Sequence-to-Sequence CNN-BiLSTM Based Glottal Closure Instant Detection from Raw Speech

Jindřich Matoušek[1,2]*, Daniel Tihelka*[2]

[1]Department of Cybernetics, [2]New Technology for the Information Society (NTIS)
Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Rep.

jmatouse@kky.zcu.cz, dtihelka@ntis.zcu.cz

Abstract

In this paper, we propose to frame glottal closure instant (GCI) detection from raw speech as a sequence-to-sequence prediction problem and to explore the potential of recurrent neural networks (RNNs) to handle this problem. We compare the RNN architecture to widely used convolutional neural networks (CNNs) and to some other machine learning-based and traditional non-learning algorithms on several publicly available databases. We show that the RNN architecture improves GCI detection. The best results were achieved for a joint CNN-BiLSTM model in which RNN is composed of bidirectional long short-term memory (BiLSTM) units and CNN layers are used to extract relevant features.

Index Terms: glottal closure instant detection, deep learning, recurrent neural network, convolutional neural network

1. Introduction

Deep learning has recently been successfully applied in many areas of signal processing, replacing the established and refined signal processing techniques (such as autocorrelation, convolution, Fourier and wavelet transforms and many others), or speech/audio processing techniques (such as Gaussian mixture models or hidden Markov models) [1]. Deep *convolutional neural networks* (CNNs) were also shown to beat traditionally used algorithms for *glottal closure instant detection* [2] (such as SEDREAMS [3] or DYPSA [4]) [5, 6, 7, 8, 9].

Detection of glottal closure instants (GCIs) could be viewed as a task of determining peaks in the *voiced parts* of the speech signal that correspond to the moment of glottal closure, a significant excitation of the vocal tract during speaking. Accurate location of GCIs can be beneficial in many practical applications, especially in those where *pitch-synchronous* speech processing is required [2, 10, 11].

From the point of view of machine learning, GCI detection could be described as a two-class classification problem: whether or not a peak in a speech waveform represents a GCI. Unlike the classical ("non-deep") machine learning, deep learning, and especially CNNs, can help solve the problem of identifying features. In general, deep learning can help in finding more complex dependencies between raw speech and the corresponding GCIs. CNNs can directly be applied to the raw speech signal without requiring any pre- or post-processing, such as feature identification, extraction, selection, dimension reduction, etc. – steps that must be carried out in the case of classical machine learning [7, 12]. This may be the main reason why studies are mostly limited to CNN-based architectures; only a few studies seem to have investigated recurrent architectures in the context of deep learning-based GCI detection [13].

In this paper, we propose to frame GCI detection as a sequence-to-sequence prediction problem and we explore the potential of *recurrent neural networks* (RNNs), specifically their gated variants, *long short-term memory* (LSTM) networks and *gated recurrent units* (GRUs) to handle this problem. We also examine a joint CNN-BiLSTM architecture which involves using CNN layers for feature extraction on raw speech data combined with *bidirectional* LSTMs to support sequence prediction.

2. Data Description

2.1. Speech Material

Experiments were performed on clean 16 kHz sampled speech recordings primarily intended for speech synthesis. We used 3200 utterances from 16 voice talents (8 male and 8 female voices with 200 utterances per voice) of different languages (8 Czech, 2 Slovak, 3 US English, Russian, German, and French). Two voices were from CMU ARCTIC database [14, 15] (Canadian English JMK and Indian English KSP), the rest were our proprietary voices. For our purposes, speech waveforms were mastered to have equal loudness and negative polarity (dominant peaks are under zero) [16]. Ground truth GCIs were detected from contemporaneous electroglottograph (EGG) recordings by the Multi-Phase Algorithm (MPA) [17] and shifted towards the neighboring minimum negative sample in the speech signal. 3136 utterances (196 from each voice) were used for training, the rest was used for tuning and validation.

