VOICE COMMAND GENERATION USING PROGRESSIVE WAVEGANS

Thomas Wiest
ZD.B Chair of Embedded Intelligence
for Health Care and Wellbeing
University of Augsburg, Germany
thomas.wiest@student.uni-augsburg.de

Nicholas Cummins
ZD.B Chair of Embedded Intelligence
for Health Care and Wellbeing
University of Augsburg, Germany
nicholas.cummins@ieee.org

Alice Baird
ZD.B Chair of Embedded Intelligence
for Health Care and Wellbeing
University of Augsburg, Germany

Simone Hantke
ZD.B Chair of Embedded Intelligence
for Health Care and Wellbeing
University of Augsburg, Germany

Judith Dineley
ZD.B Chair of Embedded Intelligence
for Health Care and Wellbeing
University of Augsburg, Germany

Björn Schuller *
ZD.B Chair of Embedded Intelligence
for Health Care and Wellbeing
University of Augsburg, Germany

November 1, 2021

ABSTRACT

Generative Adversarial Networks (GANs) have become exceedingly popular in a wide range of data-driven research fields, due in part to their success in image generation. Their ability to generate new samples, often from only a small amount of input data, makes them an exciting research tool in areas with limited data resources. One less-explored application of GANs is the synthesis of speech and audio samples. Herein, we propose a set of extensions to the WaveGAN paradigm, a recently proposed approach for sound generation using GANs. The aim of these extensions – preprocessing, Audio-to-Audio generation, skip connections and progressive structures – is to improve the human likeness of synthetic speech samples. Scores from listening tests with 30 volunteers demonstrated a moderate improvement (Cohen's d coefficient of 0.65) in human likeness using the proposed extensions compared to the original WaveGAN approach.

Keywords Generative Adversarial Network · Speech Synthesis · WaveGAN · Progressive Structure · Audio-to-Audio generation

1 Introduction

Today, arguably one of the biggest applications of audio generation is text-to-speech synthesis. From public service announcements, through to chatbots and digital assistants, synthesised voices are becoming omnipresent in everyday life. Tech giants such as Amazon and Google are investing heavily in improving the authenticity – the human likeness [1] – of these voices [2, 3]. Google's WaveNet generative approach is a state-of-the-art example [4].

Generating natural speech is a highly non-trivial task. Predominant approaches in the literature over the years have included *concatenative synthesis* [5], and *statistical parametric speech synthesis* [6]. Concatenative synthesis approaches are based on the concept of unit selection that stitches together small units of pre-recorded waveforms. Such an approach

*Björn Schuller is also with Group on Language, Audio, and Music, Imperial College London, UK

is comparatively simple when compared to other synthesis paradigms [7]. However, the generated speech can lack human likeness due to effects relating to boundary artefacts.

Statistical parametric speech synthesis approaches help avoid issues such as boundary effects by utilising acoustic models such as *Hidden Markov Models* (HMMs) and/or *Deep Neural Networks* (DNN) [7] to generate a smoothed trajectory of speech features for synthesis in a vocoder. However, due to effects such as oversmoothing [6], or inaccurate acoustic models [6], speech generated using such methods is often described as having a muffled and unnatural quality. At worst, the effects can result in generated speech that sounds even less human than that produced using concatenative synthesis [6, 8]. Moreover, such systems tend to have highly complex processing pipelines that require extensive expertise and time to research and develop [7].

Newer approaches like Google's WaveNet [4] have been developed to reduce the complexity of this pipeline by directly generating raw audio samples. WaveNet is a neural network based autoregressive approach which conditions each generated audio sample on a set of previously generated samples. WaveNet also handles longer-term temporal dependencies using a novel dilated causal convolutional structure [4].

Generative approaches based on adversarial networks have also begun to be explored for voice synthesis [9, 10]. These approaches are based on *Generative Adversarial Networks* (GANs) [11] and use the associated zero-sum paradigm to directly generate new audio samples from a noise distribution. Despite the success of GANs in a range of applications [12], speech samples synthesised using WaveGANs [9] are still clearly distinguishable from real human speech [10].

