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

Formant Tracking Using Dilated Convolutional Networks Through Dense Connection with Gating Mechanism

Wang Dai[1], *Jinsong Zhang*[1], *Yingming Gao*[2], *Wei Wei*[1], *Dengfeng Ke*[3], *Binghuai Lin*[4], *Yanlu Xie*[1]

[1]School of Information Science, Beijing Language and Culture University, China
[2]Institute of Acoustics and Speech Communication, TU Dresden, Germany
[3]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China
[4]MIG, Tencent Science and Technology Ltd., Beijing, China

daiwang_ai@163.com, {jinsong.zhang,xieyanlu}@blcu.edu.cn,
yingming.gao@mailbox.tu-dresden.de, wwei906@163.com,
dengfeng.ke@nlpr.ia.ac.cn, binghuailin@tencent.com

Abstract

Formant tracking is one of the most fundamental problems in speech processing. Traditionally, formants are estimated using signal processing methods. Recent studies showed that generic convolutional architectures can outperform recurrent networks on temporal tasks such as speech synthesis and machine translation. In this paper, we explored the use of Temporal Convolutional Network (TCN) for formant tracking. In addition to the conventional implementation, we modified the architecture from three aspects. First, we turned off the "causal" mode of dilated convolution, making the dilated convolution see the future speech frames. Second, each hidden layer reused the output information from *all* the previous layers through dense connection. Third, we also adopted a gating mechanism to alleviate the problem of gradient disappearance by selectively forgetting unimportant information. The model was validated on the open access formant database VTR. The experiment showed that our proposed model was easy to converge and achieved an overall mean absolute percent error (MAPE) of 8.2% on speech-labeled frames, compared to three competitive baselines of 9.4% (LSTM), 9.1% (Bi-LSTM) and 8.9% (TCN).
Index Terms: formant tracking, convolutional architecture

1. Introduction

Formants are considered to be resonances of the vocal tract during speech production. An accurate estimation of formant frequencies in spontaneous speech is often desired in many phonological experiments of laboratory phonology, sociolinguistics, and bilingualism [1, 2]. They also play a key role in the perception of speech and are useful in the coding, synthesis and enhancement of speech, as every phoneme has a unique formants distribution, especially on vowels and sonorous consonants.

Classical formant tracking algorithms are based on peak picking from Linear Predictive Coding (LPC) spectral analysis [3, 4, 5, 6]. LPC spectral coefficients yield intra-frame point estimates of candidate frequency parameters via root finding or peak-picking. The inter-frame parameter selection and smoothing can be performed by minimizing various cost functions in a dynamic programming environment [7, 8]. However, these classical approaches have an obvious shortcoming that the required root-finding or peak-picking procedure cannot be written in closed form [9]. More elaborate methods used probabilistic and statistical models to obtain confidence intervals around the estimated formant tracks [9], such as quantization of Vocal Track Resonances (VTR) space [10], Kalman filtering [9, 11, 12], HMM [13, 14, 15] and GMM [16].

The aforementioned ad-hoc signal processing methods [17] usually emerge false peaks and formant merging when affected by high pitch or coarticulation. These problems can be alleviated by visually correcting with the help of linguistic knowledge and spectral analysis. Motivated by this idea, Deng et al. released a handpicked VTR/Formants corpus in 2006 [18]. It was subsequently adopted by some researchers as benchmark dataset to develop and evaluate new algorithms for formant tracking. For example, Mehta et al. evaluated their proposed Kalman-based autoregressive moving average modeling methods on this database [9]. Inspired by the great success of deep learning in many application areas, Dissen et al. employed Long Short-Term Memory (LSTM) networks to train a supervised regression model between LPCCs plus Pitch-Synchronous Cepstrum Coefficients (named PSCCs) and hand-corrected formant frequencies for every speech frame [17]. Later, Dissen et al. [19] investigated the potential of raw spectrograms (55×50 PSCCs) for formant tracking with Convolutional LSTM networks [20] and found that incorporating the PSCCs and LPCCs achieved the better general performance than using them separately.

