Interspeech 2018
2-6 September 2018, Hyderabad

Evolving Learning for Analysing Mood-Related Infant Vocalisation

Zixing Zhang[1], *Jing Han*[2], *Kun Qian*[2,3], *Björn Schuller*[1,2]

[1]GLAM – Group on Language, Audio & Music, Imperial College London, UK
[2]ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany
[3]Machine Intelligence & Signal Processing Group, MMK, Technical University of Munich, Germany

zixing.zhang@imperial.ac.uk

Abstract

Infant vocalisation analysis plays an important role in the study of the development of pre-speech capability of infants, while machine-based approaches nowadays emerge with an aim to advance such an analysis. However, conventional machine learning techniques require heavy feature-engineering and refined architecture designing. In this paper, we present an evolving learning framework to automate the design of neural network structures for infant vocalisation analysis. In contrast to manually searching by trial and error, we aim to automate the search process in a given space with less interference. This framework consists of a controller and its child networks, where the child networks are built according to the controller's estimation. When applying the framework to the Interspeech 2018 Computational Paralinguistics (ComParE) Crying Subchallenge, we discover several deep recurrent neural network structures, which are able to deliver competitive results to the best ComParE baseline method.

Index Terms: infant vocalisation, evolving learning, neural network architecture, speech/voice analysis

1. Introduction

Analysing the mood-related infant vocalisations with daylong recordings is of importance in the context of pedology [1, 2, 3, 4, 5, 6]. It can benefit at least two groups: child language development scientists and pediatricians. Several studies have reported that the ratio of linguistic vocalisations increases along with the months of age, whereas non-linguistic vocalisation (e. g., laugh and crying) decrease [7, 4, 6]. Some research also suggests that the different backgrounds on language and culture of families, as well as the frequency of interaction between the infants and their caregivers, have an impact on the development of the pre-speech capability of infants through analysing the infant vocalisation [3, 8]. Besides, the vocalisation analysis highly relates to the healthcare of infants. For example, the children who suffer from autism disorder have fewer emotional vocalisations, and even fewer linguistic vocalisations, than normal ones [5]. The children with severe hearing or voice articulatory impairment have an obvious delay of language acquisition capability [9]. Apart from the groups at risk, analysing the mood-related infant vocalisation also helps for tracking the children's daily state (e. g., comfortability, pain degree, environment sensitivity) variation, and assists the caregivers for their judgement [10, 11, 12]. Despite the necessity for infant vocalisation analysis, the conventional analysis process is quite laborious and time-consuming, since human annotators or even linguistic experts are required to manually track the information of interests from daylong recordings [7]. For this reason, automated analysis systems have nowadays attracted increasing interest in this domain [13]. LENA [14, 13] is one of such a system that is implemented with Gaussian Mixture Models and Hidden Markov Models and enables one to automatically distinguish the linguistic and non-linguistic vocalisations. However, it is a commercial product and can only be used with its corresponding hardware.

With the tremendous success of deep learning technologies in a variety of applications (e. g., computer vision [15, 16], speech recognition [17, 18], and natural language processing [19]), they are considered to be emerging tools as well to deal with the present task. To achieve maximal performance, it is of importance to choose an appropriate network architectures, especially corresponding parameters. However, how to efficiently determine the network architectures or hyper-parameters remains an open challenge [20, 21]. The conventional approach to address this problem involves in a brute-force search. That is, all the possibilities in the parameter space is browsed and implemented in the learning process. Then, these parameters that leads to best performance on the development set determines the final network architecture. Nevertheless, this approach require high expert experiences and computational cost [20, 21, 22].

To lower the usage barrier of deep learning and reduce the computational requirement, some efforts have been made in an automated way to find the optimised parameters [20, 21, 22]. The major advantage of these approaches associates to the fact that it requires no much knowledge from experts and limited computational resources, but still is able to achieve reasonable modelling performance. Among these efforts, Reinforcement Learning (RL) has been introduced and shown appealing empirical results most recently [20, 23]. For instance, in [20], the authors regarded the neural network determination process as a prediction process by using a Recurrent Neural Network (RNN). The prediction process is awarded if it leads to a better result, otherwise, is punished. By repeatedly doing this, it facilitates the decision process to determine the best performed network architecture and parameters.