The work presented in this paper extends the original WaveGAN network. Specifically, we explore the benefits of encoder-decoder networks and progressive structures and adding simple processing methods. The autoencoder [13] and progressive [14] inclusions allows us to condition the WaveGAN using other speech samples simplifying the overall learning task, while the core goal of preprocessing is to ensure the maximum amount of speech-only samples are used in training. The presented results, gained using listening tests, indicate that our extensions improve the human-likeness of the generated samples when compared to the original WaveGAN structure.

2 WAVEGANS PRELIMINARIES

In this section, the basic structures for understanding the WaveGAN approach are explained. This includes the basic GAN architecture (cf. Section 2.1), the Wasserstein loss (cf. Section 2.2) function and the original WaveGAN network (cf. Section 2.3).

2.1 Generative Adversarial Networks

The original GAN architecture proposed by Goodfellow et al. [11] is based on a two-player minimax game whose outcome is synthetically generated data samples. A GAN consists of a discriminator, D, and a generator, G. The generator's role is to learn to generate samples with a distribution P_g from a target data distribution P_x using an a priori known distribution P_z to fool the discriminator into believing that instances sampled from the learnt distribution P_g are from the actual real distribution. In most cases, the data sampled from P_z are uniform random noise vectors with a value range from -1 to 1. The discriminator outputs single scalar values from 0 to 1 indicating the probability of a sample belonging to P_g or P_x.

The generator and the discriminator are trained jointly in a minimax fashion formulated as:

$$V(G,D) = \mathbb{E}_{x\sim P_x}[log(D(x))] + \mathbb{E}_{z\sim P_z}[log(1-D(G(z)))]. \tag{1}$$

The associated value function $V(G,D)$ is minimised by the generator and maximised by the discriminator, i. e. , they try to optimise opposing objective functions in a zero-sum game. This set-up ensures that both networks compete against each other during training; the generator generates more and more realistic samples, while the discriminator becomes increasingly accurate at distinguishing the generated samples from the authentic data. For full details on the GAN training process, the interested reader is referred to [11].

2.2 Wasserstein GANs

Training instability is arguably one of the biggest limitations of GANs [15, 16]. This can result in two major problems: *mode collapse* and *gradient vanishing*. In *mode collapse*, the generator produces a very similar set of samples regardless of the input. In this case, the generator only learns a small subspace of the target data distribution. It switches to another small subspace as soon as the discriminator determines the current learnt model and labels all samples as generated; this

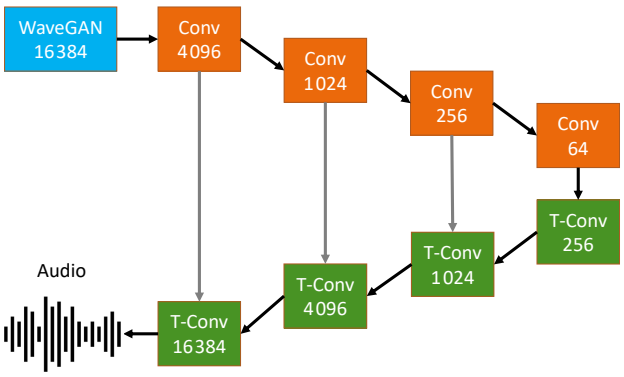

Figure 1: An illustrative example of our Audio-to-Audio generator architecture, enhanced with skip connections (in gray) using a WaveGAN as the input. Also shown are the size of the convolutional (Conv) and transposed convolutional (T-Conv) layers used during our experiments.

process can continue indefinitely. *Gradient vanishing* occurs when the discriminator achieves optimum performance early in training and is able to precisely differentiate between generated and real samples. Consequently, the generator cannot improve its data output any further as the required feedback from the generator is missing.