Recent studies showed that generic convolutional architectures can outperform recurrent networks on tasks such as speech synthesis and machine translation [21, 22]. In particular, the Temporal Convolutional Network (TCN) for sequence modeling was proposed [23], which was composed of dilated causal convolutional networks with residual connection. Stacking convolutional layers with different dilation factors can capture the long-range dependence of the sequence. Integrating different hidden features through residual connection makes model more robust. In this work, therefore, we explored whether such advantages of TCN are beneficial for formant tracking. In additon to the application of the conventional TCN model, we modified its architecture from three aspects: 1) we turned off the "causal" mode of dilated convolution, making sure the dilated convolution see the future speech frames; 2) all the dilated convolutions are closely connected, thus effectively reusing the shallow features; 3) we adopted a gating mechanism to automatically select forgetting unimportant information during training. In terms of quantitative error analysis, we compared our proposed approach with other five methods for formant tracking on the VTR test set, including WaveSurfer [7], Praat [8], LSTM model, Bi-LSTM model and TCN based model.

http://dx.doi.org/10.21437/Interspeech.2020-1804

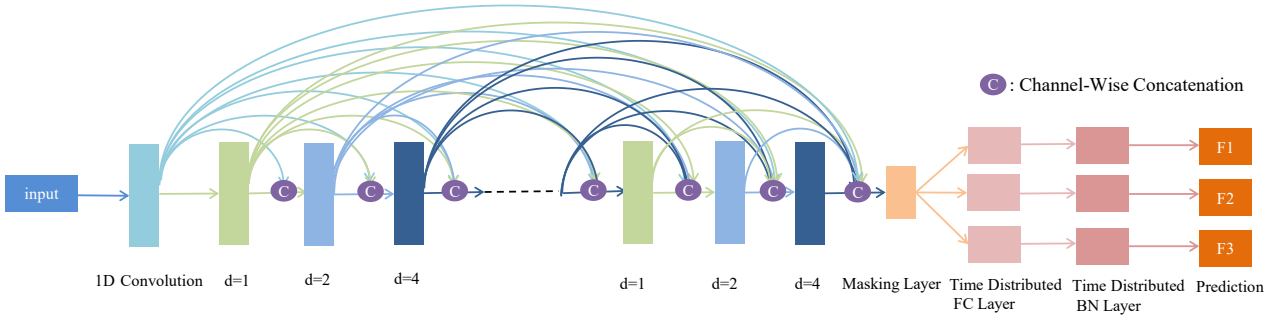

Figure 1: *Overview of proposed model framework for formant tracking.*

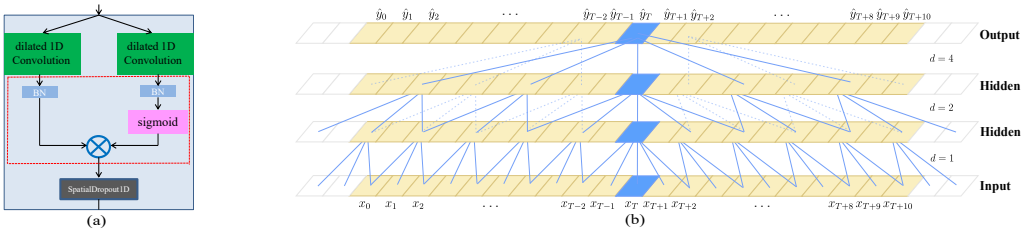

Figure 2: *(a) Gated linear unit. (b) An example of dilated 1D convolution with dilation factors* $d = 1, 2, 4$ *and filter size* $k = 3$.

2. Model Description

The architecture of proposed model for formant tracking is shown in Figure 1, our framework mainly consists of three components: (1) dilated convolutions, (2) dense connections and (3) gated linear units, all of which are described in subsequent sections.

2.1. Dilated convolutions

The generic TCN architecture uses dilated 1D causal convolution where the convolution filter is applied over an area larger than its length by skipping certain input values [23]. Compared with LSTM, the dilated causal convolution needs less nonlinear operations, making the model converge easier. The receptive field can be set to an arbitrary large size by increasing dilation factor. A major disadvantage of dilated causal convolution is the "causal" mode when handling the context-dependent sequence tasks. It can only look back at history information because the output at time T is convolved only with elements from earlier to current time step. In speech formant tracking, formants of each frame will be affected by future frames, so we turned off the "causal" mode of dilated convolution, making the dilated convolution see the future information. A common practice is to use a dilation factor sequence of form $\{1, 2, 4, 8, ...\}$. When $d = 1$, a dilated convolution reduces to a regular convolution. Using larger dilation factor yields an top level output which can capture a wider range of inputs. We provide an illustration in Figure 2(b). In this work, there are 9 1D convolutions stacked with the dilation factors $\{1, 2, 4, 1, 2, 4, 1, 2, 4\}$ to obtain long context dependence, as shown in Figure 1. Every dilated convolution layer has 64 filters with a size of 3.

