INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Frequency Estimation from Waveforms using Multi-Layered Neural Networks

Prateek Verma & Ronald W. Schafer

Stanford University

prateekv@stanford.edu, rschafer@stanford.edu

Abstract

For frequency estimation in noisy speech or music signals, time domain methods based on signal processing techniques such as autocorrelation or average magnitude difference, often do not perform well. As deep neural networks (DNNs) have become feasible, some researchers have attempted with some success to improve the performance of signal processing based methods by learning on autocorrelation, Fourier transform or constant-Q filter bank based representations. In our approach, blocks of signal samples are input *directly* to a neural network to perform end to end learning. The emergence of sub-harmonic structure in the posterior vector of the output layer, along with analysis of the filter-like structures emerging in the DNN shows strong correlations with some signal processing based approaches. These NNs appear to learn a nonlinearly-spaced frequency representation in the first layer followed by comb-like filters. We find that learning representations from raw time-domain signals can achieve performance on par with the current state of the art algorithms for frequency estimation in noisy and polyphonic settings. The emergence of sub-harmonic structure in the posterior vector suggests that existing post-processing techniques such as harmonic product spectra and salience mapping may further improve the performance.

Index Terms: frequency estimation, pitch detection, waveform processing, neural networks

1. Introduction

We present a neural network approach for estimating the fundamental frequency of a periodic signal directly from the time domain waveform. The goal of this work is to demonstrate the performance of fully connected neural network architectures without doing any preprocessing such as correlation or Fourier analysis or any post-processing such as dynamic programming [1,2] or pitch smoothing [3]. Fundamental frequency extraction is a problem that arises for signals in a wide variety of disciplines: EEG signals, speech signals, genomic sequences to name a few. The present work, although tested on speech signals, can be generalized to any signal of interest. In traditional signal processing approaches, a series of known complex signal processing transformations is applied in order to extract the frequency of the signal. [1,2,4]. We show here that a shallow, but large, fully-connected neural network can learn these transformations, with performance comparable to current state-of-the-art approaches. The neural network approach requires a lot of training data, and its implementation is computationally expensive. However, current work on bitwise neural networks [5,6] and compression of the network [7], may make it possible to implement such large networks efficiently.

The paper is organized as follows: Section 2 presents the related work for this task, Section 3 discusses the methodology used, followed by experimental results and discussion.

2. Related Work

Frequency estimation by time-domain, frequency-domain and hybrid signal processing methods has been an active area of research for many years [8]. Frequency estimation in polyphonic audio has been a topic of interest [3,8,9,10] and is considered a difficult problem due to similarity of the signal to the background "noise". The current state-of-the-art algorithm [11] uses a salience function which involves accumulating of frequency content in spectral bins. For frequency estimation in noisy conditions, [1] used a constant Q transform, filtered it to enhance the periodicity of the signal followed by training a neural network on the enhanced signal. There has been some work on a simpler sub-problem of frequency estimation; i.e. voice activity detection [12,13]. Recently several papers have appeared on raw time domain audio processing [14,15,16] using deep neural networks. The results in [14] showed that the first layer of a convolutional neural network (CNN) can learn gammatone filter-bank-like characteristics when trained on raw waveforms for the task of acoustic modeling. Using NNs on raw waveforms in timbral-based recognition similar to acoustic modeling in instrument identification has shown improvements over mel-filter based inputs [15]. There has also been some recent work on designing filter banks for periodicity estimation for genomic sequences [17]. The important point to note in all of these approaches for frequency estimation is the problem boils down to mappings from an n-dimensional vector (previously autocorrelation or spectrogram representation) of waveform inputs (our approach) to a single output such as the pitch state (or value). We have studied fully connected networks that can map vectors of signal samples directly to the corresponding pitch states. We have found that the network learns a frequency representation in the first layer followed by comb filtering-like processing similar to [18,19]