2.2. GCI Detection Measures

GCI detection techniques are usually evaluated by comparing locations of the detected and reference GCIs. The measures typically concern the *reliability* and *accuracy* of the GCI detection algorithms [4]. The former includes the percentage of glottal closures for which exactly one GCI is detected (*identification rate*, IDR), the percentage of glottal closures for which no GCI is detected (*miss rate*, MR), and the percentage of glottal closures for which more than one GCI is detected (*false alarm rate*, FAR). The latter includes the percentage of detections with the identification error $\zeta \leq 0.25$ ms (*accuracy to* ± 0.25 *ms*, A25) and standard deviation of the identification error ζ (*identification accuracy*, IDA).

In addition, we use a more *dynamic evaluation measure* [18]

$$E10 = \frac{N_{GT} - N_{\zeta > 0.1 T_0} - N_M - N_{FA}}{N_{GT}} \qquad (1)$$

that combines the reliability and accuracy in a single score and reflects the local *pitch period* T_0 pattern (determined from the ground truth GCIs). N_{GT} stands for the number of reference GCIs, N_M is the number of missing GCIs (corresponding to MR), N_{FA} is the number of false GCIs (corresponding to FAR),

This research was supported by the Czech Science Foundation (GA CR), project No. GA22-27800S.

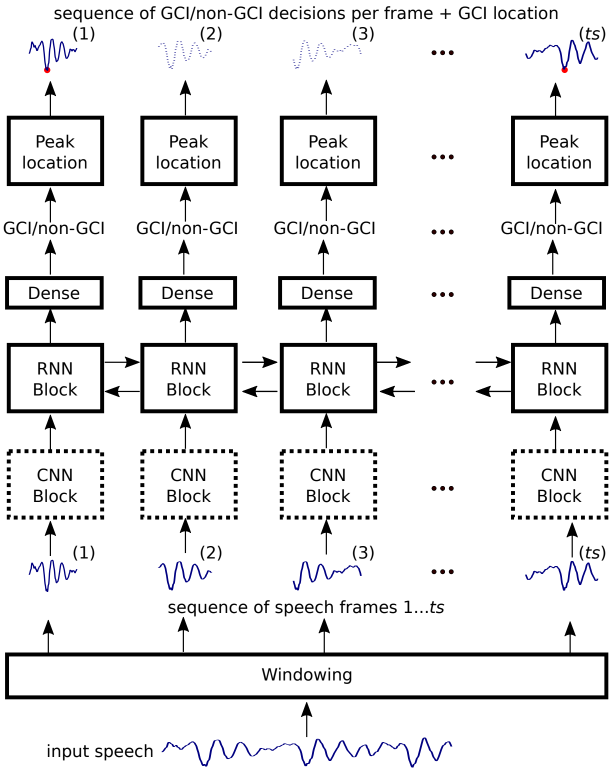

Figure 1: *A simplified scheme of the proposed CNN-BiLSTM based GCI detection. The CNN block (dotted line) works as a feature extractor. When omitted, RNN-based GCI detection on raw speech is performed. For 16kHz input speech, three BiLSTM layers with 256 cells in each layer and 900 time steps (ts) were used in RNN blocks. In CNN blocks, three convolutional blocks with two convolutional layers in each block followed by batch normalization and maximum pooling layers were used (with the number of filters 16, 32, 64, kernel size 7 with stride 1, pooling size 3, and "same" padding). The dense layer outputs a prediction whether or not a frame contains a GCI. The dotted speech signals at the top indicate that no GCI was detected in the corresponding speech frames; otherwise,* ● *marks GCI location.*

and $N_{\zeta>0.1T_0}$ is the number of GCIs with the identification error ζ greater than 10% of the local pitch period T_0. For the alignment between the detected and ground truth GCIs, dynamic programming was employed [18].

As we consider reliability more important than accuracy (we prefer better identification over absolute accuracy in GCI location), the proposed models were tuned with respect to IDR (and also E10) measures.