2.2. Dense connections

The depth of the neural network model is important for learning advanced representations, but it is also accompanied with the challenge of gradient disappearance. Residual training [24] is considered to be an effective way to address this problem. Using this connection mode, TCN can build a very deep network. Densely connected networks were recently proposed in [25]. They can be regarded as a natural evolution version of [24] where the inputs to a given layer in the network are a concatenation of the outputs from all the previous layers. This way avoids the vanishing gradient problem in depth model. Another advantage is that each layer reused output from all previous layers, such that different level features are fused to improve the robustness of the model. Inspired by the effectiveness of dense connection, we adopted it in our model. A slight difference from [25] is that all the dilated convolutions are closely connected to capture more fine-grained features as shown in the densely connected arcs of Figure 1.

2.3. Gated linear units

There are several Gating mechanisms that had been explored in modern convolutional architectures for sequential modeling [26, 27, 28]. Parallel to our work, [27] has shown the form of $(X \times W + b) \bigotimes \sigma(X \times V + c)$ is more effective than others for language modeling. Coupling linear units to the gates, referred to as gated linear units, reduce the vanishing gradient problem. This retains the non-linear capabilities of the layer while allowing the gradient to propagate through the linear unit without scaling. Similarly, in this work, after applying the linearity to the Batch Normalization output of every dilated 1D convolution, we attenuated it with a sigmoid gate (shown in Figure 2(a)). Moreover, we used SpatialDropout1D [29] at the back of each gated linear unit to sparse the output dimensions (channels) information, thus improving the robustness of the model.

3. Experiment

3.1. Dataset

VTR corpus [18] was used in this study to evaluate our model and baselines. It contains 538 SX or SI utterances, selected as a representative subset of TIMIT corpus. Here, SX de-

notes phonetically compact utterances and SI denotes phonetically diverse utterances. The training set consists of 346, out of which 324 utterances have handpicked VTR. These 346 utterances cover 173 speakers with one SX and one SI utterance from each speaker. The test set consists of 192 utterances covering 24 speakers, and each speaker has 5 SX utterances and 3 SI utterances. Both training and test sets were first annotated by an automatic formant tracking algorithm [11], and subsequently hand-corrected for every 10 ms frame by a group of phonologists based on visual inspection of the first three formants in the spectrogram. We further set aside 24 utterances of 12 speakers (fecd0, mgrl0, falk0, mjrh1, fpaf0, mtrt0, fcdr1, mwsh0, fbch0, msjk0, fjrp1, mdlc1) from the training set as the validation set.

3.2. Data preprocessing

Following the study [19], we used the same acoustic features (LPCCs + PSCCs). A new frame of 30 ms consisted of three original frames of 10 ms in VTR, formant values of which were averaged. In our experiments, we used MATLAB software to extract $8 \sim 17$ order 30-dim LPCCs (total 300-dim). We first removed the DC component and applied a pre-emphasis filter ($H(z) = 1 - 0.97z^{-}1$) to the input speech signal. Then the input signal was divided into frames, and the acoustic features were extracted from each frame. The frame shift was 10 ms, and frames were overlapping with Hamming windows of 30 ms. The 50-dim PSCCs was directly extracted from Dissen's open source code [19].

3.3. Loss function

Dissen et al. [19] used a fully connected layer with 3 neurons as the output layer to predict F1, F2 and F3 of each speech frame. The high level features, i.e., the output of the last hidden layer, were shared by each formant but not specific. In fact, there is an inner relationship between formants, each of which has a specific frequency band. Inspired by the success of multitask output [30], we adopted a similar hard parameter sharing structure. In our framework, there were three parallel branches of fully-connected layers with 256 neurons from dilated convolutional networks. Finally, each of them was linearly transformed to predict the formant. The formant prediction was consider to be independent but mutually restricted to each other in this way. The error between output and reference formant frequency was optimized by the following objective function,

$$\mathcal{L} = \alpha \times \mathcal{L}_{F1} + \beta \times \mathcal{L}_{F2} + \gamma \times \mathcal{L}_{F3} \quad (1)$$

where $\mathcal{L}$ is the sum loss of first three formants prediction. $\mathcal{L}_{F1}$, $\mathcal{L}_{F2}$, and $\mathcal{L}_{F3}$ are the losses for F1, F2 and F3, respectively. α, β, and γ represent the weights for the three losses, and they are set to the same value of $1/3$ as each formant prediction deemed to be equally important. To make a fair comparative study, this loss function is applied to all baseline models.