One may note that, most prior studies have focused on discovering Convolutional Neural Network (CNN) architectures for image classification problems [24, 25, 26]. In the domain of audio analysis, RNN instead has frequently shown to be efficient, since it is capable of capturing long-range context information that is of importance for audio analysis. Automatically predicting the RNN architectures, however, seems to be missing to date. To bridge this gap, we in the first time investigate the feasibility of the RL-based neural network research to design RNN, in an application of an audio-based problem, i. e., infant vocalisation analysis.

10.21437/Interspeech.2018-1914

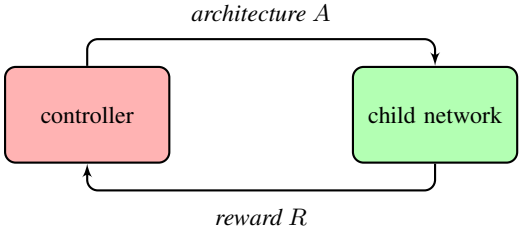

Figure 1: *Overview of the RL-based network search framework, which consists of a controller to predict an architecture A from a search space and a child network with the architecture A to be trained to achieve a reward R. The reward R is then utilised to update the controller.*

2. Evolving Learning

In the following section, we will first describe how to train a RNN as a controller to predict the structure of another RNN. We will then describe how the controller is learnt with a policy gradient method.

2.1. Generating Architectures with Reinforcement Learning

The framework of evolving learning for designing architectures is depicted in Figure 1, which learns from experiences automatically rather than tuning a learner manually by a human expert. In general, it consists of two components which interact with each other in a loop. Formally, one neural network (NN), *the controller*, samples a network architecture A_i from a search space $\{A_i\}$ with $i = 1, ..., I$, where I is the total number of possible network architectures in the search space. Then, the other NN, *the child network* specified by A_i, is constructed and evaluated for a given task. Once the child network with A_i is evaluated, a scalar reward $R(A_i)$ will be provided to the controller as feedback. Hence, in every interaction loop, the controller obtains additional information which can be exploited to update its knowledge. Moreover, the ultimate goal of the controller is to estimate a structure $\hat{A}$ from $\{A_i\}$ to maximise the expected reward. In this respect, instead of searching randomly or via a lattice in $\{A_i\}$, the proposed framework can learn to find an architecture with a high reward within limited interactions, rather than spending a significant amount of time effort and computational resources.

More specifically, in this paper, we aim to find a proper RNN architecture for the task of infant vocalisation classification. Considering that a RNN structure can be represented as a variable-length string, we implement one RNN as the controller to generate such a string, which can be unfolded as illustrated in Figure 2. Inspired by the convolutional network search space of [20] in which one convolutional layer can be represented by five hyper-parameters, we design a simple search space for a RNN with two of the most important hyper-parameters to construct each recurrent layer. Particularly, for each layer indexed by i, the hyper-parameters need to be estimated from a search space containing multiple promising candidates by the controller, i. e., the number of hidden nodes h_i and the activation function ϕ_i. The RNN-based controller, therefore, predicts one parameter (h_i or ϕ_i) and feeds it into the next time step as input to predict the next parameter (ϕ_i or h_{i+1}). This process is repeated multiple times until the number of layers exceeds a predefined value I. As a consequence, the predicted string $\{h_1, \phi_1, ..., h_I, \phi_I\}$ is used to construct a child network, and the child network is trained thereafter on our specific task, i. e., infant vocalisation classification. After training, the validation accuracy of the child network R is recorded and afterwards passed to the controller to update the search algorithm. Hence, given $\theta \in \mathbb{R}^N$ as parameters of the controller, we attempt to update θ in order to maximise the expected reward R.