3. Models

3.1. Baseline CNN-Based GCI Detection System

We used the CNN-based GCI detection architecture proposed in [9] as the baseline system. Specifically, we used a one-dimensional InceptionV3-1D model that achieved the best GCI detection results.

Since CNNs predict each GCI independently on previous/next GCIs, the detection of peaks as GCI/non-GCI can be carried out in a *peak-by-peak manner* [9, 19]. In this scenario, negative peaks were detected by zero-crossing low-pass filtered (by a zero-phase Equiripple-designed filter with 0.5 dB ripple in the pass band, 60 dB attenuation in the stop band, and with the cutoff frequency of 800 Hz) speech signal exactly in the same way as described in [19]. It was also found that downsampling to 8 kHz prior filtering provided slightly better results than the use of 16 kHz. Thus, the baseline InceptionV3-1D GCI detection model use 8 kHz internally. We also tried to employ a *frame-by-frame* detection (explained further in Section 3.2) but we got worse results.

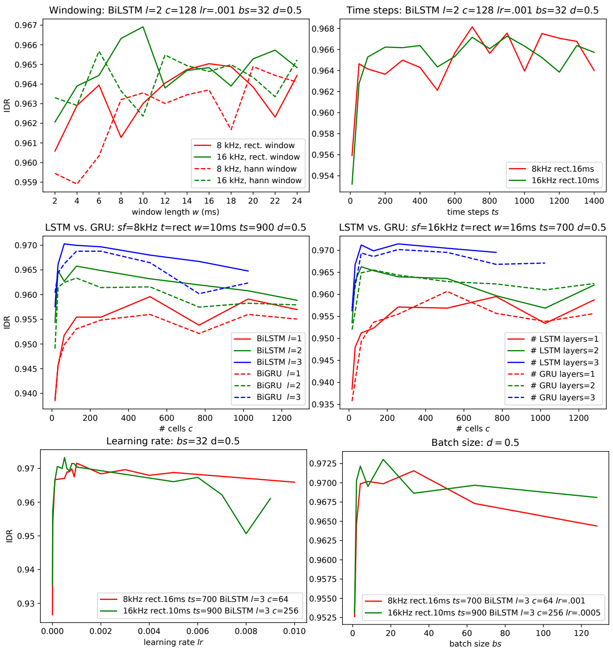

Figure 2: *The influence of different hyper-parameters on the RNN based GCI detection performance (in terms of IDR).*

3.2. Recurrent Neural Network-Based GCI Detection

While convolutional neural networks were shown to perform well on the GCI detection task, their disadvantage is that they do not take the temporal dependencies of GCIs (and the temporal structure of speech in general) into account. Since vocal folds vibrate during speaking in a quasiperiodic way, generating a GCI on each glottal closure, there is a temporal pattern present in the resulting speech signal which is not captured by the CNN architecture.

On the other hand, recurrent neural networks (RNNs) are capable of capturing the temporal structure present in the input time series data [20, 21]. The input speech signal can be viewed as a sequence of frames consisting of speech samples, such that each frame depends on previous (in the case of a bidirectional architecture also on next) frames; and RNNs can then incorporate the dependencies between these speech frames. RNNs are often thought to have the concept of "memory" (or internal state) that helps them to store the states or information of previous (and next in the case of the bidirectional architecture) inputs to generate the actual output of the sequence.