3. Methodology

Inspired by some of the work on raw audio waveform processing mentioned above, we have explored the possibility of estimating the fundamental frequency of a periodic signal using large fully-connected neural network architectures such as depicted in Fig. 1. The network learns to extract frequency content, unlike the previous works on raw waveform processing using deep nets which focused on the timbral component of the signals. Techniques such as described in [1, 10] have used operations such as salience mapping and convolution operations on spectrogram slices in order to

http://dx.doi.org/10.21437/Interspeech.2016-679

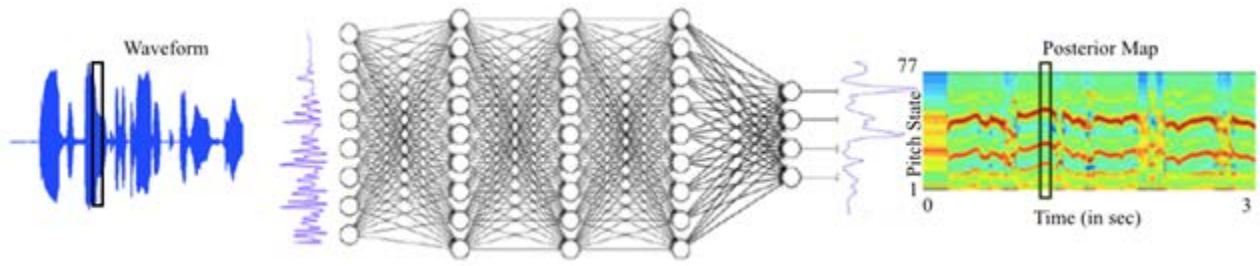

Figure 1: *Network architecture of transformation of audio waveform into stacked normalized posterior vector representation*

enhance periodicity. Our goal is to estimate the frequency directly, starting with blocks of waveform samples. In order to determine the fundamental frequency, the input signal vectors must include at least two periods. We quantize the frequency range into 24 states per octave and an additional state indicates silence or unvoiced speech as in [1]. (A total of 77 pitch states was used in current experiments for the TIMIT dataset). This also allows us to compute the accuracy on the MIREX-1k dataset for frequency extraction within 50cents. (Frequency doubling corresponds to 1200 cents). We do not train recurrent architecture since the performance of recurrent networks was shown to give marginal performance gains on spectrogram input when given suitable context of input signal [1]. The input context was a 480 dim vector of waveform samples (60ms for 8 kHz sampling rate) with the corresponding pre-measured pitch state corresponding to the center of the vector. In our first experiment depicted in Fig. 1, we trained a 3-layer fully-connected network using speech from the TIMIT data base with the pitch states at 10ms intervals being the output in the final layer with cross entropy loss as the objective function [1]. Each layer had 2048 neurons with RELU nonlinearity [20]. Dropout regularization is used in each layer, which helped to prevent over-fitting [21]. Stochastic gradient descent with momentum was used to update the weights using the backpropagation algorithm. The simulations were run using the Caffe framework [20]. To use the trained network, we pass a 480 sample block of the signal every 10ms through a 3-layer fully connected network. The posterior vector is normalized to a maximum of 1 for better visualization in Fig. 1. The posterior vectors were stacked as we would for Fourier slices in a spectrogram. The pitch state for a given waveform block is determined as the maximum probability in the posteriori vector.

In training the networks, the hyper parameters tuned were learning rate, weight regularization (L2 norm), and the dropout rates in each layer in a random grid. A total of 100 combinations were run for each of the fixed network architectures. The optimum parameters were chosen from the top five validation performances. The best model was chosen by looking at the loss curves and training them further. The training was carried out for 40 epochs. Sub-harmonic structure emerges in the posterior vector as shown in Figure 1. The reason is that a time-domain signal that is periodic with period T is also periodic with periods 2T, 3T. These correspond to the frequencies of f0/2, f0/3, etc. and strong peaks in the corresponding posterior vectors. The best trained network also distinguishes between speech/silences and, when trained on polyphonic audio, the system does voicing detection in the difficult setting of polyphonic audio,

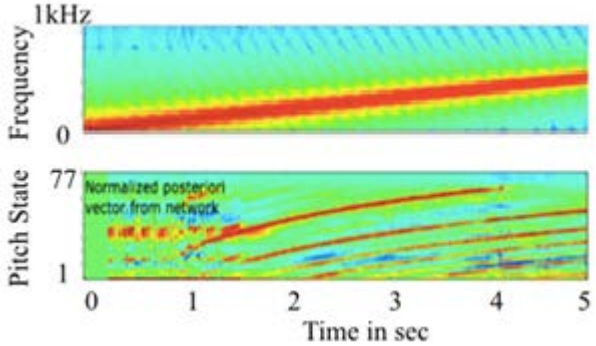

Figure 2: *Performance on chirp signals of network trained on TIMIT. Notice the emergence of sub-harmonic structure and behavior after 400 Hz.*

where it learns to distinguish different kinds of harmonic sounds.

