ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9413675

A COMPARISON OF CONVOLUTIONAL NEURAL NETWORKS FOR 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.

ABSTRACT

In this paper, we continue to investigate the use of machine learning for the automatic detection of glottal closure instants (GCIs) from raw speech. We compare several deep one-dimensional convolutional neural network architectures on the same data and show that the InceptionV3 model yields the best results on the test set. On publicly available databases, the proposed 1D InceptionV3 outperforms XGBoost, a non-deep machine learning model, as well as other traditional GCI detection algorithms.

***Index Terms*—** glottal closure instant (GCI), detection, deep learning, convolutional neural network

1. INTRODUCTION

Machine learning is gaining more and more attraction 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]. It is also the case of *glottal closure instant detection*, a traditional signal processing/detection task. 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.

In our previous research [2, 3], we showed that classical ("non-deep") machine learning, and especially the one based on *extreme gradient boosting* (XGBoost), was able to perform very well and consistently outperformed traditionally used algorithms on several test datasets [3]. 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 [4]. Unlike the traditionally used algorithms, which usually exploit expert knowledge and hand-crafted rules and thresholds to identify GCI candidates from local maxima of various speech representations (see, e.g. [5]), the advantage of a machine-learning-based method is that once a training dataset is available and relevant features identified from raw speech, classifier parameters are set up automatically without manual tuning. On the other hand, the identification of relevant features may be time-consuming and tricky, especially when carried out by hand.

Deep learning, and especially *convolutional neural networks* (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. [6, 7]. CNNs were already shown to perform very well in GCI detection [8, 9, 10, 11].

In this paper, we investigate several deep one-dimensional (1D) CNN architectures in the context of GCI detection and compare them with non-deep machine learning XGBoost and with traditional GCI detection algorithms on the same data.

Table 1. Train/validation/test dataset description.

Dataset	Train	Val.	Test	Total
# utterances	3,136	32	32	3,200
length (minutes)	331.28	3.52	3.48	338.28
# peaks	2,127,650	22,644	22,397	2,172,691
# GCIs	1,767,752	18,901	18,687	1,805,340
# non-GCIs	359,898	3,743	3,710	367,351

2. DATA DESCRIPTION

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 [12, 13] (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 [14]. Ground truth GCIs were detected from contemporaneous EGG recordings by the Multi-Phase Algorithm (MPA) [15] and shifted towards the neighboring minimum negative sample in the speech signal. The ratio of division into train/validation/test sets was set to 98/1/1 (see Table 1 for more details). Each voice was part of the train, validation and test dataset.

Since the classification of peaks as GCI/non-GCI is performed in a peak-by-peak manner, 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 [16] (see also Fig. 1). It was also found that downsampling to 8 kHz prior filtering provided slightly better results than the use of 16 kHz. Thus, all compared CNNs use 8 kHz internally.

This research was supported by the Czech Science Foundation (GA CR), project No. GA19-19324S. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

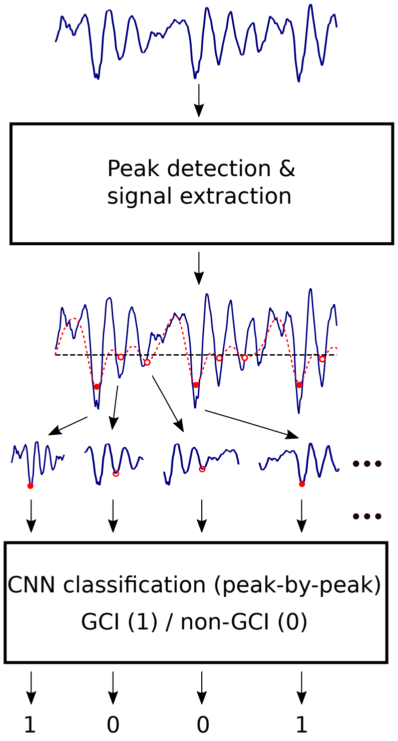

Fig. 1. A simplified scheme of a CNN-based GCI detection from raw speech signal (blue solid line). Negative peaks (either true GCIs ● or non-GCIs ○) are detected from the corresponding low-pass filtered signal (red dashed line).

