FusionRNN: Shared Neural Parameters for Multi-Channel Distant Speech Recognition

Titouan Parcollet, Xinchi Qiu, Nicholas Lane

INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

FusionRNN: Shared Neural Parameters for Multi-Channel Distant Speech Recognition

Titouan Parcollet[1], *Xinchi Qiu*[1], *Nicholas Lane*[1,2]

[1]University of Oxford, United-Kingdom
[2] Samsung AI, Cambridge, United-Kingdom
titouan.parcollet@cs.ox.ac.uk

Abstract

Distant speech recognition remains a challenging application for modern deep learning based Automatic Speech Recognition (ASR) systems, due to complex recording conditions involving noise and reverberation. Multiple microphones are commonly combined with well-known speech processing techniques to enhance the original signals and thus enhance the speech recognizer performance. These multi-channel follow similar input distributions with respect to the global speech information but also contain an important part of noise. Consequently, the input representation robustness is key to obtaining reasonable recognition rates. In this work, we propose a Fusion Layer (FL) based on shared neural parameters. We use it to produce an expressive embedding of multiple microphone signals, that can easily be combined with any existing ASR pipeline. The proposed model called FusionRNN showed promising results on a multi-channel distant speech recognition task, and consistently outperformed baseline models while maintaining an equal training time.

Index Terms: Multi-channel distant speech recognition, shared neural parameters, light gated recurrent unit neural networks.

1. Introduction

Modern automatic speech recognition (ASR) systems significantly struggle in more realistic distant-talking speech scenarios [1, 2], despite their promising results in close-talking and controlled conditions. Indeed, distant speech recognition is significantly more difficult as it often implies speech signal highly corrupted with noise and reverberation [3, 4].

The use of Multi-microphone arrays is a common approach to enhance distant-talking recognizer performance [5, 6]. The variety of microphones enables the model to receive different and complementary views of the same acoustic event. Consequently, it facilitates the differentiation between noise, reverberation, and the relevant acoustic information. Thus, improving the robustness of the speech recognition system.

Multi-microphone input arrays require the adoption of signal processing techniques aimed at efficiently combining different signals. Traditionally, the beamforming method [7] is commonly used to achieve spatial selectivity. It enables the obtained representation to privilege the spatial areas of the target speaker, and reduces the impact of noise and reverberation. Examples of beamforming are delay-and-sum and filter-and-sum [8, 9]. Most of these techniques propose to realign the different signals in the time domain, and to enhance, increase or filter the energy observed for the relevant speech information by performing specific operations.

More recently, deep multi-microphone signal processing methods have been developed on the back of the current deep learning renaissance [10, 11, 12, 13, 14]. They integrate DSP techniques into the pre-existing neural ASR systems. Such integration of deep learning and traditional signal processing tends to deliver speech pipelines that are more straightforward, transparent, and better performing. For instance, a first paper in this direction has shown that simply concatenating multi-channel features along one dimension and feed a neural acoustic model with it is sufficient to achieve competitive results [15].

Nevertheless, complex inter- and intra- dependencies existing between different signals remain difficult to capture with such simple approaches. Indeed, the different microphones are connected to various independent weights and both inter- and intra- relations are considered at the same level, while we may assume that the input representation is made of M different microphones sharing slightly different but close input distributions. For instance, R, G, B color components characterizing a single pixel may be considered as an analogy to the 3 Mel filter banks energies obtained from 3 different microphones for a single time frame. In this example, a good model must create an expressive and robust latent representation of close but noisy input distributions (*i.e.* the variations between the different microphones) to enable a better understanding of the speech information carried by the signal (*i.e.* what is being spoken). One approach to modelling such dependencies is to inject prior knowledge [12]. In this work, the authors proposed an adaptive neural beamformer with learned filters to perform a delay-and-sum beamforming. Despite good performances, this approach relies on a specific set of beamforming functions and is therefore limited by their definitions. Furthermore, this set of jointly-trained model often increases the complexity of the model, thus prolonging the training time [16].