2.2. Training Controller with Policy Gradient

When applying a controller to design an architecture, each child network architecture τ has a probability to be sampled $p(\tau|\theta)$ which is depended on the controller parameter θ. In other words, to find the optimal architecture, θ should be updated to maximise the reward $J(\theta)$:

$$J(\theta) = \sum_{\tau} R(\tau) p(\tau|\theta), \tag{1}$$

where $R(\tau)$ denotes the accuracy achieved by a designed child network structure τ. In this case, the policy gradient can be approximated using the REINFORCE rule [27] as an estimator of the gradient to update θ, which can be formulated as:

$$\nabla_\theta J(\theta) = \sum_{\tau} R(\tau) p(\tau|\theta) \nabla log p(\tau|\theta). \tag{2}$$

When sampling τ from $p(\tau|\theta)$ N times, i. e., the controller generates N child networks $\{\tau_1, ...\tau_N\}$, an estimate of Eq. (1) can be

$$J(\theta) \approx \frac{1}{N} \sum_{n} R(\tau^n), \tag{3}$$

where $R(\tau^n)$ is the validation accuracy achieved by the n-th sampled architecture. Thus, we reformulate Eq. (2) as:

$$\nabla_\theta J(\theta) = \frac{1}{N} \sum_{n=1}^{N} R(\tau^n) \nabla log p(\tau|\theta). \tag{4}$$

Then, $p(\tau|\theta)$ can be further decomposed into

$$p(\tau|\theta) = p(a_0) \prod_{t=1}^{T} p(a_{t+1}|a_t, \theta), \tag{5}$$

where a_t denotes the predicted component at the time step t and T is the total number of steps needed to design a child network structure. As a result, Eq. (4) can be estimated without knowledge of $p(\tau|\theta)$ as follows:

$$\nabla_\theta J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} R(\tau^n) \nabla log p(a_t|a_{t-1}, \theta). \tag{6}$$

As this computation relies on the empirical return of the sampled child networks, the resulting gradients have a high variance. In this respect, a baseline b is further subtracted from R in order to reduce the variance of the gradient estimator, yielding an extension as follows:

$$\nabla_\theta J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} (R(\tau^n) - b) \nabla log p(a_t|a_{t-1}, \theta). \tag{7}$$

In this work, the baseline b is a moving average of the previous architecture accuracies, to maintain the unbiasedness of the gradient estimate.

3. Experiments and Results

3.1. Database and Features

The INTERSPEECH 2018 ComParE Crying sub-challenge is based on the Cry Recognition In Early Development (CRIED),

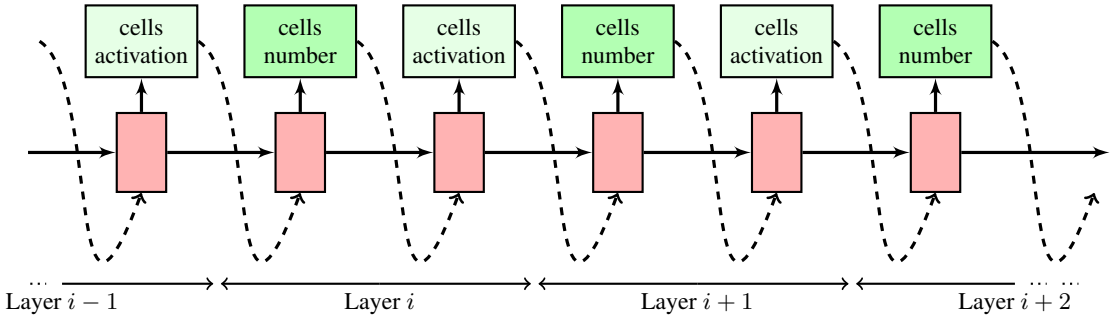

Figure 2: *The RNN controller to generate a RNN child network which allows us to search among all possible structures in a predefined search space. Red blocks symbolise a RNN-based controller unfolded over several time steps, while green blocks indicate the predicted hype-parameters to build a child network. At each step, the controller predicts a single hype-parameter which is then fed back as input to the next time step in an autoregressive fashion.*

Table 1: *Data distribution over different partitions and categories of the CRIED database.*