Unlike the CNN based detection, where GCIs were detected in a *peak-by-peak* manner (see Section 3.1), a *frame-by-frame* detection was carried out for the RNN based detection to capture the temporal structure of the input speech signal. In this way, speech signal was divided into overlapping frames using a sliding

Table 1: *Comparison of RNN-based GCI detection on the validation set. The model name consists of an RNN unit, number of layers, number of cells in each layer, sampling frequency in kHz, and* ***f****rame- or* ***p****eak-based detection.*

Model	**IDR** (%)	**A25** (%)	**E10** (%)
BiLSTM3-256-16f	**97.24**	97.76	**95.89**
BiLSTM3-64-8f	97.16	97.58	95.86
BiGRU3-128-16p	96.01	98.60	94.92
BiLSTM3-256-8p	95.89	**98.62**	94.82

Table 2: *Comparison of CNN-RNN GCI detection on the validation set with different CNN models and the best RNN model. The last number in model names denotes sampling frequency in kHz.*

Model	**IDR** (%)	**A25** (%)	**E10** (%)
CNN3-16	**97.49**	98.95	**96.68**
CNN3-8	97.37	99.03	96.63
InceptionV3-16	97.41	99.01	96.65
InceptionV3-8	96.45	99.04	95.73
SwishNet-16	96.96	99.00	96.20
SwishNet-8	96.54	99.02	95.81
VGG11-16	96.14	99.01	95.35
VGG11-8	95.68	**99.07**	94.95

window of a given length w and given hop length h. Note that no speech filtering and peak detection is performed here. Frame-based detection proved to be better than the peak-based one (see Table 1).

So, using RNNs, GCI detection could be viewed as a sequence-to-sequence prediction problem. In this framework, for each input sequence of speech frames, an output sequence of the the same length is predicted assigning to each frame a prediction of whether or not the frame contains a GCI. If the frame contains the GCI, the minimum negative sample in the frame is selected as the GCI. The length of the sequences is often referred to as a number of *time steps*. If the input signal contained fewer frames than the given number of time steps, it was zero-padded accordingly. As it is well-known that simple RNNs are prone to training problems known as vanishing or exploding gradient, we used their gated variants, *long short-term memory* (LSTM) networks [22] and *gated recurrent units* (GRU) [23], to alleviate the problems. Finally, a *dense* (fully connected) layer is stacked on the top of the recurrent layers to output a prediction. A simplified scheme of RNN-based GCI detection is given in Figure 1.

There are several hyper-parameters that should be experimented with when training a RNN model for our purposes. The following ones were taken into account in our comparison and tuned on the validation set: sampling frequency (sf={8, 16} kHz), window type (t = {rectangular, von Hann}), window length (w=2-24 ms), number of time steps (ts=10-1400), RNN type (r = {LSTM, GRU}), number of RNN cells (c=16-1280), number of recurrent layers (l=1-3), learning rate (lr=0.000001-0.01), mini-batch size (bs=1-128), and dropout to avoid overfitting (d=0.0-0.9 with d=0.5 giving the best results). The hop length was set to h=2 ms as this value corresponds to the minimum possible pitch period (assuming that the highest vocal fold frequency in our data is 500 Hz). The influence of different hyper-parameters on the GCI detection performance is shown in Figure 2. Briefly, lower learning rates and mini-batch sizes and higher number of time steps are preferred, 3-layer architectures with LSTMs are better, smaller (for 16 kHz) and longer (for 8 kHz) rectangular windows are a good choice.

In all experiments, the networks were trained to minimize a *binary cross-entropy loss* using *mini-batch gradient descent* with the *Adam optimizer*. Default activation functions, *tanh* and *sigmoid*, were applied in the recurrent layers, and *sigmoid activation* was used in the last (dense) layer. To speed up the training, it was stopped when the validation loss did not improve for 10 epochs and the maximum number of epochs was set to 100. Bidirectional versions of the recurrent models, i.e. BiLSTM and BiGRU, were used.

3.3. CNN-BiLSTM GCI Detection

In the next series of experiments, we examined a joint CNN-RNN architecture in which the feature extraction power of CNNs is combined with the ability of RNNs to capture the temporal structure of the input time series data and to model temporal dependencies between a sequence of speech frames [24]. Specifically, CNNs were used to extract GCI detection-relevant features from input raw speech data, and simultaneously, RNNs were used both to interpret the features across time steps and to detect GCIs.