3. EXPERIMENTS

Unlike classical (non-deep) machine learning algorithms, CNNs have the unique ability to fuse feature extraction and classification into a single learning body, and thus eliminate the need for fixed and hand-crafted features. Typically, each CNN consists of a series of *convolutional layers* (convolving their input with learnable kernels and computing feature maps) interleaved with *pooling layers* (downsampling the learned feature maps), followed by one or more dense layers which perform the actual classification. Conventional (2D) CNNs were originally introduced to perform object recognition/detection tasks for 2D signals (images or video frames), and since then they became the state-of-the-art technique for many computer vision tasks [17, 18].

In this section, we compare several 1D CNN architectures on the same data. For this purpose, we implemented 1D versions of well-known 2D architectures primarily proposed for image processing (LeNet-5 [19], various VGG networks [20], Inception [21], ResNet [22], Xception [23], NASNet [24], ShuffleNet [25]) in the Keras framework [26]. We also included two 1D architectures proposed directly for audio processing: “Yang2018” (a VGG10-like network [8]) and SwishNet [27]. To our knowledge, this is the first such extensive comparison of 1D CNN models.

Due to the unbalanced number of GCIs and non-GCIs in our data (see Table 1), the comparison was made with respect to $F1$, *recall* (R), and *precision* (P) scores.

Table 2. Initial comparison of GCI detection performance using 1D CNNs on the validation set (in percents) including the approximate number of learnable parameters. VGG4/6 are simple VGG networks with only 4/6 layers including only one dense layer with 32 neurons.

CNN	$F1$	R	P	# params
InceptionV3 [28]	98.77	98.86	98.67	12.3M
NASNet-Mobile [24]	98.77	98.81	98.72	4.0M
VGG13 [20]	98.76	98.73	98.79	34.6M
ResNet101 [22]	98.73	98.73	98.74	28.3M
Xception [23]	98.73	98.48	98.97	20.7M
NASNet-Large [24]	98.72	98.77	98.68	83.6M
Inception-ResNetV2 [29]	98.71	98.66	98.76	44.7M
ResNet152 [22]	98.67	98.73	98.62	38.5M
ResNet50 [22]	98.66	98.56	98.77	16.0M
VGG11 [20]	98.66	98.32	99.00	34.5M
VGG16 [20]	98.65	98.37	98.94	36.4M
VGG19 [20]	98.61	98.58	98.64	38.2M
SwishNet-Wide [27]	98.50	98.39	98.61	17k
ShuffleNet [25]	98.49	98.32	98.67	0.5M
SwishNet-Slim [27]	98.49	98.42	98.56	4k
LeNet-5 [19]	98.44	98.28	98.60	0.5M
VGG4 [20]	98.07	97.62	98.52	0.1M
VGG6 [20]	98.02	97.58	98.46	0.1M
Yang2018 [8]	97.55	96.68	98.43	2.9M

Table 3. Optimal hyper-parameter values of the selected models and the best $F1$ score achieved on the validation set.

CNN	FS	W	LR	BS	F1 (%)
Inception-ResNetV2	80	rect.	0.001	64	98.97
NASNet-Mobile	80	rect.	0.001	256	98.91
InceptionV3	80	hamm.	0.001	128	98.90
VGG13	48	rect.	0.001	128	98.88
ResNet101	80	rect.	0.0001	64	98.87
Xception	48	rect.	0.0001	16	98.86

3.1. Initial Comparison

The purpose of the initial comparison of different architectures/models was to identify the capabilities of the models in the context of GCI detection, and to discard some less powerful models from further evaluation.

The architecture of each model (the number of layers, the number of filters and their sizes, the usage of pool layers, batch normalization, etc.) was used the same as proposed by the authors of each model. In all our experiments, the networks were trained to minimize a *binary cross-entropy loss* using *mini-batch gradient descent* with the *Adam optimizer*. *ReLU activations* were applied in all inner layers, whereas a *sigmoid activation* was used in the last (dense) layer. For the initial comparison, the mini-batch size was 128 and the learning rate was 0.001. 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. The length of the input speech segment extracted around each negative peak was fixed to 30 ms (i.e., the frame size was 240 samples).