Our proposal is to consider other inductive biases related to the multi-dimensionality of the data rather than just to its speech representation. First, this proposal draws on the treatment of multidimensional features in deep learning applications including image processing (3D pixels), natural language processing (N-dimensional vectors for a given token), robotic (3D coordinates). In these different contexts, high-dimensional neural networks are applied partly due to the fact that their algebras enable the learning of embeddings that take into account both intra- and inter- dependencies. Examples are complex-valued neural networks [17], quaternion neural networks [18], and models relying on the Clifford algebra [19, 20]. All these architectures have demonstrated promising performances on a wide range of applications due to a specific weight sharing scheme, ensuring natural encodings of the internal relations [21]. Recently, shared weights have been investigated in the context of heterogeneous audio datasets processing [22], showing better out-of-distribution generalisation capabilities than recent transfer learning methods. Siamese neural networks also heavily rely on shared weights [23] to project their input vectors into a shared latent subspace. Finally, a recent related work pro-

http://dx.doi.org/10.21437/Interspeech.2020-2102

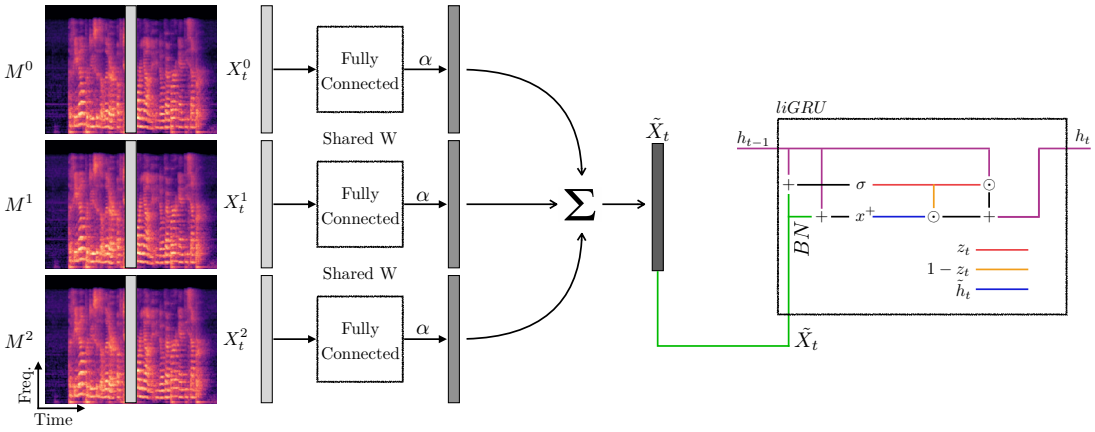

Figure 1: *Illustration of a FusionRNN combining a Fusion Layer (left) and a Light Gated Recurrent Unit RNN (right). A new embedding of the M different microphones is learned through a sum of M non-linear projections with shared parameters of the input signals. This embedding is then used a the new input representation for the neural acoustic model.*

poses to train an encoder across the different microphones and to freeze it to generate embedding features [24].

Our proposal delivers a unified speech pipeline for multi-channel distant ASR referred as FusionRNN. It is composed of a novel Fusion Layer (see section 2) and a Light Gated Recurrent Unit (liGRU) neural network. We hypothesized that our Fusion Layer will produce a more expressive and robust embedding from different microphone signals, and increase the accuracy of the recurrent acoustic model for distant ASR. The experiments conducted on the DIRHA-English [25] dataset support this and highlight consistent improvements in terms of Word Error Rate (WER). We compared the FusionRNN to baseline approaches in the same required training time category. Finally, we release the code under PyTorch-Kaldi [26] to facilitate reproducibility [1].

2. FusionRNN

This Section first motivates and presents the fusion layer (Section 2.1). Then, the FusionRNN is introduced as a combination of a FL and a liGRU network to compose a neural acoustic model (Section 2.2).

2.1. Fusion Layer

Our Fusion Layer (FL) is at the core of the FusionRNN. We expect that if multi-channel signals have close but noisy input distributions then building a common latent representation for these data will help to reduce both the noise and the final transcription error rate. Therefore, we made our FL project each microphone signal to a latent sub-space with a weight matrix that is shared among the different channels (Figure 1). The power of this approach lies in its versatility as it can efficiently encode multi-channel input features for any well-known neural architecture while maintaining an equivalent training time.