We experimented with several CNN architectures CNNn where n is a number of convolutional blocks. Each block typically consisted of two convolutional layers followed by batch normalization, dropout and maximum pooling layers. We also tried some more complex models – the InceptionV3-1D model which yielded the best results in the CNN-based GCI detection [9], see Section 3.1, 1D version of VGG11 (a lightweight version of the well-known VGG architecture proposed for image processing [25]), and SwishNet which was proposed directly for audio processing [26].

4. Results

4.1. Comparison of Proposed Models

As can be seen in Table 1 and Figure 2, the best results for the RNN-based detection described in Section 3.2 were achieved for the 3-layered BiLSTMs with 256 cells in each layer and 16kHz frame-based speech input (BiLSTM3-256-16f). 10ms-long (w=10) rectangular window, 900 time steps (ts=900), learning rate lr=0.0005, and mini-batch size bs=16 were the best options. Finally, this model (hereinafter referred to simply as BiLSTM) was finalized, i.e., trained both on train and validation datasets, and evaluated on the evaluation datasets in Section 4.2. For the experiments with a joint CNN-RNN architecture in Section 3.3, the best model on 8kHz frame-based speech input, i.e. 3-layered BiLSTMs with 64 cells in each layer was used as well. The best setting for this model was to use 16ms-long rectangular window (w=16), ts=700, lr=0.001, bs=32.

As for the joint CNN-RNN architecture (with the best RNNs for each sampling frequency) described in Section 3.3, the best results were achieved for the simple architecture with 3 convolutional blocks (CNN3) and 16kHz speech input (see CNN3-16 in Table 2). The best setting found was as follows: the number of filters in the blocks 16, 32, 64, the kernel size 7 with the stride of 1, the pooling size 3, and the padding was "same" (please see e.g. [12] for a closer explanation). Again, the resulting model (referred to as CNN-BiLSTM) was finalized and evaluated on the evaluation datasets in Section 4.2.

4.2. Comparison of Different GCI Detection Models

We compared the proposed BiLSTM and CNN-BiLSTM with the convolutional network InceptionV3-1D [9], with a classical

Table 3: *Comparison of GCI detection of the proposed BiLSTM and CNN-BiLSTM models with other models and algorithms.*

Dataset	**Method**	**IDR** (%)	**MR** (%)	**FAR** (%)	**IDA** (ms)	**A25** (%)	**E10** (%)
BDL	CNN-BiLSTM	**95.14**	2.76	2.10	0.64	98.18	**93.44**
	BiLSTM	94.49	4.49	**1.02**	0.44	98.21	92.85
	InceptionV3-1D [9]	94.34	3.99	1.67	0.53	**98.89**	93.37
	XGBoost [19]	93.85	**2.37**	3.78	**0.41**	98.34	92.36
	SEDREAMS [3]	91.80	3.03	5.16	0.45	97.37	90.02
	DYPSA [4]	89.43	4.38	6.19	0.54	97.13	86.89
SLT	CNN-BiLSTM	**97.04**	1.76	**1.20**	**0.14**	**99.78**	**96.83**
	BiLSTM	96.82	1.96	1.22	0.15	99.73	96.57
	InceptionV3-1D [9]	96.84	1.36	1.80	0.17	99.73	96.59
	XGBoost [19]	96.05	**0.57**	3.38	0.17	99.71	95.78
	SEDREAMS [3]	94.66	1.13	4.21	0.17	99.67	94.36
	DYPSA [4]	93.25	2.91	3.84	0.32	99.39	92.75
KED	CNN-BiLSTM	**96.64**	1.67	1.69	0.26	99.63	**96.29**
	BiLSTM	96.49	2.61	**0.90**	**0.22**	**99.69**	96.21
	InceptionV3-1D [9]	96.22	2.71	1.08	0.24	99.60	95.87
	XGBoost [19]	95.70	**1.29**	3.02	0.25	99.64	95.37
	SEDREAMS [3]	92.30	6.03	1.66	0.29	99.12	91.76
	DYPSA [4]	90.27	7.07	2.65	0.30	99.25	89.72
TOTAL	CNN-BiLSTM	**96.31**	2.12	1.58	0.41	99.18	**95.54**
	BiLSTM	95.93	2.95	**1.11**	**0.29**	99.18	95.18
	InceptionV3-1D [9]	95.87	2.46	1.68	0.35	**99.41**	95.35
	XGBoost [19]	95.22	**1.30**	3.48	**0.29**	99.21	94.49
	SEDREAMS [3]	93.37	2.34	4.29	0.31	98.79	92.51
	DYPSA [4]	90.27	7.07	2.65	0.30	99.25	89.72

