat{\boldsymbol{z}}$ to be consistent with the target fundamental frequency $\boldsymbol{f}$ in the voiced regions and is defined as the average mean squared error over the training set:

$$\mathcal{L}_{\text{pitch}} = \frac{1}{N} \sum_{n}^{N} \left\| \boldsymbol{f}^{(n)} - \hat{\boldsymbol{z}}^{(n)} \odot \mathbb{1}_{\{\boldsymbol{y}^{(n)}=1\}} \right\|_2^2. \tag{5}$$

The classification loss $\mathcal{L}_{\text{vu}}$ encourages the voiced/unvoiced detection $\hat{\boldsymbol{y}}$ to be an accurate estimate of the voiced label $\boldsymbol{y}$. We use the binary cross-entropy function for this loss:

$$\mathcal{L}_{\text{vu}} = \frac{1}{N} \sum_{n}^{N} \left(-{\boldsymbol{y}^{(n)}}^T \log \hat{\boldsymbol{y}}^{(n)} - (\mathbf{1} - \boldsymbol{y}^{(n)})^T \log(\mathbf{1} - \hat{\boldsymbol{y}}^{(n)}) \right), \tag{6}$$

where $\mathbf{1}$ is the all-one vector of length L.

Our framework has an advantage over other state-of-the-art DNN pitch detectors such as CREPE that our F_0 regressor is assisted by a dedicated voiced/unvoiced classifier producing the final F_0 prediction, whereas CREPE uses a single output to predict both the pitch value and the prediction confidence. In other words, our network output is more expressive than that of CREPE.

3. Experimental results

In this section, we empirically evaluate and benchmark our pitch detection framework on a public noisy speech dataset.

3.1. Dataset

We use the widely-used VCTK dataset by Valentini et al [16] which is publicly available at [17]. The dataset includes clean and noisy speech data sampled at 48 kHz. In our experiments, we downsample the data to 16 kHz.

For training, we use the clean speech audio data of 28 speakers selected from the Voice Bank corpus [18]. The noisy training data are created by adding to the clean speech ten different types of noise at four signal-to-noise ratios (SNR), yielding 40 noise conditions. The ten noise types include eight real noise samples selected from the Demand data [19] and two artificially ones. The four training SNRs are 0 dB, 5 dB, 10 dB, and 15 dB.

The test data is different from the training data. The noisy test set is created by adding five different types of noises from the Demand database to the clean speech of two speakers from the Voice Bank corpus. The noise types and speakers in the test set are different from the ones used in training. The four test SNR values are 2.5 dB, 7.5 dB, 12.5 dB, and 17.5 dB. Consequensely, there are 20 different noise conditions.

3.2. Pitch detection metrics

We use four commonly used pitch estimation error metrics [20] to comprehensively evaluate the performance of our proposed pitch predictor. They intuitively quantify the effect of the individual loss terms in our loss function: detecting voiced/unvoiced frames and fitting F_0 in voiced segments. Below, we let M_{voiced} denote the number of voiced frames in the test set and M_{unvoiced} the number of unvoiced frames.

The first two metrics concern our voiced/unvoiced detection performance.

- *Unvoiced-to-Voiced Error (*UVE*).* This error counts the incorrect unvoiced frames detection normalized by the total number of unvoiced frames in the test set:

$$\text{UVE} = \frac{|\{y_m = 0 \text{ and } \hat{y}_m \geq \tau, m = 1, ..., M_{\text{unvoiced}}\}|}{M_{\text{unvoiced}}}. \tag{7}$$

- *Voiced-to-Unvoiced Error (*VUE*).* This metric quantifies the error rate in incorrectly detecting the voiced frames in the test set:

$$\text{VUE} = \frac{|\{y_m = 1 \text{ and } \hat{y}_m < \tau, m = 1, ..., M_{\text{voiced}}\}|}{M_{\text{voiced}}}. \tag{8}$$

When the network correctly detects the voiced frames in the test set, we use the next two measures to evaluate its ability of fitting pitch in these frames. They assess the F_0 prediction by how far it deviates from the ground-truth in the time domain. Below, both the ground-truth and predicted F_0 are nonzero as the corresponding voiced frame is predicted to be voiced.