Let $x_n^{m,t}$ be the input to a node n coming from the microphone m ($0 \leq m \leq M$) at a time-step t, where M is the total number of microphones ($M \in [1, +\infty[$). The shared weight matrix W is composed of $N \times H$ weight parameters w, with N and H corresponding to the input and hidden vectors sizes respectively. The output $\tilde{x}_h^t$ describing the fusion of the M microphones obtained from the shared weights of the FL layer at the output node h is computed as following:

$$\tilde{x}_h^t = \sum_{i=0}^{M} \alpha(\sum_{j=0}^{N} w_{n,j} x_j^{i,t} + b_h), \quad (1)$$

with α be any non-linear activation function and b_n a bias term.

The introduction of α before the summation over the different signal allows the FL to exhibit non-linear responses with respect to different inputs while sharing the same weight parameters. Indeed, without α, a fusion layer can be reduced to a common fully-connected layer with a specific constraint and three times fewer degrees of freedom.

Note that Eq. 1 can be transformed into a 1D convolutional layer in the context of a precise setup. For instance, we implement Eq. 1 by concatenating the different channels along one dimension and by applying a 1D convolution based on a kernel size and a stride equals to N. Then, the resulting outputs H can be summed to obtain $\tilde{x}_h^t$.

The FL reduces the number of neural parameters of the input layers from $(N \times M) \times H$ for an equivalent fully-connected layer to $N \times H$, while preserving the number of computations. We believe that this reduction will make a speech recognizer equipped with a FL faster to converge and better at generalizing from noisy inputs.

Following prior works that investigate weight sharing architectures such as high-dimensional neural networks [27, 21, 28] or Siamese neural networks [29, 23], our FL is expected to provide an efficient way to capture global variations affecting different microphones, thus increase the robustness of the speech recognizer. In comparison, an embedding obtained with a fully-connected layer may exhibit higher variances across different dimensions of the generated vector, in presence of strong perturbations, which potentially may harm the final performance.

2.2. Integration to Light Gated Recurrent Units

Due to its simple formulation, the fusion layer can easily be integrated to existing neural architectures by simply replacing all the fully-connected layers. In this work, we propose to extend the FL to the light gated recurrent unit (liGRU) first introduced in [30] to compose the acoustic modelling part of an hybdrid DNN-HMM automatic speech recognizer.

[1] https://github.com/mravanelli/pytorch-kaldi/

LiGRUs are a revised version of the well-known GRU recurrent neural networks that account for the specificities of speech recognition. More precisely, liGRU models remove the reset gate and replace the hyperbolic tangent across the hidden state with a rectified linear unit, and a specific input-to-hidden batch-normalisation to stabilize and fasten the training. In practice, liGRU have been shown to always outperform both GRU and LSTM RNNs in different context of speech recognition including distant ASR [30, 26] both in terms of training speed and accuracy. LiGRU equations are summarized as follow:

$$z^t = \sigma(BN(W_z x^t) + U_z h_{t-1}), \quad (2)$$

$$\tilde{h}^t = ReLU(BN(W_h x^t) + U_h h_{t-1}), \quad (3)$$

$$h^t = h^{t-1} \odot z^t + \tilde{h}^t \odot (1 - z^t), \quad (4)$$

with z^t and h^t the update gate and the hidden state at time-step t respectively. The batch-normalisation denoted $BN(x)$ follows the definition given in [31] and normalises the processed mini-batch by considering internal statistics. Biases are integrated to the BN and are therefore omitted from the liGRU equations.

The FusionRNN inference process is obtained by replacing the input layers of z^t and $\tilde{h}$ with fusion layers:

$$z^t = \sigma(BN(FL(x^t)) + U_z h_{t-1}), \quad (5)$$

$$\tilde{h}^t = ReLU(BN(FL(x^t)) + U_h h_{t-1}). \quad (6)$$

$$(7)$$

The hidden state h^t is then updated following the standard liGRU formulation. Finally, the FusionRNN is trained following the backpropagation through time with respect to common cost functions.

