"HELLO? WHO AM I TALKING TO?" A SHALLOW CNN APPROACH FOR HUMAN VS. BOT SPEECH CLASSIFICATION

A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini, S. Tubaro

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

ABSTRACT

Automatic speech generation algorithms, enhanced by deep learning techniques, enable an increasingly seamless and immediate machine-to-human interaction. As a result, the latest generation of phone-calling bots sounds more convincingly human than previous generations. The application of this technology has a strong social impact in terms of privacy issues (e.g., in customer-care services), fraudulent actions (e.g., social hacking) and erosion of trust (e.g., generation of fake conversation). For these reasons, it is crucial to identify the nature of a speaker, as either a human or a bot. In this paper, we propose a speech classification algorithm based on Convolutional Neural Networks (CNNs), which enables the automatic classification of human vs non-human speakers from the analysis of short audio excerpts. We evaluate the effectiveness of the proposed solution by exploiting a real human speech database populated with audio recordings from various sources, and automatically generated speeches using state-of-the-art text-to-speech generators based on deep learning (e.g., Google WaveNet).

Index Terms— Audio forensics, convolutional neural network, speaker detection

1. INTRODUCTION

The recent Google and Microsoft keynote events had a wide appeal with the Artificial Intelligence (AI) community. The two presented projects, Google Duplex [1] and Microsoft Xiaoice [2] respectively, unveiled the giant step of the two tech companies in reproducing human-to-bot conversations, making incredibly challenging for human ears to distinguish between artificial and natural speeches. In particular, Google Duplex uses WaveNet as a core technology [3], which is a Deep Neural Network (DNN) model to generate raw audio waveforms introduced by the Google AI working group, DeepMind. This Text-To-Speech (TTS) service achieves state-of-the-art performance in terms of speech conversion, and provides great improvements in reproducing natural speech.

On one hand, the advance in human-like generated speech opens new business opportunities for diverse markets [4] (e.g., audiobooks, voice-to-voice translation, etc.). On the other hand, it paves the way for new privacy and security issues to rise. As a matter of fact, the availability of speech generation tools that can be used by any person increases concerns about the authenticity of a speaker. Intruders could pretend to speak on behalf of other people, perfectly reproducing their voices. That would be a danger for social hacking and fake-news generation, especially when paired with realistic video generation tools [5]. Authorities have already sensed that risk and first regulations on bot speech generation have been proposed. For example, in California the so called "Bot bill", SB-1001 [6], would make it unlawful for any person to use a social bot to communicate or interact with people on-line without disclosing that the bot is not a natural person. However, the legislation cannot prevent fraudulent actions. Therefore trusted procedures need to be enforced.

With the deployment of Voice over IP (VoIP) services, telecommunication operators have already faced a similar problem caused by the so called *robocalls*, which initiate unsolicited and undesired communications. Previously, Spam over Telephony Internet (SPIT) detection techniques [7, 8, 9] were used to tackle this problem. Nowadays, these algorithms are not as effective as they used to be, since AI-based services are able to generate conversational speech almost identical to human speech. Therefore, it becomes urgent to develop new solutions to distinguish human from bot audio speeches.

In this paper, we propose a method to identify whether the source of an audio frame from a conversational speech is either a human or a bot, considering the challenging scenario of state-of-the-art Google, Amazon, Microsoft and IBM Text-To-Speech services. Specifically, we leverage a Convolutional Neural Network (CNN) applied to the audio spectrogram (i.e., a 2D representation of the audio signal) computed on short chunks of a recorded audio signal. This approach shows promising results in classifying human and bot speeches. It is also proven to achieve comparable performance when the algorithm is tested on different speech corpora from the ones used for training.

The rest of the paper is structured as follows. In Section 2, we introduce the state-of-the-art on bot recognition. In Section 3, we provide all the details about the proposed detection technique, from audio pre-processing to network definition. In Section 4, we describe all the performed experiments and comment the results achieved. Finally, Section 5 concludes the paper providing some final remarks.