To see the behavior of the network on a general signal, we used a linear chirp signal as the input to a network trained on clean speech signals. The strongest peak in the posteriori vector corresponds to the frequency prediction. (The upper spectrogram has a linear frequency scale while the posteriori scale is nonlinear.) As seen from Figure 2, the network predicts silence in the frequencies below about 100 Hz void of the sub-harmonic structure. The reason is that the TIMIT dataset contains mostly speech with pitch in the range of 100-400 Hz, and the trained network is not confident for the class of labels it has not seen before. Since the network is trained only in the range below 400Hz, it is unable to predict the correct pitch state for the chirp signal beyond 400Hz. However since the signal is also periodic with period 2T, it predicts the state corresponding to f0/2 with highest probability. This also validates that the network is not over fitting to a specific kind of signal and is able to generalize its performance.

4. Datasets

For training and validation in our experiments, we have used the TIMIT and MIREX [22] datasets. TIMIT has English sentences spoken in a clean environment. In order to compare the results for noisy settings, with that of [1], we added babble noise at different SNR levels from non-speech sound datasets [23]. The MIREX Chinese pop song dataset includes ground truth, which was previously extracted using the clean singing channel, and thus our results are comparable with other work that use this as ground truth. However for TIMIT, no ground truth pitch data is available. Therefore we employed an off-

the-shelf pitch detector provided by [24] to get the ground truth data to train on. The input in all the cases was downsampled to 8 kHz primarily to reduce the computational complexity and size of the training vector. The noises were added at -5, 0, +5db. For the case of MIREX, we used 200 randomly chosen songs for testing and the remaining 800 songs for training and validation. The training data and testing data are a mixture of vocal and instrumental parts mixed at 0 and +5 db. The evaluation is also performed at the same signal-to-noise ratios. We performed evaluation on clean speech, speech mixed with babble noise, and polyphonic audio which can be treated as added noise to singing voice that is both harmonic and non-harmonic.

5. Experiments and Results

Using the MIREX 1k dataset, we experimented with different combinations of number of neurons and number of layers. Convolutional neural network architectures were not tried due to sheer number of the possible network topologies to be searched [25] which may be beyond the scope of current work. The results are shown in Table 1 where we see that there is little improvement in the performance going from two to three layers, and the best 3-layer performance occurred with 4096 neurons per layer while the best 2-layer performance required 2048 neurons per layer. Also, increasing the depth to five layers results in over-fitting and poor generalization to samples outside the training set. Thus the first two layers appear to be sufficient for this task. For each of the network architectures, i.e. fixing the number of layers and neurons in each layer, we sampled 100 hyper parameter vectors and trained the given network as mentioned before. We adjusted dropout rates in each layer independently, and along with learning rates, regularization, and momentum update picking these parameters using random subsampling. We chose the hyper parameters that gave the best performance for the validation set and then continued training the network to achieve better performance. The performance of different architectures on a test set of 200 songs from the MIREX dataset is as shown in Table 1. The general trend is that system performance improves with increasing number of neurons in each layer. In Figure 3, we show the posterior map for a test segment of polyphonic audio compared with the corresponding spectrogram of the signal. The sub-harmonic structure appears in the posteriori map with less strength than shown in the Fig. 3. Notice how the network is able to learn to do voicing decisions along with accurate pitch predictions. As per the section marked in the middle of the spectrogram, we see background harmonic content present. The network learns to distinguish voice from other similar harmonic content. We predict the pitch state with the highest value in posteriori vector as the pitch estimate for the current frame of signal. Notice how the silent intervals of very small durations in between the voice phrases are also captured by this approach. The intermittent silences are smaller than the analysis frame length.