These issues are addressed by *Wasserstein GANs* [17]. In this paradigm, the GAN learns a function that transforms the existing distribution P_z into P_g instead of directly learning the probability density function P_g that approximates P_x . This is achieved by calculating the *Earth Mover* or *Wasserstein* distance between P_x and P_g. In this paper we use the regularised Wasserstein loss function:

$$\begin{aligned} L = \mathbb{E}_{x\sim P_g}[D(x)] - \mathbb{E}_{y\sim P_r}[D(y)] \\ +\lambda \mathbb{E}_{m\sim P_m}[(||\nabla_m D(m)||_2 - 1)^2], \end{aligned} \tag{2}$$

where m is sampled from x and y and t is uniformly sampled between 0 and 1.

$$m = tx + (1-t)y \tag{3}$$

This loss function ensures we avoid the negative effects of any weight clipping by including the gradient penalty which regulates the equation [18].

2.3 The WaveGAN Architecture

The original WaveGAN network, inspired by the *deep convolutional GAN* (DCGAN) method proposed in [19], was the first attempt using GANs to synthesise raw audio [9]. It is similar to the DCGAN, but is naturally achieved using one-dimensional filters. The network uses a set of *transposed convolution* operations to iteratively upsample and convert a 100-dimensional input vector (uniformly distributed noise values sampled between 0 and 1) into a high-resolution 16 384-dimensional audio file.

The discriminator essentially does the opposite process. It receives the audio files as an input, iteratively downsamples the input throughout its layers, and finally generates a single output value. The convolutional layers have the same kernel size and strides as the generator. Zero padding is used to avoid unwanted effects that otherwise result from shrinking the spatial dimensions. A *phase shuffle* operation is also added between the convolutional layers of the discriminator, as the transposed convolution of the generator produces characteristic artefacts [20] when generating new samples. The operation randomly shifts individual vector entries by a small number to the left or the right. In doing so, it helps prevent the discriminator using the artefacts when distinguishing between the real and the fake samples.

3 Progressive WaveGan Architecture

In this section our proposed improvements to the WaveGAN architecture are described. This includes the preprocessing of audio sample in our datset (cf. Section 3.1), the Audio-to-Audio generation extension (cf. Section 3.2), the inclusion of skip-connections (cf. Section 3.3), and the adoption of a progressive refinement structure (cf. Section 3.4).

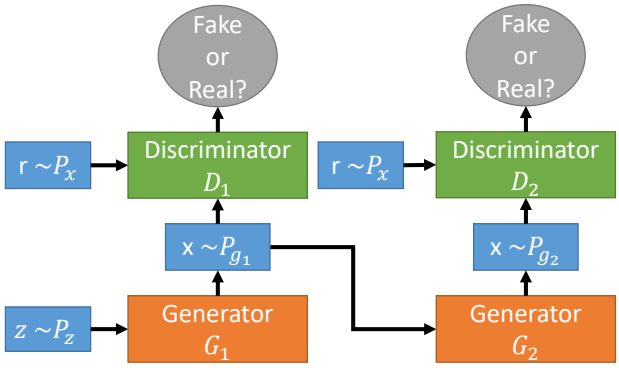

Figure 2: An example of a progressive GAN architecture in which the second GAN structure is conditioned on the generated samples from the first.

3.1 Preprocessing

To improve training speed and data quality, we first do some simple preprocessing on the data. We observed that most samples in the *speech command dataset* ([21], cf. Section 4.1) contain a small period of silence, both before and after the command. To avoid our GAN structure having to operate on this 'silence', we estimate the actual beginning of the spoken word by calculating the short-term energy of the signal and applying a threshold.

3.2 Audio-to-Audio Generation

To improve the quality of the audio output, we adopt a similar technique to the Image-to-Image Translation approach proposed in [13]. Thereby, instead of using a noise vector as the input to the generator, we take a *conditional adversarial* approach and use other speech samples as the input. This approach should make the learning task more manageable as the input is already in a similar structure to the desired output. We apply this technique in a *command independent manner* on the *speech command dataset*, i. e. , we do not condition the generator on the command we wish to generate.

For this step, we adopt an autoencoder architecture (cf. Figure 1), with the input audio being compressed to smaller dimensions using stride convolutions. The output then represents a meaningful input vector which should still include some key structures that originate from speech. This representation then is used to generate new samples via transposed convolutions.