2. RELATED WORK

The classification problem tackled in this paper is a well-known problem of spam detection in telephone communications. The authors in [8] provide a comprehensive survey of the techniques applied by telecommunication providers to block spam communications. In this survey, the Audio-CAPTCHA based techniques are the ones which are more related to our problem. For instance, the

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

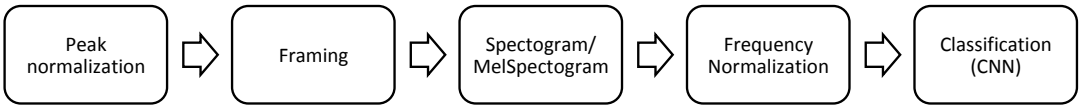


Fig. 1. Pipeline for audio tracks classification. The first four blocks represent the audio pre-processing steps, whereas the last block performs classification by means of a CNN.

works in [7, 9] propose tools based on audio CAPTCHA, where a user starting a SIP session is required to solve an audio test, reporting the sequence of characters that are generated by the authentication server. In particular, the authors in [7] analyze the Short-Term Fourier transform (STFT) of the generated audio and analyze quantitative indicators of the spectrum, such as average amplitude or energy. By tuning properly the contribution of those indicators, they were able to recognize with an accuracy of 97% whether the incoming call was generated by a human or a bot. Another approach was proposed in [10], where a Turing Test is conducted in order to recognize the caller in a telephone conversation. Their study focused on the recognition of specific patterns in human communications, which are not replicated by machine generators. However, the aforementioned approaches account for an excessive delay in the communications and, furthermore, are not tailored to be robust against the novel speech generation services. As a matter of fact, considering that TTS services providing human-like performances in terms of speech fidelity have been only recently proposed [1, 2, 3], to the best of our knowledge no specific techniques targeting their detection have been proposed so far. Nevertheless, given that this new generation of bots can fool human ears, the problem of bot detection can be cast as a speaker recognition problem. A very wide literature focuses on automatic speaker recognition [11]. Most of the classical systems use Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM), which exploit the statistical features of time and frequency components of the audio input, and model the probability distribution over the vector of input features. In the last years, thanks to deep neural networks, the field of speaker recognition has taken a step forward. Some works, like the ones in [12, 13], define CNN architectures in order to identify speaker change detection. The work in [14] applies CNN concepts adapted for an hybrid HMM model.

However, speaker recognition systems are typically tailored to detect biometric properties of specific speakers to be recognized [15]. Conversely, in bot detection problems, we are more interested in recognizing a whole class of speakers (i.e., all bots). Moreover, if changing voice biometric traces is challenging for a human speaker, fine-tuning a TTS algorithm to fool a biometric-based detector is a far less challenging task [16]. For these reasons, in the next section we propose a methodology designed to distinguish humans from bots, assuming that not all bots speeches are available at training time.

3. PROPOSED APPROACH

Given a speech audio track $\mathbf{x}(t)$, our goal is to detect whether it has been recorded from a human speaker or a bot. The solution we propose follows the pipeline depicted in Fig. 1. The audio track under analysis is normalized in terms of amplitude, and a standard windowing technique is used to extract equal length audio frames. Each frame is turned into a 2D spectrogram that is fed to a CNN used for classification purpose. In the following, we detail each step by treating separately signal pre-processing procedures and the proposed CNN.

Signal pre-processing. The audio signal $\mathbf{x}(t)$ is normalized using peak normalization in order to obtain $\bar{\mathbf{x}}(t) = \mathbf{x}(t)/\max(|\mathbf{x}(t)|)$. A series of W frames $\bar{\mathbf{x}}_w(t),\ w \in [1, W]$ is obtained by windowing $\bar{\mathbf{x}}(t)$ with a 50% overlap Hann window of 1 second length (i.e., 16 000 samples in our experiments). Frames with energy below a pre-defined threshold are discarded as simple silence detection technique.

The spectrogram magnitude is computed for each frame $\bar{\mathbf{x}}_w(t)$ in order to obtain a 2D image-like representation. The rationale behind this approach is that 2D CNNs can be powerful instruments to tackle classification problems also in the audio field [17]. In this way the problem of identifying whether a speech is human or bot can be treated as an image classification task, for which many CNN architectures already proved to be promising [18].