Based on the results shown in Table 2, the following networks were chosen for further fine-tuning: InceptionV3, NASNet-Mobile, VGG13, ResNet101, Xception, and Inception-ResNetV2.

Table 4. Comparison of the finalized GCI detection models (trained for the given number of epochs (# ep.) on train and validation sets, including the number of learnable parameters # par.) on the test set (left) and the corresponding statistical significance according to McNemar's test [30] (right). The symbols $\gg$ and $>$ mean that the row model is significantly better at the significance level $\alpha = 0.01$ and $\alpha = 0.05$, respectively, than the column model. The symbol $=$ means that the respective models perform the same.

CNN	F1	R	P	# ep.	# par.
InceptionV3 (INC)	98.94	98.94	98.94	8	12.3M
Xception (XCE)	98.85	98.81	98.89	7	20.7M
Inception-ResNetV2 (INR)	98.84	99.15	98.92	9	44.7M
ResNet101 (RSN)	98.84	98.93	98.74	10	28.3M
VGG13 (VG)	98.81	98.94	98.69	17	34.6M
NASNet-Mobile (NNM)	98.78	98.91	98.65	7	4.0M

	INC	XCE	INR	RSN	VGG	NNM
INC	$=$	$>$	$>$	$>$	$>$	$>$
XCE	$<$	$=$	$=$	$=$	$=$	$>$
INR	$<$	$=$	$=$	$=$	$=$	$>$
RSN	$<$	$=$	$=$	$=$	$=$	$>$
VGG	$<$	$=$	$=$	$=$	$=$	$=$
NNM	$<$	$<$	$<$	$<$	$=$	$=$

3.2. Model Tuning

In this stage, we focused on the selected models and tuned their hyper-parameters on the validation set. The following hyper-parameters were taken into account in our comparison: the size of the frame around each negative peak (FS: 30–128 ms, i.e. 240–1024 samples) and windowing (W: rectangular or Hamming), learning rate (LR: 0.0001–0.1), and mini-batch size (BS: 16–512). The optimal hyper-parameter values are shown in Table 3.

3.3. Model Testing

Finally, the tuned models were finalized, i.e., trained on both train and validation datasets for the number of epochs found during the model tuning phase in Section 3.2 (see Table 4), and evaluated on the test set. Due to the stochastic nature of neural network training algorithms (especially when using GPU), we repeated the training five times. A more robust final comparison was then achieved by evaluating the models' performance over all runs.

It could be seen in Table 4 that the 1D InceptionV3 model outperforms all other models at the statistical significance level $\alpha = 0.05$. Other models performed about the same except for 1D NASNet-Mobile which achieved the worst results.

4. COMPARISON WITH OTHER METHODS

To compare the proposed Inception model with different GCI detection algorithms, standard GCI detection measures that concern the *reliability* and *accuracy* of the GCI detection algorithms were used [31]. 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 detection with the identification error $\zeta \leq 0.25$ ms (*accuracy to* ± 0.25 *ms*, A25) and the standard deviation of the identification error ζ (*identification accuracy*, IDA). In addition, we use a more *dynamic evaluation measure* [32]

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

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 ground truth GCIs, N_M is the number of missing GCIs (corresponding to MR), N_{FA} is the number of false GCIs (corresponding to FAR), 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 [32].

4.1. Compared methods

We compared the proposed 1D convolutional network InceptionV3 with a traditional machine learning-based algorithm XGBoost [3] and with six existing state-of-the-art GCI detection methods shown in Table 5. We used the implementations available online; no modifications of the algorithms were made. Since all algorithms (except REAPER) 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 for this purpose. There is no need to apply such post-processing on GCIs detected by InceptionV3-1D and XGBoost since the voiced/unvoiced pattern is used internally in these methods. To obtain consistent results for all methods, the detected GCIs were shifted towards the neighboring minimum negative sample in the speech signal.