3.4. Training configuration

The following experiment settings were also applied to all deep learning models including the baselines. The deep learning toolkit used in this work is Keras. The loss function to minimize was mean absolute error, and we used Adam [31] as the optimizer. The initial learning rate of optimizer was set to 0.001 and decreased by 0.0005 after training 50 epochs. All configurations were trained for maximum 100 epochs with a batch size of 4 spoken utterances. The model which had the smallest loss on validation set was selected. Silent segments at both ends of utterances were not trained and evaluated. We fixed a maximum length of 710 frames on VTR dataset. Short utterances were padded zeros if they were shorter than the fixed maximal length. During training and testing, we used the Masking layer of Keras to locate the zero time step to be skipped.

3.5. Baselines

The LSTM tracking model was trained using the same model configuration from [19] except for the previously mentioned optimizer and loss function. On this basis, we trained the Bi-LSTM tracking model by replacing the LSTM layers with Bi-directional LSTM layers. We further trained the TCN based tracking model using the same model parameter settings for our proposed model. In addition to the three neural network formant tracking models, we also extracted formants using two widely used speech analysis tools: WaveSurfer and Praat.

3.6. Metrics

Two quality measures were calculated to quantify the distance of formant tracker output to the annotation reference:

- MAE: mean absolute error between reference and formant tracker output calculated over speech frames.
- MAPE: mean absolute percent error between reference and formant tracker output calculated over speech frames.

The smaller the value is, the more the formant tracker output matches the reference.

4. Results and Discussion

Figure 3 shows the training (dashed lines) and validation (solid lines) loss for different neural network models. Although they follow the same trend in the beginning stage, the LSTM based models (curves with obvious fluctuations as shown in Figure 3) appear to be more difficult to converge than other models after the first 20 epochs, even over-fitting happened to the Bi-LSTM model. With a faster convergence speed, "Ours" achieved even better performance than the TCN model.

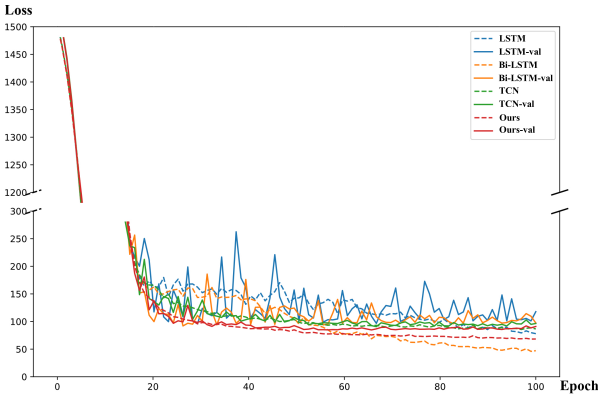

Figure 3: *Training and validation loss for different neural network models.*

Tables 1-4 present the quantitative error analysis for our model. Different from [19] where they trained models on a subset but testing on the whole dataset, we assured that there was

Table 1: *MAE(Hz) and MAPE(%) on whole VTR test utterances*.

	WaveSurfer		Praat		LSTM		Bi-LSTM		TCN		Ours	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
F1	111	26.7	232	55.0	89	16.9	86	16.6	82	15.2	**73**	**15.0**
F2	160	11.1	301	21.6	94	6.6	89	6.1	88	5.7	**80**	**5.6**
F3	255	10.6	368	15.6	119	4.8	115	4.6	111	4.2	**98**	**4.0**
Overall	176	16.1	300	30.7	101	9.4	97	9.1	94	8.9	**84**	**8.2**

Table 2: *MAE(Hz) on broad phone classes*.