Two types of spectrograms are analyzed: the classical spectrogram $\bar{\mathbf{X}}_w(t, f)$, and the mel-frequency spectrogram $\bar{\mathbf{X}}_w(t, m)$. The former provides a time-frequency representation of the signal through Short-Time Fourier Transform (STFT). In this representation, samples are uniformly spaced in the frequency domain. However, it has been observed that the human perception of audio signals is not uniform along the spectrum. Human ear acts as a bank of filters logarithmically distributed in the frequency domain. For this reason, we compute also the mel-frequency spectrogram, that accounts for this behavior. Differently from the classical spectrogram, the mel spectrogram does not provide a uniform frequency representation of the audio signal, but concentrates the components at lower frequencies. The frequency axis is converted in a mel axis, where the mel is defined as [19]

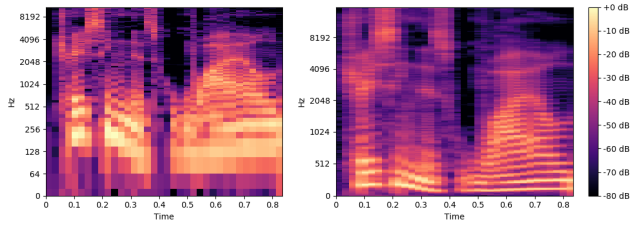
$$m = \begin{cases} f, & f \leq 1\,\text{kHz} \\ 2595 \cdot \log\left(1 + \frac{f}{700}\right), & f > 1\,\text{kHz}. \end{cases} \quad (1)$$

For a better understanding of the difference between the two kinds of spectrograms, Fig. 2 shows both time-frequency representations of a portion of human speech.

Z-score normalization is then applied to spectrogram magnitudes on the frequency/mel axis. The final result is a 2D image of size 256×32 samples as shall be explained in the experimental setup.

CNN architecture. The proposed shallow CNN architecture takes as input the 2D representation of the audio signal $\bar{\mathbf{X}}_w$ (where we drop the time and frequency indexes to denote either normal or mel spectrogram), and outputs a two-element vector indicating the likelihood of the analyzed audio window to belong to each class (i.e., human and bot).

A graphical representation of the used architecture is shown in Fig. 3. The architecture is composed by four 2D convolutional layers, each one followed by a max-pooling layer (2×2 pool size and 2×2 stride). The first and third convolutional layers are composed by 32 and 64 filters with size 4×4, respectively. The second layer



(a) Spectrogram (b) Mel-frequency spectrogram

Fig. 2. Frequency and mel-frequency spectrogram of 1 second of human speech

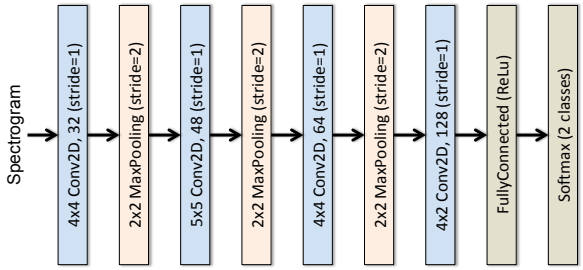


Fig. 3. Shallow convolutional neural network architecture

consists in 48 filters with size 5×5. The last convolutional layer is composed by 128 filters with size 4×2. All the four convolutional layers have stride equal to 1×1. The last convolutional layer is then followed by a fully connected layer with ReLU activation function outputting a 128-element vector. A last fully connected layer with soft-max activation outputs a 2-element vector used to get the final classification result, i.e., human or bot.

The architecture has been designed following the common approach of extracting high-level features from the image (i.e., the spectrogram in our case) through a series of convolutional layers and then classifying the result using fully connected layers [20]. Notice that, due to the different spectrogram resolution in time and frequency domain, the last convolutional layer has been adapted to work on rectangularly shaped inputs. Moreover, the shallow design of the used CNN accommodates for training on smaller datasets.

4. EXPERIMENTS

In this section, we report details regarding the performed experimental campaign tailored at validating the proposed approach.