Neurons	**64**	**128**	**512**	**1024**	**2048**	**4096**
1 Layer	55.45	72.15	73.22	77.69	80.08	79.22
2 Layer	64.31	71.29	77.53	74.37	81.58	81.56
3 Layer	62.15	71.81	78.79	82.94	81.31	**83.31**
5 Layer	64.21	72.81	77.47	78.9	74	72.4

Table 1. Comparison of the performance of different architectures of hidden layer for MIREX dataset

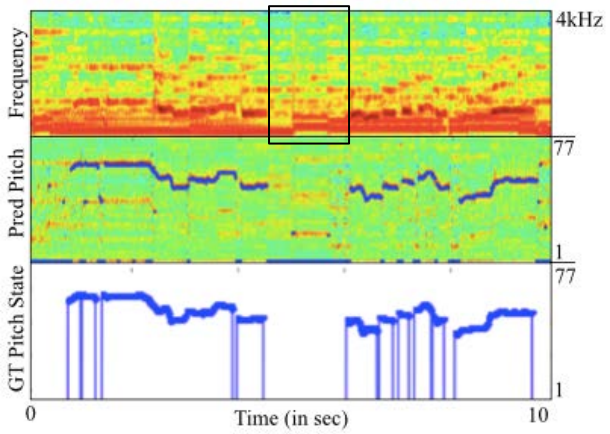

Figure 3: Comparison of the spectrogram of a polyphonic audio excerpt (top), posterior map, with the predicted pitch in blue(middle) and the ground truth(bottom)

The network learns to predict the pitch of the center of a frame accurately from a wider temporal context as seen from pitch contours of the faster transition regions. Comparing the performance of the best network, we see that since we had trained the network on a mix of 0dB and 5dB SNRs, we achieve an accuracy of **83.31**% raw pitch accuracy using the current approach. Salamon, et. al.[4], reported accuracies of **85**% and **78**% at +5db and 0db SNR respectively. The precision, recall and F-measure for just the voicing task was 0.764, 0.8015 and 0.7825 respectively. Also noteworthy is that within an analysis frame, the network picks the fundamental frequency corresponding to the singing voice even in presence of other harmonic sounds. We do not carry out any pre-processing or post processing and still achieve performance on par with the current state of the art methods. The network trained on singing voices did not perform as well on noisy speech and vice-versa, with the audio having vastly different background instrumentation. This limitation of neural nets on generalization outside the training dataset has been reported in the past, on work carried out for speech denoising [26]. Therefore, we retrained our system on part of the speech corpus in the TIMIT dataset various levels of additive noise, and then evaluated the performance on the remaining part of the database. The testing set was 1000 randomly chosen utterances and the remaining 3000 utterances comprised the set for training, with added babble noise in both the training and test sets at different SNR levels. Babble noise is probably the most widely encountered and challenging noise, apart from wind noise and traffic and thus was chosen for the study. This setup was used in order to compare the performance with that of [1]. We achieve comparable performance by retraining the 3 layered network. As the results were not available in tabular form in [1], we obtained test set accuracies of **49, 56, 65** and **79**% at 5,0,+5 and ∞ dB SNRs as compared to 42, 60, 70 and 83% SNRs as seen from the graphs in [1] respectively. Recall that ground truth data for our experiments were obtained by converting the output of an off-the shelf algorithm [24] to desired pitch states and is not same as that of [1]. Clearly, errors made by the ground algorithm will be reflected in the measured performance of the NN system. Further improvements can be obtained by using more training data and using synthetic noise augmentation techniques similar to that proposed for robust speech recognition in noise [27].

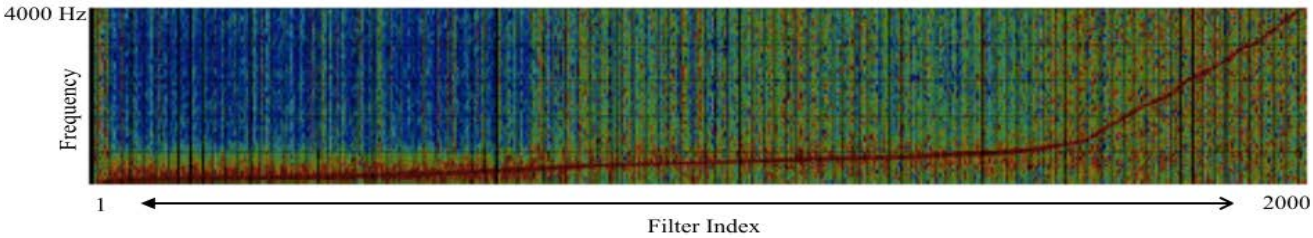

Figure 4: *Frequency response of the learned filters in the first layer sorted according to the highest peak for TIMIT. Red denotes high values whereas blue represents smaller value.*