("non-deep") machine learning-based algorithm XGBoost [19] and with two traditional GCI detection methods SEDREAMS [3] and DYPSA [4]. Since SEDREAMS and DYPSA estimate GCIs also during unvoiced segments, their authors recommend filtering the detected GCIs by the output of a separate voiced/unvoiced detector. We applied an F_0 contour estimated by the REAPER algorithm [27] for this purpose. There is no need to apply such post-processing on GCIs detected by the machine learning-based methods since the voiced/unvoiced pattern is used internally in these methods. To obtain consistent results, the detected GCIs were shifted towards the neighboring minimum negative sample in the speech signal.

Two voices, a US male (BDL) and a US female (SLT) from the CMU ARCTIC database [14, 15], were used as a test material. Each voice consists of 1132 phonetically balanced utterances of total duration ≈54 minutes per voice. Additionally, KED TIMIT database [15], comprising 453 phonetically balanced utterances (≈20 min.) of a US male speaker, was also used for testing. All these datasets comprise clean speech. Ground truth GCIs were detected from contemporaneous EGG recordings in the same way as described in Section 2.1 (again shifted towards the neighboring minimum negative sample in the speech signal)[1]. Original speech signals were downsampled to 16 kHz and checked to have the same polarity as described in Section 2.1. It is important to mention that none of the voices from these datasets was part of the training dataset used to train the machine-learning models.

The results in Table 3 confirm that machine learning-based algorithms clearly outperform the traditional ones for all testing datasets. Deep learning approaches (CNN-BiLSTM, BiLSTM, and InceptionV3-1D) tend to perform better than non-deep XGBoost.

As for the comparison of RNN (BiLSTM) and CNN (InceptionV3-1D) GCI detection, the RNN model tend to be better in *reliability*, especially with respect to the identification (IDR) and false alarm (FAR) rates, suggesting that capturing temporal dependencies by a sequence-to-sequence modeling of input speech frames helps in better identification of GCIs.

The joint CNN-RNN (CNN-BiLSTM) architecture further enhances the GCI detection performance and excels in terms of IDR and the combined dynamic evaluation measure (E10). As for the *accuracy*, all three models performed comparably well with InceptionV3-1D being on average the best in terms of the smallest number of timing errors higher than 0.25 ms (A25) and BiLSTM being on average the best in terms of identification accuracy (IDA).

5. Conclusions

In this paper, we showed that framing GCI detection as a sequence-to-sequence prediction problem in which temporal dependencies could be interpreted across a sequence of speech frames leads to better GCI detection, especially with respect to the reliability measures (identification rate, IDR). Adding CNN layers on the front end (thus extracting relevant features from input speech) followed by recurrent layers with a dense layer on the output further improves the GCI detection performance. The proposed CNN-BiLSTM GCI detection model outperforms other machine learning-based models (either deep learning or classical non-deep learning ones) and also clearly outperforms traditional GCI detection algorithms on several public datasets.

The frame-based modeling, which respects temporal structure and dependencies present in speech, outperforms peak-based modeling. It is a good finding because no speech filtering and peak detection is required when processing input speech frame by frame.

[1] The ground truth GCIs and other data relevant to the described experiments are available online [28].