4.2. Test datasets

Two voices, a US male (BDL) and a US female (SLT) from the CMU ARCTIC database [12, 13], 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 [13], 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 (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. It is important to mention that none of the voices from these datasets was part of the training dataset used to train InceptionV3-1D and XGBoost models.

4.3. Results

The results in Table 5 show that the Inception network performs very well for all tested datasets[2]. It generally outperforms the baseline non-deep learning XGBoost algorithm and also other non-machine learning algorithms. It excels in terms of *reliability*, especially with respect to the identification (IDR) and false alarm (FAR) rates. In average, the deep 1D CNN architecture based on the Inception V3 network improved GCI detection substantially (of more than 0.6%

[1] The ground truth GCIs and other data relevant to the described experiments are available online [38].

[2] A possible explanation of lower performance metrics (cf. e.g. [5, 31]) is the use of different ground truth GCIs, a different strategy of GCI filtering in unvoiced segments, and perhaps also a different implementation of GCI computation evaluation (also available in [38]).

Table 5. Comparison of GCI detection of the proposed InceptionV3-1D CNN with other algorithms.

Dataset	**Method**	**IDR** (%)	**MR** (%)	**FAR** (%)	**IDA** (ms)	**A25** (%)	**E10** (%)
BDL	InceptionV3-1D	**94.34**	3.99	**1.67**	0.53	98.89	**93.37**
	XGBoost [3]	93.85	**2.37**	3.78	**0.41**	98.34	92.36
	SEDREAMS [33]	91.80	3.03	5.16	0.45	97.37	90.02
	MMF [34]	90.42	4.63	4.95	0.56	97.15	87.87
	DYPSA [31]	89.43	4.38	6.19	0.54	97.13	86.89
	REAPER [35]	93.24	4.39	2.37	0.56	98.01	91.47
	GEFBA [36]	87.93	10.05	2.02	1.02	**99.11**	87.18
	PSFM [37]	87.05	9.65	3.30	0.71	96.95	84.50
SLT	InceptionV3-1D	**96.84**	1.36	**1.80**	**0.17**	99.73	**96.59**
	XGBoost [3]	96.05	**0.57**	3.38	**0.17**	99.71	95.78
	SEDREAMS [33]	94.66	1.13	4.21	**0.17**	99.67	94.36
	MMF [34]	92.44	5.29	2.26	0.40	99.17	91.78
	DYPSA [31]	93.25	2.91	3.84	0.32	99.39	92.75
	REAPER [35]	95.57	1.66	2.77	0.19	99.67	95.27
	GEFBA [36]	94.85	2.62	2.53	**0.17**	**99.76**	94.63
	PSFM [37]	86.95	10.46	2.60	0.45	99.26	86.42
KED	InceptionV3-1D	**96.22**	2.71	1.08	0.24	99.60	**95.87**
	XGBoost [3]	95.70	**1.29**	3.02	0.25	99.64	95.37
	SEDREAMS [33]	92.30	6.03	1.66	0.29	99.12	91.76
	MMF [34]	90.16	7.16	2.68	0.35	98.99	89.52
	DYPSA [31]	90.27	7.07	2.65	0.30	99.25	89.72
	REAPER [35]	91.05	8.18	**0.78**	0.28	99.47	90.67
	GEFBA [36]	88.51	10.36	1.13	**0.21**	**99.74**	88.30
	PSFM [37]	89.47	9.59	0.94	0.39	99.22	88.85
TOTAL	InceptionV3-1D	**95.87**	2.46	**1.68**	0.35	99.41	**95.35**
	XGBoost [3]	95.22	**1.30**	3.48	**0.29**	99.21	94.49
	SEDREAMS [33]	93.37	2.34	4.29	0.31	98.79	92.51
	MMF [34]	91.47	5.25	3.29	0.46	98.41	90.12
	DYPSA [31]	90.27	7.07	2.65	0.30	99.25	89.72
	REAPER [35]	94.25	3.34	2.41	0.37	99.05	93.40
	GEFBA [36]	91.66	6.14	2.20	0.62	**99.53**	91.26
	PSFM [37]	87.25	10.07	2.68	0.55	98.41	85.98

than the XGBoost baseline and even of 2.5% than the well-known SEDREAMS).