Dataset. The used dataset is composed by a series of human and bot-generated speech recordings. Lots of human speech datasets are available online. Usually, these datasets are used to perform automatic speech recognition and to train text-to-speech systems. We decide to select at least 2 of them to avoid overfitting on noise or intrinsic characteristics of a single dataset. Specifically we use the *"LibriSpeech ASR corpus"* (LIBRI) [21] and the *"CMU_ARCTIC databases"* (ARTIC) [22]. The former is a large-scale corpus of read English speech generated from audiobooks taken from the LibriVox project. The latter is a dataset built at the Language Technologies Institute at Carnegie Mellon University as phonetically balanced with a single US English female and a single US English male speaker. In the following, the union of the 2 datasets will be referred to as HUMAN dataset. The LIBRI dataset consists of 2 700 tracks, and the ARTIC consists of 2 260 tracks.

Table 1. Test accuracy using HUMAN+3_BOTS for training

Test Bot	Spectrogram Type	Classification accuracy
POLLY	Classic	**98.12%**
	Mel	**95.80%**
AZURE	Classic	48.15%
	Mel	**74.29%**
WATSON	Classic	56.32%
	Mel	**77.56%**

No standard datasets are currently available online for the bot-generated speech, to the best of our knowledge. Therefore we automate the dataset generation process exploiting cloud TTS services APIs that can be exploited to generate speech starting from simple text. Starting from the transcript of the *"LibriSpeech ASR corpus"* dataset, we consider 4 different cloud services with different bots configuration. Namely, we use Google Cloud TTS service with both Standard (G_STD) and WaveNet (G_WAVE) bots[1], Amazon AWS Polly (POLLY)[2], Microsoft Azure TTS (AZURE)[3] and IBM Watson TTS (WATSON)[4]. All selected TTS services allow to specify different types of voice. We exploit this possibility to generate the wildest bot speech dataset as possible. In the following the union of G_STD, G_WAVE and POLLY datasets will be referred as 3_BOTS dataset. Each bot dataset consists of 2 700 audio tracks, except for the WATSON dataset, which is limited to 1 075 (due to service limitations at dataset creation time). Notice that bots read the same sentences used in the human dataset.

All audio files are WAV PCM files, sampled at 16KHz with 16bit per sample and 1 channel (i.e., mono), of average length of 10 seconds. This configuration is shared between all TTS services and the human speech. No conversion, sub-sampling, or channel mixing is performed in order to avoid introducing any further trace that could bias the achieved results.

Experimental setup. The CNN is built using Keras [23] running on top of TensorFlow [24]. To calculate spectrograms we use the Kapre library [25] that allows (mel)spectrogram calculation on-the-fly on-GPU without the need to store them into disks. Kapre allows to integrate the audio signal processing pipeline with Keras.

As loss function we select binary cross-entropy as typically done in classification problems. We use Adam optimization algorithm [26] on batches of 50 audio files split in 1 second frames. We set the number of mels for spectrogram computation to 256, and we use 512 samples per STFT time window. Thus, the input size of the CNN is 256×32[5]. We train the model for up to 100 epochs. Specifically, we start with a learning rate of 0.001, and we decimate it if validation loss does not decreases over 5 epochs. If validation loss does not decrease for more than 10 epochs, training stops. On average, the network reaches convergence after 25 epochs.

In the experiments we are considering different datasets combination. For each combination of dataset, we split it in training, validation and testing set. We keep 30% of each dataset for testing,

[1] https://cloud.google.com/text-to-speech/

[2] https://aws.amazon.com/polly/

[3] https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/

[4] https://www.ibm.com/watson/services/text-to-speech/

[5] Each second is composed by 16 000 samples, with 512 samples per STFT window, it means 32 sample in the time domain of the spectrogram

Table 2. TPR values with different training and testing sets. LIBRI and ARTIC (and their union HUMAN) are the human datasets. Test sets containing bots and humans used also for training are marked in bold.