3. Experimental Protocol

We propose to evaluate the performance of the FusionRNN on the multi-channel distant speech recognition task of the DIRHA dataset [25] (Section 3.1). Our FusionRNN is compared to equivalent neural networks by varying the number of considered microphones and the initial acoustic features representation (Section 3.2).

3.1. The DIRHA Dataset

Experiments are conducted on the DIRHA-English corpus [32]. This dataset models a domestic environment characterized by the presence of non-stationary noise and acoustic reverberation enabling various benchmarks of speech-based systems in more realistic conditions. 40 Mel filter bank energies and 13 MFCC were computed with windows of 25 ms and an overlap of 10 ms to be used as initial acoustic representations [33]. These two common acoustic features are considered to illustrate the independence of the results with respect to the initial input representation of the signal. Then a delay-and-sum beamforming is applied over the different six microphones to be considered as a baseline.

The training is based on the original Wall-Street-Journal-5k (WSJ) corpus (*i.e.* consisting of 7138 sentences uttered by 83 speakers) contaminated with a set of impulse responses measured in a real apartment [34, 35]. Both a real (Test Real) and a simulated (Test Sim) dataset are used for testing, each consisting of 409 WSJ sentences uttered by six native American speakers. Note that a validation set of 310 WSJ sentences is used for hyper-parameter tuning.

The full circular array of six microphones is considered to evaluate the impact of the number of microphones available on our models.

3.2. Models Architectures

LiGRU and FusionRNN are parametrized following the best model corresponding to the DIRHA recipe proposed in [26]. Models are fed with m microphone signals ($1 \leq m \leq M$) corresponding to a single time frame from each microphone (*i.e.* no right or left context). The bidirectional liGRU layers are composed of 512 neurons and are stacked before being fed to the last softmax-based layer for classification. Then the output labels are the different HMM states of the Kaldi decoder. The fusion layer activation functions are all parametric ReLU (PReLU) [36].

Recurrent weights are initialized orthogonally [37] while input to hidden weights are sampled from a normal distribution following the Glorot criterion [38]. Both FusionRNN and liGRU models are composed of roughly 8M neural parameters and are trained with RMSPROP across 20 epoch with an initial learning rate of $1.6e^{-3}$. The learning rate is halved every time the loss on the validation set increases to ensure an optimal convergence. A dropout rate of 0.2 is applied on all the recurrent layers. Input sequences are chunked to 100 time-steps to warm up the training and doubled at the end of each epoch up to 500. The models are based on the same PyTorch implementation to alleviate the variations that could be observed with different source codes.

4. Results and Discussions

First, the results obtained with different number of microphones on the DIRHA dataset are reported in Table 1. The FusionRNN always outperform equivalent standard liGRU conditioned on the same number of microphones with both the real and simulated test sets. Hence average absolute improvements of 0.7% and 0.6% are obtained with the FusionRNN on the real test set with MFCC and FBANKs features respectively. The same results are observed on the simulated test set with an average gain of 0.9% and 1% based on the same input conditions. This phenomenon highlights the transferability of the fusion layer to different initial acoustic representations. In particular, a best WER of 24.5% is reported on the real test set with a FusionRNN fed with 6 microphones compared to 25.0% for standard liGRU. It is worth underlining that this represents a decrease of 2.9% in WER over prior experiments with DIRHA and non speaker-adapted acoustic features [30].

The gap in transcription error rate between FusionRNN and liGRU increases with the number of microphones as shown in the last column of Table 1. An initial absolute gain of 0.2% in average (*i.e.* with respect to all test sets and features) is observed with 2 microphones, increasing to 1.1% with 5 microphones. This behaviour tends to validate the assumption that weight sharing is helpful in the context of increasing number of acoustic sources by learning an expressive latent representation of the different input distributions.

The introduction of the sixth microphone (LA6) slightly harms the liGRU results while FusionRNN performance is not changed. This may be because the sixth microphone (LA6) is disposed at the centre of the circular array while LA1-LA5 form the circle. This finding is a first step to support the increased robustness offered by the fusion layer.