	WaveSurfer			Praat			LSTM			Bi-LSTM			TCN			Ours		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
vowels	52	94	156	101	158	219	61	65	95	60	68	99	52	58	86	**46**	**56**	**79**
semivowels	67	119	218	120	243	309	76	86	134	74	80	128	69	86	125	**60**	**72**	**108**
nasal	102	324	285	191	406	366	71	153	142	69	139	136	67	152	126	**60**	**133**	**108**
fricatives	255	269	490	572	624	731	145	129	144	141	113	131	142	122	138	**128**	**105**	**121**
affricatives	287	330	372	779	551	607	180	159	178	169	**135**	174	168	140	174	**161**	143	**161**
stops	149	151	264	278	285	394	129	114	138	124	103	127	124	108	143	**111**	**98**	**121**

Table 3: *MAPE(%) on broad phone classes*.

	WaveSurfer			Praat			LSTM			Bi-LSTM			TCN			Ours		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
vowels	10.7	6.2	6.6	22.3	11.0	9.5	11.3	4.4	3.8	11.1	4.5	3.9	9.8	3.9	3.6	**8.9**	**3.7**	**3.2**
semivowels	14.9	9.9	9.7	27.5	23.1	14.4	14.8	7.7	5.7	14.5	6.7	5.3	13.7	7.8	5.4	**12.4**	**6.4**	**4.6**
nasal	24.4	22.4	11.5	48.3	30.9	15.2	15.5	11.2	5.6	15.1	9.8	5.4	15.0	11.3	5.0	**13.8**	**9.7**	**4.3**
fricatives	65.5	18.6	19.7	138.2	42.6	29.7	27.9	9.0	5.7	27.5	7.6	5.1	27.4	8.5	5.6	**26.7**	**7.2**	**4.8**
affricatives	75.0	18.7	14.4	190.4	30.9	23.9	32.8	8.5	6.6	31.8	**7.3**	6.5	**31.3**	7.6	6.6	32.6	7.7	**6.1**
stops	36.0	10.3	11.2	65.1	20.8	17.1	24.3	7.9	5.8	23.6	6.8	5.3	23.6	7.5	6.1	**22.7**	**6.7**	**5.1**

Table 4: *MAE(Hz) on CV transitions and VC transitions*.

	WaveSurfer			Praat			LSTM			Bi-LSTM			TCN			Ours		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
CV transitions	253	316	463	538	562	731	113	151	157	110	139	147	112	153	162	**101**	**133**	**140**
VC transitions	245	291	451	523	526	703	110	145	152	108	134	143	110	146	157	**99**	**127**	**135**

no overlap between the training set and the test set, thus objectively evaluating the generalization performance of supervised model.

Table 1 shows the precision of F1, F2, F3 and overall in MAE and MAPE. (Note that the results were only caculated over speech-labeled frames [9] for the formant estimation of non-speech is meaningless.) From this table, we can see neural network models/trackers significantly outperformed WaveSurfer and Praat. The results also show the effectiveness of convolutional architectures for formant tracking. It is worth mentioning that our model achieved the smallest error rate even compared with the advanced Bi-LSTM and TCN model. We further categorized the speech sounds to six categories like [19].

Table 2 and Table 3 respectively presents the accuracy in MAE and MAPE for each broad phone class. The depth formant tracking models outperformed Praat and WaveSurfer almost in every category, except that WaveSurfer had a better estimation of F1 on vowels. The TCN model had a better accuracy than the LSTM and Bi-LSTM model on vowels, semivowels (excepts for F2) and nasal (excepts for F2). The overall best performance on almost every phone was achieved by our proposed model, excepts for F2 of affricatives.

We also examined the errors of the algorithms when limiting the error-counting regions to only the consonant-to-vowel (CV) and vowel-to-consonant (VC) transitions. In this study, the number of frames of the transition regions were not fixed like [19] as the CV or VC boundary was not known in advance in practical application. Our model also achieved the best precision among all methods.

5. Conclusions

In this paper, we proposed a novel temporal convolutional network upon the conventional TCN model for formant tracking. The "causal" mode of dilated convolution was turned off to capture the impact of speech context. Each layer reused the output from *all* previous layers through the dense connection. With the gating mechanism, the model selectively forgets unimportant information. The approach was validated on an open access dataset. The experimental results showed that our model achieved the best performance on almost all broad phone classes and transitions, compared to LSTM based models and TCN model. In the future, we will consider estimating the formants of vowels segments and investigating whether pre-training is helpful for this task.

6. Acknowledgements

This work is supported by National Social Science Foundation of China (18BYY124), Advanced Innovation Center for Language Resource and Intelligence (KYR17005), Discipline Team Support Program of Beijing Language and Culture University (GF201906), Wutong Innovation Platform of Beijing Language and Culture University (19PT04), the Fundamental Research Funds for the Central Universities, and the Research Funds of Beijing Language and Culture University (20YCX158). The corresponding author of the paper is Yanlu Xie.