As for the *accuracy*, it also performed reasonably well as it often achieved the second-best results (behind the GEFBA algorithm which, however, tends to miss GCIs quite often) in terms of identification accuracy (IDA) and of the smallest number of timing errors higher than 0.25 ms (A25). The proposed InceptionV3-1D also achieved the best results with respect to the combined dynamic evaluation measure (E10).

5. CONCLUSIONS

In this paper, we followed up on our previous work concerning the use of machine learning to detect GCIs from raw speech. We compared several deep one-dimensional CNN architectures on the same data and selected InceptionV3-1D that achieved the best results on the test set ($F1 = 98.94\%$). The InceptionV3-1D model outperforms other traditional state-of-the-art algorithms on several publicly available test datasets.

InceptionV3-1D also outperforms XGBoost, a non-deep machine learning model. It is a good finding because, thanks to its convolutional structure, InceptionV3-1D 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.) which makes the use of classical machine-learning algorithms more difficult.

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] J. Matoušek and D. Tihelka, “Classification-based detection of glottal closure instants from speech signals,” in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3053–3057.

[3] ——, “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.

[4] E. Barnard, R. A. Cole, M. P. Vea, and F. A. Alleva, “Pitch detection with a neural-net classifier,” *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 298–307, 1991.

[5] 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.

[6] 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.

[7] 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.

[8] S. Yang, Z. Wu, B. Shen, and H. Meng, “Detection of glottal closure instants from speech signals: a convolutional neural network based method,” in *INTERSPEECH*, Hyderabad, India, 2018, pp. 317–321.

[9] 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.

[10] 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.

[11] 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.

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

[13] “FestVox Speech Synthesis Databases.” [Online]. Available: http://festvox.org/dbs/index.html

[14] 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.

[15] 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.

[16] J. Matoušek and D. Tihelka, “Glottal closure instant detection from speech signal using voting classifier and recursive feature elimination,” in *INTERSPEECH*, Hyderabad, India, 2018, pp. 2112–2116.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Neural Information Processing Systems*, Lake Tahoe, USA, 2012.

[18] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016.

[23] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 2017.

[24] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018.

[25] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018.

[26] F. Chollet, “Keras,” 2015. [Online]. Available: https://keras.io

[27] 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

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception architecture for computer vision,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *Artificial Intelligence*, San Francisco, USA, 2017.

[30] T. Dietterich, “Approximate statistical tests for comparing supervised classification learning algorithms,” *Neural Computation*, vol. 10, pp. 1895–1923, 1998.

[31] 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.

[32] M. Legát, J. Matoušek, and D. Tihelka, “A robust multi-phase pitch-mark detection algorithm,” in *INTERSPEECH*, vol. 1, Antwerp, Belgium, 2007, pp. 1641–1644.

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

[34] V. Khanagha, K. Daoudi, and H. M. Yahia, “Detection of glottal closure instants based on the microcanonical multiscale formalism,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1941–1950, 2014.

[35] “REAPER: Robust Epoch And Pitch EstimatoR.” [Online]. Available: https://github.com/google/REAPER

[36] A. I. Koutrouvelis, G. P. Kafentzis, N. D. Gaubitch, and R. Heusdens, “A fast method for high-resolution voiced/unvoiced detection and glottal closure/opening instant estimation of speech,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 316–328, 2016.

[37] A. M. V. Rao and P. K. Ghosh, “PSFM – A probabilistic source filter model for noise robust glottal closure instant detection,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 9, pp. 1645–1657, 2018.

[38] “Data used for comparison of CNN-based glottal closure instant detection.” [Online]. Available: https://github.com/ARTIC-TTS-experiments/2021-ICASSP