Table 1: *Results are expressed in terms of Word Error Rate (WER) (i.e lower is better) for different models on the DIRHA dataset with different acoustic features. 'Test Sim.' corresponds to the simulated test set of the corpus, while "Test Real" is the set composed of real recordings. Beam-liGRU is a liGRU fed with delay-and-sum beamformed acoustic features. "Gain" is the absolute average improvement observed with FusionRNN on all test sets and features.*

Models	**Nb. of Mic.**	**Test Real (MFCC)**	**Test Sim. (MFCC)**	**Test Real (FBANK)**	**Test Sim. (FBANK)**	**Gain (%)**
Beam-liGRU	6	27.2	22.0	27.9	21.9	
liGRU [30]	1	27.8	21.3	27.6	21.4	
liGRU	2	26.4	20.1	27.1	21.2	
FusionRNN	2	26.4	20.0	26.8	20.8	-0.2
liGRU	3	26.4	19.3	26.2	20.2	
FusionRNN	3	25.3	18.5	25.7	19.3	-0.9
liGRU	4	26.0	19.1	26.1	20.1	
FusionRNN	4	25.0	18.5	25.5	19.0	-1.0
liGRU	5	25.2	19.8	25.8	20.1	
FusionRNN	5	24.7	18.4	24.9	18.5	-1.1
liGRU	6	25.0	19.9	25.9	19.5	
FusionRNN	6	24.5	18.4	25.1	18.5	-1.0

A crucial benefit of the Fusion Layer lies in the consistency of improvements in terms of WER it deliverers. The proposed FusionRNN always performs better. Furthermore, FusionRNN transfer well to different acoustic representations by achieving superior results with both MFCC and FBANK features. Therefore, it is expected that similar improvements may be expected with other common extraction techniques including fMLLR and PLP that have been shown to be particularly helpful in noisy and reverberated conditions [30].

Second, we investigated the impact of the fusion layer (Eq. 1) on the training time of the FusionRNN compared to a standard liGRU equipped with a plain fully-connected layer. Figure 2 reports the different duration in seconds needed to complete each epoch by the different models. This is to quantify the latency introduced by the FL as described in Section 2.1. We find that the cost in absolute as well as relative terms is marginal. The FusionRNN completes in average one round in 935 seconds compared to 929 for liGRU. This 0.7% difference could be even further reduced with a well-designed PyTorch optimisation of the fusion layer.

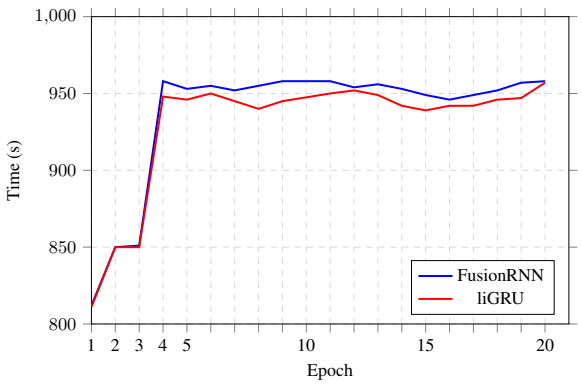

Figure 2: *Training time is seconds recorded for each epoch with both FusionRNN and liGRU ASR systems with 6 input microphones. Models have been trained on a single RTX 2080 Ti.*

Note that the significant increase in duration observed between the first epoch and the fourth one with both models is due to the training strategy. Sequences sizes are gradually increased to warm up the training of the acoustic models. Here, the upper limit is 500 time frames and is reached at epoch number four. Finally, and as shown in the experiments, a fusion layer can easily be embedded in any pre-existing multi-channel neural acoustic model to reduce the number of transcription errors at the only cost of implementation.

5. Conclusion

Summary. This paper first introduced a fusion layer that can easily be plugged into any existing multi-channel ASR systems to learn expressive embeddings from multiple microphone signals while maintaining the training time effectively unaltered. Furthermore, its shared neural parameters allow the FL to project close but different input distributions coming from different microphones in a shared latent subspace, which is more robust to perturbations induced by distant ASR. FL when included in FusionRNN delivered a consistent, material, and robust improvement in transcription error rate over resource-equivalent architectures on a multi-channel distant speech recognition tasks.