6. References

[1] H. Purwins, B. Li, T. Virtanen, J. Schl, S.-y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[2] T. Drugman, M. Thomas, J. Gudnason, P. Naylor, and T. Dutoit, "Detection of glottal closure instants from speech signals: A quantitative review," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 994–1006, mar 2012.

[3] T. Drugman and T. Dutoit, "Glottal closure and opening instant detection from speech signals," in *INTERSPEECH*, Brighton, Great Britain, 2009, pp. 2891–2894.

[4] P. A. Naylor, A. Kounoudes, J. Gudnason, and M. Brookes, "Estimation of glottal closure instants in voiced speech using the DYPSA algorithm," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 34–43, 2007.

[5] M. Goyal, V. Srivastava, and A. P. Prathosh, "Detection of glottal closure instants from raw speech using convolutional neural networks," in *INTERSPEECH*, Graz, Austria, 2019, pp. 1591–1595.

[6] G. M. Reddy, K. S. Rao, and P. P. Das, "Glottal closure instants detection from speech signal by deep features extracted from raw speech and linear prediction residual," in *INTERSPEECH*, Graz, Austria, 2019, pp. 156–160.

[7] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, "1-D convolutional neural networks for signal processing applications," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Brighton, United Kingdom, 2019, pp. 8360–8363.

[8] L. Ardaillon and A. Roebel, "GCI detection from raw speech using a fully-convolutional network," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Barcelona, Spain, 2020, pp. 6739–6743.

[9] J. Matoušek and D. Tihelka, "A comparison of convolutional neural networks for glottal closure instant detection from raw speech," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Toronto, Canada, 2021, pp. 6938–6942.

[10] M. R. P. Thomas, J. Gudnason, and P. A. Naylor, "Estimation of glottal closing and opening instants in voiced speech using the YAGA algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 82–91, jan 2012.

[11] T. Drugman, P. Alku, A. Alwan, and B. Yegnanarayana, "Glottal source processing: From analysis to applications," *Computer Speech and Language*, vol. 28, no. 5, pp. 1117–1138, 2014.

[12] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, pp. 85–112, 2020.

[13] P. Steiner, I. S. Howard, and P. Birkholz, "Glottal closure instance detection using Echo State Networks," in *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung*, Berlin, Germany, 2021, pp. 161–168.

[14] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 223–224.

[15] "FestVox Speech Synthesis Databases." [Online]. Available: http://festvox.org/dbs/index.html

[16] M. Legát, D. Tihelka, and J. Matoušek, "Pitch marks at peaks or valleys?" in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, vol. 4629, pp. 502–507.

[17] M. Legát, J. Matoušek, and D. Tihelka, "On the detection of pitch marks using a robust multi-phase algorithm," *Speech Communication*, vol. 53, no. 4, pp. 552–566, 2011.

[18] ——, "A robust multi-phase pitch-mark detection algorithm," in *INTERSPEECH*, vol. 1, Antwerp, Belgium, 2007, pp. 1641–1644.

[19] J. Matoušek and D. Tihelka, "Using extreme gradient boosting to detect glottal closure instants in speech signal," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Brighton, United Kingdom, 2019, pp. 6515–6519.

[20] R. Socher, C. Chiung-Yu Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *International Conference on Machine Learning*, Bellevue, Washington, USA, 2011, pp. 129–136.

[21] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1724–1734.

[24] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2015.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, San Diego, USA, 2015.

[26] M. S. Hussain and M. A. Haque, "SwishNet: A fast convolutional neural network for speech, music and noise classification and segmentation," 2018. [Online]. Available: http://arxiv.org/abs/1812.00149

[27] "REAPER: Robust Epoch And Pitch EstimatoR." [Online]. Available: https://github.com/google/REAPER

[28] "Data used for CNN-BiLSTM glottal closure instant detection." [Online]. Available: https://github.com/ARTIC-TTS-experiments/2022-Interspeech