- *Gross Pitch Error (*GPE*).* This measure indicates the rate at which the pitch predictor produces unsatisfactory prediction: the predicted pitch period is outside an acceptable range from the actual pitch period.

$$\text{GPE} = \frac{\left|\left\{\left|\frac{1}{\hat{f}_m} - \frac{1}{f_m}\right| > \frac{10}{f_s}, m = 1, ..., M_{\text{voiced}}\right\}\right|}{M_{\text{voiced}}}. \tag{9}$$

  Here, f_s is the signal sampling rate [20]. A small GPE implies a superior F_0 fitting performance and vice versa.

- *Fine Pitch Error (*FPE*)* This metric specifies the rate at which the F_0 estimator yields acceptable prediction.

$$\text{FPE} = \frac{\left|\left\{\left|\frac{1}{\hat{f}_m} - \frac{1}{f_m}\right| \leq \frac{10}{f_s}, m = 1, ..., M_{\text{voiced}}\right\}\right|}{M_{\text{voiced}}}. \tag{10}$$

  It is desirable to have a large FPE as it indicates that the estimator frequently generates satisfactory prediction.

It is important to note that VUE+GPE+FPE $= 1$ and these three metrics depend on the voiced/unvoiced decision threshold τ. Therefore, there is a trade-off when choosing the best τ value for the predictor using these metrics.

3.3. Experimental setup

In this subsection, we describe the data format that our network expects and the training setup.

Data format. We use the log magnitude spectrum of the noisy speech as the input to the network. To obtain the log magnitude spectrum, we apply the Short-Time-Fourier-Transform (STFT) to the raw audio using 512 FFT points with a hop size of 128 and Hann window. This results in 16×257 overlapping time-frequency frames. Finally, we remove the last frequency bin yielding 16×256 time-frequency frames, where 16 frames correspond to 128 ms.

We use the popular World vocoder [21] to extract the ground-truth fundamental frequencies from the clean speech audio. We use the same FFT and hop sizes as above to be consistent with the STFT frames.

Training setup. We train our network using the ADAM optimizer with a learning rate of 0.0001, decay rates $\beta_1 = 0.5$ and $\beta_2 = 0.9$, and a batch size of 64. The leaky ReLU constant is 0.2. To balance the scales of the loss terms in the loss function, we set $\lambda = 220$. To prevent overfitting, we apply ℓ_2 regularization to the convolutional weights with a value of 0.1. The network is trained for 100 epochs.

3.4. Results

We evaluate and benchmark our proposed framework against two state-of-the-art DNN based pitch estimators: CREPE [13] and FCNF0 [15]. We retrain both CREPE and CFNF0 on the same training data we use to train our network and test all the methods on the same test dataset described in Subsection 3.1. We evaluate the methods in two aspects: detecting voiced/unvoiced frames and predicting F_0 in voiced frames.

Recall that CREPE and FCNF0 predict a single smoothed one-hot vector in the cent domain, e.g., a Gaussian curve over the cent scale. In a perfect condition, the peak value of this vector serves two purposes: its index in the vector indicates the predicted pitch value and its actual value represents the confidence of the prediction. If the confidence is low, the corresponding frame is considered unvoiced, and the predicted F_0 is zero. If the confidence is high, the predicted pitch value is retained. In other words, a single prediction vector is used for both voiced/unvoiced prediction and F_0 fitting. We argue that this approach is less expressive than our hybrid framework in which voiced/unvoiced classification and pitch regression are individually performed by dedicated parts of the network, and the final prediction is the combination of the outputs from both tasks. The next subsections validate our argument.

3.4.1. Voiced/unvoiced detection

To evaluate the voiced/unvoiced detection performance of the approaches, we compare the UVE and VUE of the methods. Intuitively, it is desirable to have small UVE and VUE. Fig. 2 draws the receiver operating characteristic (ROC) curves for the benchmarked frameworks. Each point on the ROC curve for our proposed approach corresponds to the UVE and $(1 - \text{VUE})$ values given a value of the threshold τ defined in Eq. (1). Similarly, each point on the ROC curves for CREPE and FCNF0 represents the UVE and $(1 - \text{VUE})$ decided by the predicted confidence value given a threshold between 0 and 1.