Train \ Test	LIBRI	ARTIC	POLLY	G_STD	G_WAVE	AZURE	WATSON
LIBRI+POLLY	**92.47%**	81.83%	**95.77%**	26.94%	38.65%	24.10%	34.91%
LIBRI+G_STD	**90.97%**	74.58%	35.78%	**97.23%**	81.90%	57.17%	46.45%
LIBRI+G_WAVE	**90.16%**	77.18%	42.95%	78.71%	**90.16%**	63.69%	66.75%
LIBRI+3_BOTS	**84.82%**	57.21%	**93.67%**			73.56%	70.61%
HUMAN+3_BOTS	**85.21%**		**96.66%**			74.29%	77.56%

whereas the remaining part is further split in 75% for training and 25% for validation. Then, we also test our trained models with human and bot datasets not used for training. This approach allows us to evaluate how the models are able to recognize general features to identify the two classes (human or bot), and not overfitting on the specific human or bot. To avoid asymmetric datasets, we always balanced the number of human and bot samples.

As a final note, results are reported in terms of True Positive Rate (TPR) and accuracy in distinguishing bots from humans considering a single 1 second audio excerpt (i.e., one spectrogram).

Spectrogram choice. Our first experiment aims at evaluating which kind of spectrogram better fits our needs. We therefore train and test both methods using different combination of sources (e.g., training on LIBRI and POLLY and testing on G_WAVE datasets). By using classical spectrogram, we reach high accuracy in the validation phase (greater than 98%) but very poor accuracy when testing on a different bot dataset (around 50%). This implies that with classical spectrogram it is not possible to obtain a general model for our classification problem. On the contrary, when we feed the CNN with the mel spectrogram, the testing accuracy reaches good levels (greater than 70%) even on different bots.

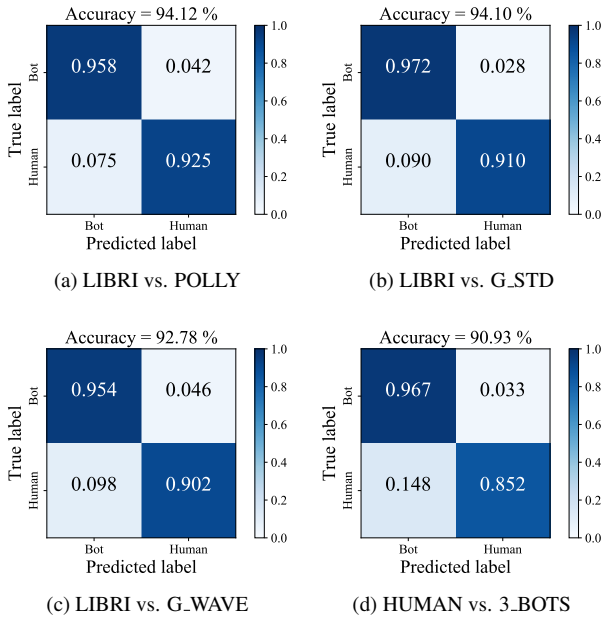
Table 1 reports an example of the aforementioned results when the training set is composed by HUMAN and 3_BOTS, and test is performed on individual bot datasets. When test is evaluated on POLLY, we get comparable accuracy values with classical or mel spectrogram, being POLLY one of the bots using also for training. On the contrary, tests on AZURE and WATSON datasets (not used during training) show different performances, and the mel spectrogram outperforms the classical one. For this reason in the following, only mel spectrogram is used.

Robustness and generalization. To further evaluate the generalization capability of our method, we analyze its performance by training on different combinations of bots and humans datasets.

Fig. 4 shows the confusion matrices obtained using single bots vs. single humans (i.e., Figs. 4a, 4b and 4c) as well as using all bots together (Fig. 4d). In this scenario, bots and humans used for training are also those used for testing. In all these configurations, the proposed solution achieves an accuracy always greater than 89%.

In Table 2 we report numerical results obtained by training and testing on different dataset combinations. In this scenario, we consider for test also bots and humans not used for training. Each row reports the accuracy obtained for a given training set, testing the model with both the corresponding test set (highlighted in bold) and the other datasets of bots and humans not belonging to the initial set.