#	train	test	$\sum$
Neutral/positive	2 292	2 172	4 464
Fussing	368	441	809
Crying	178	136	314
$\sum$	2 838	2 749	5 587

which comprises of 5 587 vocalisations from 20 healthy infants (10 females and 10 males). For the challenge, the data has been split into two partitions: one for training and the other for test. Additionally, the vocalisations of both partitions were categorised into three classes, i. e., neutral/positive mood, fussing, and crying, as shown in Table 1. For a detailed description of the CRIED corpus, the reader is referred to [28]. Note that, in this work, the training partition was further split into three disjoint folds for optimising the controller hyper-parameters, instead of the ten folds as in [28]. This is mainly owing to the fact that, for each controller setting, the evolving searching has to be carried out by exploring more than hundreds of various structures of the child network while each child network has to be trained and evaluated for each fold separately, which is time-consuming. To this end, three-fold cross-validation has been conducted on the training partition in this paper. Furthermore, as illustrated in Table 1, the data is unevenly distributed, with substantially more neutral/positive mood vocalisations. Therefore, we performed upsampling of the (training) data, by replicating vocalisations from the fussing and crying classes proportional to their relative frequency. This results in all three classes having approximately the same number of instances.

To extract acoustic features from the given vocalisations, we employed the INTERSPEECH Computational Paralinguistic Challenges (*ComParE16*) Low-Level Descriptors (LLDs) provided by the challenge. The feature set consists of 65 frame-wise descriptors and their first derivations (delta), resulting in 120 frame-level features in total. The (base) LLDs can be grouped into three parts (or types): energy-related (4), spectral-related (55), and frequency-related (6) ones. In particular, the feature set contains not only commonly used features such as MFCCs and F_0, but also other features of similar and other types, including the spectral flux/variance/skewness and the jitter/shimmer/probability of voicing. More detailed information about the ComParE16 LLDs can be found in [29]. Before utilising these LLDs to validate the performance of a child network structure, an online standardisation was conducted on all LLDs. In detail, for each fold, the global means and variances of these LLDs were calculated on all training data except the selected fold, which were then applied over the selected fold for standardisation accordingly.

3.2. Experimental Setups

In this work, for the implementation of the controller, we kept using the NASCell from TensorFlow [20] for the sake of experiment reproducibility. This NASCell consists of a two-layer Long Short-Term Memory (LSTM)-RNN with 32 hidden units on each layer. The controller was trained with the Adam optimiser with a learning rate of 10^{-4}.

As for the child network, we selected the Gated Recurrent Unit (GRU) as the recurrent hidden unit instead of LSTM. As proposed in [30], GRU can capture the long-term dependencies in sequence-based tasks and can well address the vanishing gradient problem. In addition, when comparing with a LSTM unit, GRU has less parameters to train, which results in a faster training process and less-data demand for generalisation. Besides, many previous studies have shown that the GRU performs competitive to the LSTM unit in most tasks [30, 31].

To search a GRU-RNN architecture for the task at hand, namely the infant vocalisation classification, our evolving space involved various structures which can reach up to 5 layers, as illustrated in Table 2. For each recurrent layer, the controller selects the number of nodes in the range of $[0, 200]$ at the step size of 40 and an activation function in $[tanh, RELU, sigmoid]$. To this end, the evolving space has approximately $6^5 \times 3^5 = 1\,889\,568$ architectures, which is much larger than 200, the number of architectures that the controller was required to evaluate during the experiments. When training these 200 child networks constructed by the algorithms, we employed the Adam optimisation algorithm with an optimised learning rate of 10^{-4}. The batch size was set to 128 to facilitate the training process. After feeding the sequential LLDs into the GRU-RNNs while only one prediction for each vocalisation was obtained by extracting the output after feeding the last frame. This is due to the property of GRU units, as the complete acoustic information over time of the vocalisation was maintained in these units.

Moreover, to evaluate the performance of the child networks, the Unweighted Average Recall (UAR) was employed as suggested by the ComParE challenge [28]. The UAR is calculated by the sum of recalls per class divided by the class number, and thus can well-reflect a meaningful overall accuracy despite class imbalances.