Fig. 2 shows that our ROC curve is closest to the top left of the graph among the methods implying we perform best in classifying voiced and unvoiced frames. In other words, our voiced/unvoiced detector achieves the highest discrimination power on the test data compared to CREPE and FNCF0.

3.4.2. Voiced pitch estimation

Here, we assess the pitch prediction performance of the frameworks on the correctly classified voiced frames. As the portion of voiced frames correctly predicted varies according to the voiced/unvoiced decision boundary, we compare the voiced F_0 estimation performance of the methods across all

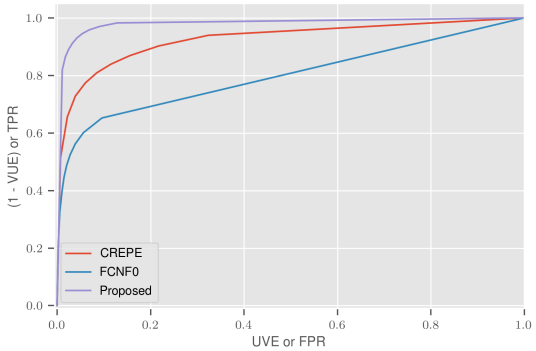

Figure 2: *ROC curve of the benchmarked methods for detecting voiced/unvoiced frames.*

voiced/unvoiced decision threshold values.

Fig. 3 shows the FPE values when τ varies. The plot suggests our network archives the biggest FPE values for all τ, with the biggest gap of 25% to the second-best estimator (CREPE) at $\tau = 0.9$. The result implies that our estimator consistently maintains the highest rate of satisfactory F_0 predictions in the voiced frames.

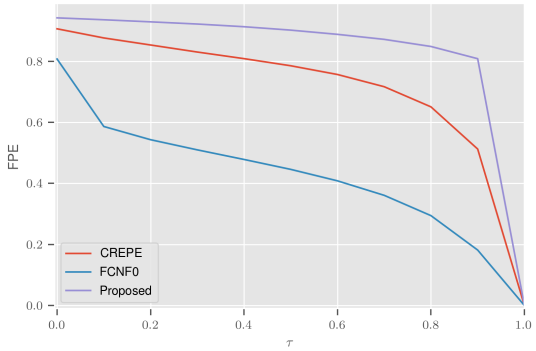

Figure 3: *FPE values of the pitch detectors when the voiced/unvoiced decision threshold varies. A larger FPE corresponds to better voiced-pitch estimations.*

The GPE values are reported in Fig. 4. The plot indicates that the proposed predictor, alongside CREPE, has the smallest values of GPE when τ changes. The result implies that our estimator produces the lowest rate of unacceptable pitch predictions and is comparable with CREPE.

Combining Fig. 3 and Fig. 4, we conclude that our proposed predictor consistently archives the best performance in predicting the pitch values in the voiced frames.

3.4.3. Overall performance

In this subsection, we comprehensively compare the detectors using all of the metrics. We also present three FPE values in the frequency domain: $\text{FPE}_{0.05}$, $\text{FPE}_{0.1}$, and $\text{FPE}_{0.2}$. Here,

$$\text{FPE}_{0.05} = \frac{\left|\left\{\frac{|\hat{f}_m - f_m|}{f_m} \leq 0.05, m = 1, ..., M_{\text{voiced}}\right\}\right|}{M_{\text{voiced}}}. \quad (11)$$

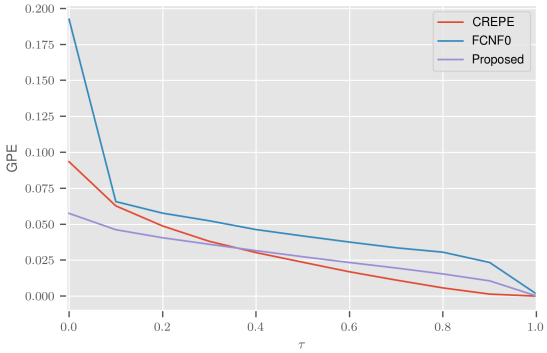

Figure 4: *GPE values of the pitch detectors when the voiced/unvoiced decision threshold varies. A smaller GPE implies better voiced-pitch estimations.*

$\text{FPE}_{0.1}$ and $\text{FPE}_{0.2}$ are defined similarly.