It is possible to notice that, when the algorithm is trained using a single bot (i.e., the first three rows of Table 2), only the bot used for training is actually correctly recognized as such. The only exception is obtained using LIBRI+G_STD and LIBRI+G_WAVE, which enables recognizing one another. However, since both datasets come from Google Cloud, they might share some characteristics.



(a) LIBRI vs. POLLY (b) LIBRI vs. G_STD

(c) LIBRI vs. G_WAVE (d) HUMAN vs. 3_BOTS

Fig. 4. Confusion matrices on test sets with different datasets

However, if the method is trained on more bots (i.e., the last two rows of Table 2), other bots not belonging to the training set (i.e., AZURE and WATSON) start being correctly recognized with accuracy ranging from 70% to 78%. Even if a small number of bots are used, the proposed CNN architecture is able to generalize and an increasing trend in TPR is shown. As a matter of fact, this is an expected behavior given that the proposed solution is strongly data-driven. The more representative and diversified the training data, the better the results.

5. CONCLUSIONS

In this paper, we proposed a CNN architecture fed by 2D representations of audio signals to classify human and bot speech, in order to deal with the new privacy and security issues due to automatic speech generation tools. Our approach showed promising results (i.e., accuracy greater than 90%) when training and testing sets are matched in terms of used bots. Additionally, if many bots were used for training, the network started generalizing also on bots never seen before (i.e., accuracy greater than 70%).

We hope the achieved results will raise research community awareness about bot detection problem, and motivates it to further investigate this issue in the near future.

6. REFERENCES

[1] Y. Leviathan and Y. Matias, "Google duplex: An ai system for accomplishing real-world tasks over the phone," https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html, May 2018, Last accessed: July 2018.

[2] A. Linn, "Like a phone call: Xiaoice, microsofts social chatbot in china, makes breakthrough in natural conversation," https://blogs.microsoft.com/ai/xiaoice-full-duplex/, April 2018, Last accessed: July 2018.

[3] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[4] Forbes, "Voice Assistants: This Is What The Future Of Technology Looks Like," https://bit.ly/2LzZ9WF, 2017, Last accessed: July 2018.

[5] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: Learning lip sync from audio," *ACM Transactions on Graphics (TOG)*, vol. 36, pp. 95:1–95:13, 2017.

[6] "SB-1001 Bots: disclosure.," https://bit.ly/2OKozlP, 2018, Last accessed: July 2018.

[7] H. Gao, H. Liu, D. Yao, X. Liu, and U. Aickelin, "An audio CAPTCHA to distinguish humans from computers," in *International Symposium on Electronic Commerce and Security*, 2010.

[8] H. Tu, A. Doup, Z. Zhao, and G. J. Ahn, "SoK: Everyone hates robocalls: A survey of techniques against telephone spam," in *IEEE Symposium on Security and Privacy (SP)*, 2016.

[9] D. Gritzalis, Y. Soupionis, V. Katos, I. Psaroudakis, P. Katsaros, and A. Mentis, "The sphinx enigma in critical VoIP infrastructures: Human or botnet?," in *International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2013.

[10] J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, and T. Ewald, "Detecting SPIT calls by checking human communication patterns," in *IEEE International Conference on Communications (ICC)*, 2007.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine (SPM)*, vol. 29, pp. 82–97, 2012.

[12] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *International Conference on Neural Information Processing Systems (NIPS)*, 2009.

[13] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[14] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[15] H. Beigi, *Fundamentals of speaker recognition*, Springer Science & Business Media, 2011.

[16] IBM, "Dynamically configuring the speech to text service or speech to text adapter," https://www.ibm.com/support/knowledgecenter/en/SS4U29/dynamicstt.html, Last accessed: October 2018.

[17] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, "CNN architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[19] D. O'Shaughnessy, *Speech communication: human and machine*, Universities press, 1987.

[20] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[22] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *ISCA Workshop on Speech Synthesis*, 2004.

[23] F. Chollet et al., "Keras," https://keras.io, 2015.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *USENIX conference on Operating Systems Design and Implementation (OSDI)*, 2016.

[25] K. Choi, D. Joo, and J. Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," *CoRR*, vol. abs/1706.05781, 2017.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.