3.3 Skip Connections

To further improve the stability of the autoencoder, we also introduced skip-connections [22] between each convolutional and the corresponding transposed convolutional layer (cf. Figure 1). Skip connections pass data directly from one layer to another to skip the otherwise sequential data flow. These connections make it easier for the gradient to be back-propagated to bottom layers, thus helping to minimise gradient vanishing issues (cf. Section 2.2). Moreover, skip connections have been shown to aid speech enhancement paradigms [23, 24].

3.4 Progressive Refinements Structure

Even with the inclusion of preprocessing, Audio-to-Audio translation and skip connections, we cannot realistically expect our WaveGAN to output 'perfect' speech. Therefore we adapt a progressive refinement scheme, as presented in [14], in which successive GAN structures are concatenated together to improve the overall speech generation quality (cf. Figure 2). This extension was inspired by two main observations. Firstly, this approach has been shown to improve the quality of generated *Positron Emission Tomography* (PET) image, and to the best of the authors' knowledge has not been applied for audio generation. Secondly, autoencoders are commonly used in a variety of audio application for denoising [25, 26].

4 EXPERIMENTAL SETTINGS

This section outlines our key experimental settings.

Table 1: The results of our human listening evaluations. 30 participants listened to 10 files generated using the original WaveGAN network, and 10 files using our proposed approach, and ranked the human-likeness of each file on a 7 point Likert scale (1 not human, 7 human).

Network	Mean Score	Std. Dev	Cohen's d
WaveGAN	3.39	± 1.67	0.65
Proposed Approach	4.48	± 1.70	

4.1 Data

For training the network, a subset of the Google speech command dataset is used [21]. Specifically, we use all spoken digits from zero to nine as an analogy to the widely used MNIST dataset. Our dataset therefore contained 23, 666 samples distributed almost equally across the ten commands. The total length of all commands 6:29:02 (hr:min:sec) at a mean length of 0.986s and a standard devation of 0.057s.

4.2 WaveGAN Parameters

We use a vector consisting of 100 random uniform noise values between 0 and 1 as input for the generator. The input first goes to a fully connected layer with a ReLU (Rectified Linear Unit) activation function. Now, instead of doing two dimensional transposed convolutions with a kernel size of 5x5 and strides of 2x2, we do one dimensional transposed convolutions with a kernel size of 25 and strides of 4. This is repeated until we reach the desired output size of (16 384, 1). On a sample rate of 16 000 Hz, this is slightly longer than a second, and should be enough to model spoken digits. The second dimension is the number of audio channels which will be 1 for all our experiments.

4.3 Audio-to-Audio Parameters

The Audio-to-Audio autoencoder architecture consists of four convolutions with a filter size of 25, a stride of four and a increasing amount of filters. The resulting data will be of size 64x512. This is scaled up again using four transposed convolutions with a decreasing amount of filters until we reach our goal dimension of 16384x1.

4.4 Progressive Parameters

In our experiments, we use two progress layers – the first layer has to be a WaveGAN which takes a noise vector as input. The output of this network is then passed to an autoencoder to further improve the quality. We trained the first layer, the base architecture, for 140 000 iterations and the second layer, the Audio-to-Audio architecture, for 20 000 iterations.

5 RESULTS AND DISCUSSION

The evaluation of results of GANs is an open research topic with no universally agreed upon metric [27]. The widely used *inception score* [28] is not suitable for our evaluation for two reasons. Firstly, it was designed for evaluating images and thus requires the conversion of the wave files into spectrograms which is a lossy operation. Secondly, it does not necessarily correlate with the preferences of humans in case of audio files [9]. We, therefore, use listening tests to evaluate our proposed architecture.

Using the iHEARu-PLAY annotation platform [29], 30 participants evaluated a total of 20 audio samples; ten generated using the original WaveGAN and ten generated using our proposed approach[2]. Each participant was asked to evaluate "how human-like the audio samples sound" on a 7 point Likert scale. In iHEARu-PLAY meta-data can be voluntarily given; 5 out of our 30 participants gave their details. The gender split was 4 females, and 1 male, the age range was $26 - 32$ years old with a mean age of 29 years and a standard deviation of ± 2.28 years.