Table 2: *Defined evolving space of the network hyper-parameters via reinforcement learning.*

types	hyper-parameters
# layers	1, 2, 3, 4, 5
# nodes per layer	0, 40, 80, 120, 160, 200
activation functions	1: tanh; 2: ReLU; 3: sigmoid

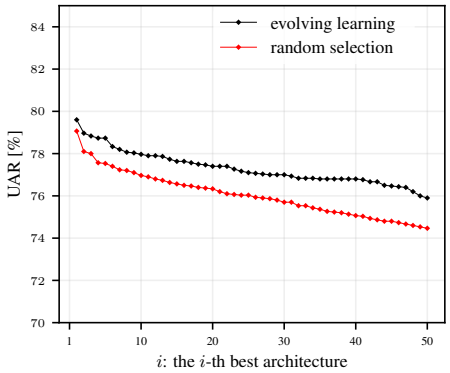

Figure 3: *The obtained UARs by using the i-th best architecture found by the strategies of evolving learning or random selection in 200 successive run times.*

3.3. Results and Discussions

After the controller sampled 200 architectures, we utilised 3-fold cross-validation on the provided training set to evaluate the performance in terms of UAR. As a consequence, we found the architecture that achieved the best UAR of 79.6 % on the training data. This best architecture can be interpreted as a string "*200, sigmoid, 120, sigmoid, 200, sigmoid, 200, tanh, 80, sigmoid*", where the first element (*200*) denotes 200 GRU units in the first layer, the second element (*sigmoid*) denotes that the activation function is sigmoid for this layer, and so forth. In this case, the evolved network has five hidden layers while three of them contain 200 nodes. In other words, the evolving algorithms explored the search space and obtained the best model when reaching the upper-bound of our space settings. This indicates that a deeper and wider architecture may be beneficial for this task. Thus, we expect a more dedicated evolving space can help the model perform better.

In spite of this, we noticed that the second best UAR of 79.0 % was delivered by a three-layer RNN "*200, sigmoid, 200, sigmoid, 160, sigmoid*". This performance is still competitive to the best UAR of 76.9 % of the challenge baselines [28] on the same data partition, and only slightly worse than the best structure. In practice, such a simpler structure can be applied when making a trade-off between the performance and the cost, considering that less weights need to be trained and less data are required for generalisation.

Furthermore, to demonstrate the effectiveness of the evolving learning algorithm, instead of being guided by the controller RNN, we randomly selected 200 architectures from the same space, then trained these architectures and evaluated the performances of them under the same procedures. Note that, as each participant has only up to five trials to upload the results on the test set, hence, these performances were evaluated based on the training data via a three-fold cross validation. The performances of the top 50 networks are reported in Figure 3 for evolving learning and random search, respectively. The plot shows that not only the best model by evolving learning is better than the best model by random selection, but also on average evolving learning is considerably better.

Table 3: *Performance comparison in term of UAR on the test set between the proposed evolved GRU-RNNs and other state-of-the-art approaches.*

approaches	UAR [%]
Seq2Seq [33]	62.1
End-to-End [34]	63.5
BoAW [35]	67.7
ComParE16 [28]	71.9
evolved GRU-RNN	70.1

For the sub-challenge, we uploaded one trial on the test set by utilising the best GRU-RNN model designed via evolving learning. The performances of our approach together with other baseline methods are shown in Table 3. As can be seen in the table, the best model from evolving learning can perform as well as other state-of-the-art approaches on this task. Specifically, the proposed model outperforms the two neural network-based approaches (Seq2Seq and End-to-End) by a large margin, and surpasses the BoAW plus SVM approach, where exactly the same LLDs are exploited. Nevertheless, our model is yet to exceed the SVM trained on the ComParE16 functional feature set, which contains 6 373 static features. This indicates that, the classic ComParE16 set plus SVM is quite compelling for tasks with sparse data, such as the given task; deep learning approaches, per contra, demand big data for good generalisation [32].