As mentioned in Subsection 3.2, choosing a voiced/unvoiced decision threshold for each estimator requires a trade-off between the metrics and depends on the application. For a fair comparison, we scientifically choose the threshold such that the UVE value for each detector is approximately 0.1. Table 2 shows the performance of the predictors given such voiced/unvoiced decision threshold values. The results indicate that our proposed framework significantly outperforms the other methods in detecting voiced samples and predicting satisfactory voiced pitch, and produces a slightly higher rate of unsatisfactory voiced pitch detection compared with CREPE.

Table 2: *Comparison of the pitch detectors when* UVE *is approximately* 0.1. *For all metrics excepts* GPE*, a larger value indicates a better performance.*

Metrics	CREPE	FCNF0	Proposed
UVE	0.115	0.096	**0.091**
VUE	0.161	0.348	**0.030**
FPE	0.809	0.586	**0.929**
GPE	**0.030**	0.066	0.041
$\text{FPE}_{0.05}$	0.761	0.565	**0.890**
$\text{FPE}_{0.1}$	0.817	0.588	**0.942**
$\text{FPE}_{0.2}$	0.833	0.605	**0.960**

4. Conclusions

We proposed a CNN based pitch detector which jointly performs voiced/unvoiced classification and voiced pitch regression. We precisely quantified its performance using popular metrics for voiced/unvoiced detection and pitch estimation. We empirically showed that our framework is significantly more robust than state-of-the-art DNN based pitch detectors on all metrics when handling severe nonstationary noise. Our results suggest that fitting voiced-pitch frames coupled with robustly detecting voiced/unvoiced regions of the target speech can improve pitch estimation performance significantly.

5. References

[1] A. M. Noll, "Cepstrum pitch determination," *The journal of the acoustical society of America*, vol. 41, no. 2, pp. 293–309, 1967.

[2] J. Dubnowski, R. Schafer, and L. Rabiner, "Real-time digital hardware pitch detector," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 2–8, 1976.

[3] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 5, pp. 353–362, 1974.

[4] D. Talkin, "A robust algorithm for pitch tracking (rapt)," *Speech Coding and Synthesis*, 1995.

[5] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," *Proceedings of Institute of Phonetic Sciences*, vol. 17, pp. 97–110, 1993.

[6] A. D. Cheveigne 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.

[7] M. Mauch and S. Dixon, "Pyin: A fundamental frequency estimator using probabilistic threshold distributions," *ICASSP*, pp. 659–663, 2014.

[8] 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, 2014.

[9] P. Verma and R. W. Schafer, "Frequency estimation from waveforms using multi-layered neural networks," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 8, pp. 2165–2169, 2016.

[10] K. Han and D. L. 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.

[11] R. M. Bittner, B. Mcfee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music," *Ismir*, pp. 23–27, 2017.

[12] A. Kato and T. Kinnunen, "Waveform to single sinusoid regression to estimate the f0 contour from noisy speech using recurrent deep neural networks," *Arxiv 1807.00752*, 2018.

[13] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," *Arxiv 1802.06182*, 2018.

[14] G. Doras, P. Esling, and G. Peeters, "On the use of u-net for dominant melody estimation in polyphonic music," *2019 IEEE International Workshop on Multilayer Music Representation and Processing (MMRP)*, pp. 66–70, 2019.

[15] L. Ardaillon and A. Roebel, "Fully-convolutional network for pitch estimation of speech signals," *INTERSPEECH*, 2019.

[16] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks," *Interspeech*, 2016.

[17] "https://datashare.is.ed.ac.uk/handle/10283/2791."

[18] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," *Proc. Int. Conf. Oriental COCOSDA*, 2013.

[19] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings," *J. Acoust. Soc. Am.*, vol. 133, no. 5, pp. 3591–3591, 2013.

[20] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 5, 1976.

[21] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.