6. Discussion

The results were surprising given that we did not do any preprocessing of the input signal or post-processing of the output. This is in contrast to approaches that use fully connected architectures on explicit auto-correlation-based representation [2], constant Q transform or spectrogram- based representations. In fact, operations in the frequency domain approaches [2, 10] appear in a sense to be learned by our networks. In order to see what the network is doing in the first and the second layers, we followed the methodology in [14] wherein we compute the Fourier transform of the learned time domain filters, i.e., the weights of each neuron, and smooth it in the frequency domain. Since the filters are learned in an arbitrary order, we sort them according to the location of the spectral peaks of the rather crude bandpass filters. The sorted filters learned in the first layer are shown in the Fig.4. We see that the passbands of the filters that are learned are piecewise uniformly distributed over the entire frequency range. Although most of the filters are assigned to the range of frequencies corresponding to the range of the target pitches, a small fraction of the filters is assigned by the training to the range of frequencies outside of this range. The outputs of these filters may be used to make voicing/unvoicing decisions in the subsequent layers. The learned filters span the entire frequency range of 0 to 4 kHz. Figure 4 also shows that the bandwidth and peak gain of these filters vary with the center frequency as there are a lot of high values present (darker regions) across the center frequencies in the filters for lower frequency ranges. Thus the very first layer appears to learn a non-linear non-constant bandwidth filter bank. Also interesting is the fact that the filters having center frequency in the range of the desired states in the filter output are more "sinusoidal" (peakiness in the spectral domain) than those outside this range. This is seen by the higher relative contrast in initial sorted filters. Thus, our experiments suggest that the first layer corresponds to learning a non-linearly-spaced, non-constant bandwidth filter bank. Traditional signal processing based approaches use techniques such as salience mapping [11,1] or comb filtering [18,19] in the second layer in order to enhance the pitch state of the interest. The two approaches are closely related and amount to summing certain frequencies in the desired frequency ranges. To interpret what is going on in our second layer, we use the single hidden layer network which had the best performance in order to see what these filters were like. We sorted the filters according to the frequency response similar to Figure 4 and stacked these 77 filters, one corresponding to each pitch state in Figure 5 with darker colors representing higher weights. This visualization was quite remarkable as it shows that each pitch state is performing summation of harmonic locations interpreted by the 1st layer filterbank corresponding to the pitch state of interest. This is also quite similar to salience mapped approach by [4]. Also it is learning different weightings to give to these peaks in a single comb filter. The filters weightings were found to have both positive/negative values and were rectified and smoothed by moving average filter of size 10 for better visualization. This point of view of filter banks followed by comb filtering suggests that there should not be a dramatic increase in performance between the best 1 layered network and the 2-layer networks. This is confirmed by the evaluation results in Table 1, which show only marginal improvement. Thus, we conclude that the neural network training algorithm has arrived at a structure that is a lot like the structures that signal processing researchers have put together by classical hierarchical design approaches over the years.

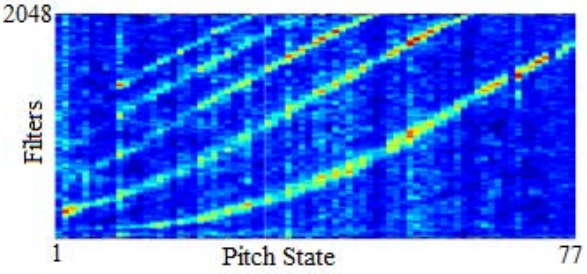

Figure 5: Weights of the learned filters in the second layer of 1 hidden layer network with 2048 neurons. Notice the comb like characteristics of these filters and strong dependencies to salience/comb filtering based approaches.

7. Future Work

The goal of this work was to show that current state of the art performance can be achieved by training fully connected networks from raw waveforms. The performance of this system can be further be improved by applying dynamic programming or nonlinear smoothing in order to correct isolated pitch errors. Further since sub-harmonics appear in the posterior vector, frame level errors can reduced by enhancing the peak corresponding to the correct frequency by existing algorithms such as saliency maps and harmonic product spectra. Since CLDNNs are a superset of the current architectures, it will be interesting to see their performance on raw waveforms.

8. Acknowledgement

The authors would like to thank Andrew Ng, his group, Stanford Artificial Intelligence Laboratory as well as Stanford Research Computing for the use of their computing resources.