4. Conclusions

In this work, we have utilised evolving learning techniques to search recurrent neural network (RNN) architectures that reach good performance for infant vocalisation analysis. In the proposed framework, a controller RNN and a child network are learnt in an interactive loop to attain a best architecture. This learning process is carried out without any human coordination. We then empirically validated the automated designed architectures on the Crying Sub-challenge of INTERSPEECH 2018 ComParE. Our tentative experiments have demonstrated that the search algorithm outperforms random search and promising architectures for this task have been attained.

Encouraged by the achieved results, we will further evaluate our framework for other paralinguistic applications, such as speaker healthiness recognition. In the future, we plan to adjust the reward function by adding a penalty based on the complexity of different architectures. A first step into Automatic Machine Learning (AutoML) for Speech Analysis and Computational Paralinguistics has been taken – many more are to follow, but seem clearly worth it.

5. Acknowledgements

This work was supported by a TransAtlantic Platform "Digging into Data" collaboration grant (ACLEW: Analyzing Child Language Experiences Around The World), with the support of the UK's Economic & Social Research Council through the research Grant No. HJ-253479 (ACLEW), the EU's Horizon 2020 / EFPIA Innovative Medicines Initiative through GA No. 115902 (RADAR-CNS), and the EU's 7th FP through the ERC StG No. 338164 (iHEARu).

6. References

[1] C. Papaeliou, G. Minadakis, and D. Cavouras, "Acoustic patterns of infant vocalizations expressing emotions and communicative functions," *Journal of Speech, Language, and Hearing Research*, vol. 45, no. 2, pp. 311–317, Apr. 2002.

[2] D. K. Oller, E. H. Buder, H. L. Ramsdell, A. S. Warlaumont, L. Chorna, and R. Bakeman, "Functional flexibility of infant vocalization and the emergence of language," *Proceedings of the National Academy of Sciences*, vol. 110, no. 16, pp. 6318–6323, Apr. 2013.

[3] E. H. Buder, A. S. Warlaumont, and D. K. Oller, "An acoustic phonetic catalog of prespeech vocalizations from a developmental perspective," in *Comprehensive Perspectives on Child Speech Development and Disorders: Pathways from Linguistic Theory to Clinical Practice*. New York, NY: Nova Science, 2013, pp. 103–134.

[4] A. S. Warlaumont and H. L. Ramsdell-Hudock, "Detection of total syllables and canonical syllables in infant vocalizations." in *Proc. INTERSPEECH*, San Francisco, CA, 2016, pp. 2676–2680.

[5] G. Esposito, N. Hiroi, and M. L. Scattoni, "Cry, baby, cry: Expression of distress as a biomarker and modulator in autism spectrum disorder," *International Journal of Neuropsychopharmacology*, vol. 20, no. 6, pp. 498–503, Feb. 2017.

[6] C.-C. Lee, Y. Jhang, G. Relyea, L.-m. Chen, and D. K. Oller, "Babbling development as seen in canonical babbling ratios: A naturalistic evaluation of all-day recordings," *Infant Behavior and Development*, vol. 50, pp. 140–153, Feb. 2018.

[7] S. Nathani, D. J. Ertmer, and R. E. Stark, "Assessing vocal development in infants and toddlers," *Clinical Linguistics & Phonetics*, vol. 20, no. 5, pp. 351–369, Jan. 2006.

[8] M. Caskey, B. Stephens, R. Tucker, and B. Vohr, "Importance of parent talk on the development of preterm infant vocalizations," *Pediatrics*, vol. 128, no. 5, pp. 910–916, Nov. 2011.

[9] E. Scheiner, K. Hammerschmidt, U. Jürgens, and P. Zwirner, "The influence of hearing impairment on preverbal emotional vocalizations of infants," *Folia phoniatrica et logopaedica*, vol. 56, no. 1, pp. 27–40, Feb. 2004.

[10] P. Pal, A. N. Iyer, and R. E. Yantorno, "Emotion detection from infant facial expressions and cries," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006, pp. II–721–II–724.