The results of this evaluation revealed that listens found our samples from our proposed approach to be the more human-like (cf. Table 1). The samples produced by our proposed network achieved a mean rating of 4.48, while those produced by the original WaveGAN network achieved a mean rating of 3.39. To quantify the observed effect we used the Cohen's d measure, calculated to be 0.65. As our d coefficient is between 0.5 and 0.8 we can conclude this a medium effect [30].

[2]Samples available at: `https://goo.gl/sBGkxR`

5.1 Limitations of the Proposed Approach

We observed during system development that both our network and the original WaveGAN approach were not always stable and could produce results of widely varying quality. More specific to our approach, the progressive structure meant our network has a longer training time than the original WaveGAN network. This increased training time limited the amount of hyperparameter optimisation we could realistically achieve. The conditional Audio-to-Audio extension also induced a new style of error. We produced incomprehensible words in approximately 5 % of cases by the mixing up two different commands. This effect was reduced, however, by the introduction of the skip connections. Finally, we faced a commonly occurring issue within GANs; aside from human evaluation, there is no appropriate objective evaluation metric to assess the quality of the generated samples.

6 CONCLUSION AND FUTURE WORK

Within this paper, we proposed a set of extensions to the WaveGAN approach for generating audio samples [9]. These extensions focused on improving the human-likeness of the generated samples. We first prepossessed the audio files to minimise the amount of silence feed into the network. We then modified the WaveGAN structure to be able to perform Audio-to-Audio generation via the inclusion of an autoencoder architecture with skip connections. Finally, we expanded this approach in a progressive structure to help refine the quality of the generated samples

Human evaluations indicated that our approach achieved the goal of improving the human-likeness over the original WaveGAN approach. However, there is still plenty of room for improvement in this regard. In particular, considerably more work is needed to help improve the stability of GANs regarding being able to consistently generating high-quality audio samples. In this regard, we will explore the benefits of using other loss functions to improve the stability of the adversarial training process.

7 ACKNOWLEDGEMENTS

This research has received funding from the EU's 7^{th} Framework Programme ERC Starting Grant No. 338164 (iHEARu), the Bavarian State Ministry of Education, Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B).

References

[1] A. Baird, E. Parada-Cabaleiro, S. Hantke, F. Burkhardt, N. Cummins and B. Schuller. The Perception and Analysis of the Likeability and Human Likeness of Synthesized Speech. In *Proc. INTERSPEECH '18*, Hyderabad, India, 2018, pp. 2863–2867, ISCA.

[2] T. J. Seppala. Amazon's Redesigned Echo Features Improved Sound, Alexa Smarts. `http://engt.co/2ytQ73t`, 2017, Accessed: 28.10.2018.

[3] A. Robertson. Google's DeepMind AI Fakes Some of the Most Realistic Human Voices Yet. `http://bit.ly/2hDhsxf`, 2016, Accessed: 28.10.2018.

[4] 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. In *Proc. 9th ISCA Speech Synthesis Workshop*, Sunnyvale, CA, 2016, pp. 125–125, ISCA.

[5] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database In *Proc. ICASSP '96*, Atlanta, GA, 1996, pp. 373–376, IEEE.

[6] H. Zen, K. Tokuda and A. W. Black. Statistical parametric speech synthesis. *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[7] Z. Ling, S. Kang, H. Zen, A. Senior, M. Schuster, X. Qian, H. M. Meng and L. Deng. Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[8] H. Zen. Acoustic modeling in statistical parametric speech synthesis–from HMM to LSTM-RNN. In *Proc. MLSLP '15*, Fukushima, Japan, 2015, 10 pages, ISCA.

[9] C. Donahue, J. McAuley and M. Puckette. Synthesizing Audio with Generative Adversarial Networks. `https://arxiv.org/abs/1809.10636`, 2018, 8 pages.