7. References

[1] C. G. Clopper and T. N. Tamati, "Effects of local lexical competition and regional dialect on vowel production," *The Journal of the Acoustical Society of America*, vol. 136, no. 1, pp. 1–4, 2014.

[2] B. Munson and N. P. Solomon, "The effect of phonological neighborhood density on vowel articulation," *Journal of speech, language, and hearing research*, vol. 47, no. 5, pp. 1048–1058, 2004.

[3] S. McCandless, "An algorithm for automatic formant extraction using linear prediction spectra," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 2, pp. 135–141, 1974.

[4] L. Deng and C. D. Geisler, "A composite auditory model for processing speech sounds," *The Journal of the Acoustical Society of America*, vol. 82, no. 6, pp. 2001–2012, 1987.

[5] K. Steiglitz, "On the simultaneous estimation of poles and zeros in speech analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 229–234, 1977.

[6] B. S. Atal and M. Schroeder, "Linear prediction analysis of speech based on a pole-zero representation," *The journal of the acoustical society of America*, vol. 64, no. 5, pp. 1310–1318, 1978.

[7] K. Sjölander and J. Beskow, "Wavesurfer-an open source speech tool [computer program]. version 1.8.5," in *Sixth International Conference on Spoken Language Processing*, 2000.

[8] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer [computer program]. version 6.0. 37," *RetrievedFebruary*, vol. 3, p. 2018, 2018.

[9] D. D. Mehta, D. Rudoy, and P. J. Wolfe, "Kalman-based autoregressive moving average modeling and inference for formant and antiformant tracking," *The Journal of the Acoustical Society of America*, vol. 132, no. 3, pp. 1732–1746, 2012.

[10] L. Deng, A. Acero, and I. Bazzi, "Tracking vocal tract resonances using a quantized nonlinear function embedded in a temporal constraint," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 2, pp. 425–434, 2006.

[11] L. Deng, L. J. Lee, H. Attias, and A. Acero, "A structured speech model with continuous hidden dynamics and prediction-residual training for tracking vocal tract resonances," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2004, pp. I–557.

[12] L. Deng, L. J. Lee, H. Attias, and A. Acero, "Adaptive kalman filtering and smoothing for tracking vocal tract resonances using a continuous-valued hidden dynamic model," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 1, pp. 13–23, 2006.

[13] G. Kopec, "Formant tracking using hidden markov models and vector quantization," *IEEE transactions on acoustics, speech, and signal processing*, vol. 34, no. 4, pp. 709–729, 1986.

[14] M. Lee, J. Van Santen, B. Mobius, and J. Olive, "Formant tracking using context-dependent phonemic information," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 741–750, 2005.

[15] D. T. Toledano, J. G. Villardebó, and L. H. Gómez, "Initialization, training, and context-dependency in hmm-based formant tracking," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 2, pp. 511–523, 2006.

[16] J. Darch, B. Milner, and S. Vaseghi, "Map prediction of formant frequencies and voicing class from mfcc vectors in noise," *Speech communication*, vol. 48, no. 11, pp. 1556–1572, 2006.

[17] Y. Dissen and J. Keshet, "Formant estimation and tracking using deep learning." in *INTERSPEECH*, 2016, pp. 958–962.

[18] L. Deng, X. Cui, R. Pruvenok, J. Huang, S. Momen, Y. Chen, and A. Alwan, "A database of vocal tract resonance trajectories for research in speech processing," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1. IEEE, 2006, pp. 369–372.

[19] Y. Dissen, J. Goldberger, and J. Keshet, "Formant estimation and tracking: A deep learning approach," *The Journal of the Acoustical Society of America*, vol. 145, no. 2, pp. 642–653, 2019.

[20] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[21] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[22] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[23] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[26] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv: Computation and Language*, 2017.

[27] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv: Computation and Language*, 2016.

[28] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. V. Den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *arXiv: Computation and Language*, 2016.

[29] J. Tompson, R. Goroshin, A. Jain, Y. Lecun, and C. Bregler, "Efficient object localization using convolutional networks," *arXiv: Computer Vision and Pattern Recognition*, 2014.

[30] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.