[11] P. Ruvolo and J. Movellan, "Automatic cry detection in early childhood education settings," in *Proc. IEEE Conference on Development and Learning (ICDL)*, Monterey, CA, 2008, pp. 204–208.

[12] R. Cohen and Y. Lavner, "Infant cry analysis and detection," in *Proc. IEEE 27th Convention of Electrical & Electronics Engineers in Israel (IEEEI)*, Tel Aviv, Israel, 2012, pp. 1–5.

[13] J. A. Richards, D. Xu, J. Gilkerson, U. Yapanel, S. Gray, and T. Paul, "Automated assessment of child vocalization development using LENA," *Journal of Speech, Language, and Hearing Research*, vol. 60, no. 7, pp. 2047–2063, July 2017.

[14] D. Xu, U. Yapanel, S. Gray, J. Gilkerson, J. Richards, and J. Hansen, "Signal processing for young child speech language development," in *Proc. 1st Workshop on Child, Computer, and Interaction*, Chania, Greece, 2008, 6 pages.

[15] J. Han, Z. Zhang, N. Cummins, F. Ringeval, and B. Schuller, "Strength modelling for real-world automatic continuous affect recognition from audiovisual signals," *Image and Vision Computing, Special Issue on Multimodal Sentiment Analysis and Mining in the Wild*, vol. 34, pp. 76–86, Sep. 2017.

[16] J. Han, Z. Zhang, M. Schmitt, and B. Schuller, "From hard to soft: Towards more human-like emotion recognition by modelling the perception uncertainty," in *Proc. ACM International Conference on Multimedia (MM)*, Mountain View, CA, 2017, pp. 890–897.

[17] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[18] Z. Zhang, J. Geiger, A. Mousa, J. Pohjalainena, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 5, May 2018, 18 pages.

[19] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th International Conference on Machine Learning (ICML)*, Helsinki, Finland, 2008, pp. 160–167.

[20] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. International Conference on Learning Representations (ICLR)*, Toulon, France, 2017, 16 pages.

[21] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *Proc. International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 2902–2911.

[22] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "SMASH: one-shot model architecture search through hypernetworks," in *Proc. International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018, 21 pages.

[23] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," *arXiv preprint arXiv:1802.03268*, Feb. 2018, 11 pages.

[24] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *Proc. International Conference on Learning Representations (ICLR)*, Toulon, France, Mar. 2017, 18 pages.

[25] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, "Efficient architecture search by network transformation," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, LA, 2018, 8 pages.

[26] Z. Zhong, J. Yan, and C.-L. Liu, "Practical network blocks design with Q-learning," in *arXiv preprint arXiv:1708.05552*, Mar. 2018.

[27] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, May 1992.

[28] B. Schuller, S. Steidl, A. Batliner, P. B. Marschik, H. Baumeister, F. Dong, S. Hantke, F. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, S. Amiriparian, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis, and S. Zafeiriou, "The INTERSPEECH 2018 computational paralinguistics challenge: Atypical & self-assessed affect, crying & heart beats," in *Proc. INTERSPEECH*, Hyderabad, India, 2018, to appear.

[29] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *Proc. INTERSPEECH*, San Francisco, CA, 2016, pp. 2001–2005.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, Dec. 2014.

[31] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. International Conference on Machine Learning (ICML)*, Lille, France, 2015, pp. 2342–2350.

[32] Z. Zhang, N. Cummins, and B. Schuller, "Advanced data exploitation in speech analysis: An overview," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 107–129, July 2017.

[33] Z. Zhang, D. Liu, J. Han, and B. Schuller, "Learning audio sequence representations for acoustic event classification," *arXiv preprint arXiv:1707.08729*, July 2017.

[34] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "End-to-end multimodal emotion recognition using deep neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1301–1309, Dec. 2017.

[35] M. Schmitt and B. Schuller, "openXBOW-Introducing the Passau open-source crossmodal bag-of-words toolkit," *Journal of Machine Learning Research*, vol. 18, no. 96, pp. 1–5, Oct. 2017.