[10] C. Y. Lee, A. Toffy, G. J. Jung and W.-J. Han Conditional WaveGANs. https://arxiv.org/abs/1809.10636, 2018, 8 pages.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *NIPS Proc. 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberge Eds, 2014, pp. 2672–2680, Curran Associates, Inc.

[12] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[13] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, . Image-to-Image Translation with Conditional Adversarial Networks. https://arxiv.org/abs/1611.07004, 2016, 17 pages.

[14] Y. Wang, B. Yu, L. Wang, C. Zu, D. S. Lalush, W. Lin, X. Wu, J. Zhou, D. Shen and L. Zhou 3D conditional generative adversarial networks for high-quality PET image estimation at low dose. *NeuroImage*, vol. 174, pp. 550–562, 2018.

[15] P. Manisha and S. Gujar. Generative Adversarial Networks (GANs): What it can generate and What it cannot?. http://arxiv.org/abs/1804.00140, 2018, 14 pages.

[16] D. Bang and H. Shim Improved Training of Generative Adversarial Networks Using Representative Features. http://arxiv.org/abs/1801.09195, 2018, 10 pages.

[17] M. Arjovsky and S. Chintala and L. Bottou Wasserstein Generative Adversarial Networks In *Proc. ICML '17*, D. Precup and Y. W. Teh Eds., Sydney, Australia, 2017, pp. 214–223, PMLR.

[18] I. Gulrajani, F. Ahmed, ,M. Arjovsky, V. Dumoulin and A. C. Courville. Improved Training of Wasserstein GANs. In *NIPS Proc. 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett Eds., 2017, pp. 5767–5777, Curran Associates, Inc.

[19] A. Radford, L. Metz and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. https://arxiv.org/abs/1511.06434, 2015, 16 pages.

[20] A. Odena, V. Dumoulin and C. Olah. Deconvolution and Checkerboard Artifacts. *Distill*, http://distill.pub/2016/deconv-checkerboard, 2016, Accessed: 28.10.2018.

[21] P. Warden Speech Commands Dataset. https://goo.gl/RZmbLH, 2017, Accessed 21.06.2018.

[22] X. Mao, C. Shen, Y.-B. Yang. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections In *NIPS Proc. 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett Eds., 2016, pp. 2802–2810, Curran Associates, Inc.

[23] S. R. Park and J. W. Lee. A Fully Convolutional Neural Network for Speech Enhancement. In *Proc. INTERSPEECH '17*, Stockholm, Sweden, 2017, pp. 1993–1997, ISCA.

[24] S. Pascual and A. Bonafonte and J. Serrà, SEGAN: Speech Enhancement Generative Adversarial Network. In *Proc. INTERSPEECH '17*, Stockholm, Sweden, 2017, pp. 3642–3646, ISCA.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.-A Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion *Journal of Machine Learning Research*, vol 11, no. Dec., pp. 3371–3408, 2010.

[26] F. Weninger, S. Watanabe, Y. Tachioka, B. Schuller Deep Recurrent De-Noising Auto-Encoder and blind de-reverberation for reverberated speech recognition. In *Proc. ICASSP '14*, Florence, Italy, 2014, pp. 4656–4660, IEEE.

[27] J. Han, Z. Zhang, N. Cummins, B. Schuller Adversarial Training in Affective Computing and Sentiment Analysis: Recent Advances and Prospectives. *IEEE Computational Intelligence Magazine, Special Issue on Computational Intelligence for Affective Computing and Sentiment Analysis*, 13 pages, 2018.

[28] S. Barratt and R. Sharma. Note on the Inception Score. https://arxiv.org/abs/1801.01973, 2018, 9 pages.

[29] S. Hantke, F. Eyben, T. Appel and B. Schuller. iHEARu-PLAY: Introducing a game for crowdsourced data collection for affective computing In *Proc. WASA '15* held in conjunction with *ACII '15*, Xi'an, P. R. China, 2015, pp. 891–897, IEEE.

[30] J. Cohen. A power prime. *Psychological bulletin*, vol 112, no. 1, pp. 155–159, 1992.