9. References

[1] Han, Kun, and DeLiang Wang. "Neural Network Based Pitch Tracking in Very Noisy Speech." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 12.22 (2014): 2158-2168.

[2] Lee, Byung Suk. *Noise robust pitch tracking by subband autocorrelation classification*. Diss. Columbia University, 2012

[3] Salamon, Justin, et al. "Melody extraction from polyphonic music signals: Approaches, applications, and challenges." *Signal Processing Magazine, IEEE* 31.2 (2014): 118-134.

[4] Salamon, Justin, and Emilia Gómez. "Melody extraction from polyphonic music signals using pitch contour characteristics." *Audio, Speech, and Language Processing, IEEE Transactions on* 20.6 (2012): 1759-1770.

[5] Kim, Minje, and Paris Smaragdis. "Bitwise Neural Networks." arXiv preprint arXiv:1601.06071 (2016).

[6] Courbariaux, Matthieu, and Yoshua Bengio. "BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to+ 1 or-1." arXiv preprint arXiv:1602.02830 (2016).

[7] Han, Song, Huizi Mao, and William J. Dally. "A deep neural network compression pipeline: Pruning, quantization, huffman encoding." arXiv preprint arXiv:1510.00149 (2015)

[8] De La Cuadra, Patricio, Aaron Master, and Craig Sapp. "Efficient pitch detection techniques for interactive music." *Proceedings of the 2001 international computer music conference*. 2001.

[9] Vocal melody extraction from musical audio with pitched accompaniment, V. Rao, Department of Electrical Engineering, IIT Bombay, PhD thesis, 2011

[10] Salamon, J. (2013). Melody Extraction from Polyphonic Music Signals. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2013.

[11] Salamon, Justin, Emilia Gómez, and Jordi Bonada. "Sinusoid extraction and salience function design for predominant melody estimation." *Proc. of 14th Int. Conf. on Digital Audio Effects (DAFx-11)*. 2011.

[12] Leglaive, Simon, Romain Hennequin, and Roland Badeau. "Singing voice detection with deep recurrent neural networks." *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.

[13] Rao, Vishweshwara, Chitralekha Gupta, and Preeti Rao. "Context-aware features for singing voice detection in polyphonic music." *Adaptive Multimedia Retrieval. Large-Scale Multimedia Retrieval and Evaluation*. Springer Berlin Heidelberg, 2011. 43-57.

[14] Sainath, Tara N., et al. "Learning the Speech Front-end With Raw Waveform CLDNNs." *Proc. Interspeech*. 2015.

[15] Li, Peter, Jiyuan Qian, and Tian Wang. "Automatic Instrument Recognition in Polyphonic Music Using Convolutional Neural Networks." *arXiv preprint arXiv:1511.05520* (2015).

[16] Trigeorgis, George, et al. "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network." *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016.

[17] Tenneti, Srikanth V., and P. P. Vaidyanathan. "Ramanujan filter banks for estimation and tracking of periodicities." *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.

[18] Martin, Philippe. "Comparison of pitch detection by cepstrum and spectral comb analysis." *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*. Vol. 7. IEEE, 1982.

[19] Tadokoro, Yoshaaka, Watam Matsumoto, and Machim Yamaguchi. "Pitch detection of musical sounds using adaptive comb filters controlled by time delay." *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*. Vol. 1. IEEE, 2002.

[20] Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models." Proc. ICML. Vol. 30. 2013.

[21] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15.1 (2014): 1929-1958

[22] MIR-1k,Dataset:https://sites.google.com/site/unvoiced-soundseparation/mir-1k

[23] G. Hu, 100 Nonspeech Sounds 2006[Online]. http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html

[24] Wu, Mingyang, DeLiang Wang, and Guy J. Brown. "A multipitch tracking algorithm for noisy speech." *Speech and Audio Processing, IEEE Transactions on* 11.3 (2003): 229-241.

[25] Sainath, Tara N., et al. "Deep convolutional neural networks for large-scale speech tasks." *Neural Networks* 64 (2015): 39-48

[26] Maas, Andrew L., et al. "Recurrent Neural Networks for Noise Reduction in Robust ASR." *INTERSPEECH*. 2012..

[27] Hannun, Awni, et al. "Deep speech: Scaling up end-to-end speech recognition." *arXiv preprint arXiv:1412.5567* (2014).