Perspectives. Despite few prior works on shared weights, it remains unclear how this mechanism affects the filtering of the acoustic signal in the fusion layer. Therefore, a future work will be to measure the response of the FusionRNN to out-of-distribution speech samples. In this context, we expect that the obtained embedding will offer fewer variance across the different recordings compared to traditional approaches, hence inducing better generalisation capabilities.

6. Acknowledgements

This work was supported by the EPRSC through MOA (EP/S001530/) and Samsung AI. We also would like to thank Filip Svoboda for the numerous and useful comments.

7. References

[1] M. Wölfel and J. W. McDonough, *Distant speech recognition*. Wiley Online Library, 2009.

[2] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition - A Bridge to Practical Applications (1st Edition)*, October 2015.

[3] M. Ravanelli, *Deep learning for Distant Speech Recognition*. PhD Thesis, University of Trento, 2017.

[4] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'chime' speech separation and recognition challenge: An overview of challenge systems and outcomes," 12 2013, pp. 162–167.

[5] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2013.

[6] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[7] W. Kellermann, "Beamforming for speech and audio signals," in *Handbook of signal processing in acoustics*. Springer, 2008, pp. 691–702.

[8] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[9] M. Kajala and M. Hamalainen, "Filter-and-sum beamformer with adjustable filter characteristics," in *Proc. of ICASSP*, 2001, pp. 2917–2920.

[10] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel asr system," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5325–5329.

[11] S. Braun, D. Neil, J. Anumula, E. Ceolini, and S.-C. Liu, "Multi-channel attention for end-to-end speech recognition," *2018 Interspeech*, pp. 0–0, 2018.

[12] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," 2016.

[13] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1274–1288, 2017.

[14] S. Kim and I. Lane, "End-to-end speech recognition with auditory attention for multi-microphone distance speech recognition," in *Proc. Interspeech 2017*, 2017.

[15] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *Proc. of ICASSP*, 2014, pp. 5542–5546.

[16] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1274–1288, 2017.

[17] A. Hirose, *Complex-valued neural networks: theories and applications*. World Scientific, 2003, vol. 5.

[18] T. Parcollet, M. Morchid, and G. Linarès, "A survey of quaternion neural networks," *Artificial Intelligence Review*, pp. 1–26, 2019.

[19] S. Buchholz and G. Sommer, "On clifford neurons and clifford multi-layer perceptrons," *Neural Networks*, vol. 21, no. 7, pp. 925–935, 2008.

[20] Y. Liu, P. Xu, J. Lu, and J. Liang, "Global stability of clifford-valued recurrent neural networks with time delays," *Nonlinear Dynamics*, vol. 84, no. 2, pp. 767–777, 2016.

[21] T. Parcollet, M. Morchid, and G. Linarès, "Quaternion convolutional neural networks for heterogeneous image processing," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8514–8518.

[22] J. S. Larsen and L. Clemmensen, "Weight sharing and deep learning for spectral data," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4227–4231.

[23] A. Gresse, M. Quillot, R. Dufour, V. Labatut, and J.-F. Bonastre, "Similarity metric based on siamese neural networks for voice casting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6585–6589.

[24] R. Li, G. Sell, X. Wang, S. Watanabe, and H. Hermansky, "A practical two-stage training strategy for multi-stream end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7014–7018.

[25] L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmüller, and P. Maragos, "The dirha simulated corpus." in *LREC*, 2014, pp. 2629–2634.

[26] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6465–6469.

[27] T. Isokawa, T. Kusakabe, N. Matsui, and F. Peper, "Quaternion neural network and its application," in *International conference on knowledge-based and intelligent information and engineering systems*. Springer, 2003, pp. 318–324.

[28] M. Morchid, G. Linares, M. El-Beze, and R. De Mori, "Theme identification in telephone service conversations using quaternions of speech features." in *INTERSPEECH*, 2013, pp. 1394–1398.

[29] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[30] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[32] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, "The dirha-english corpus and related tasks for distant-speech recognition in domestic environments," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 275–282.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[34] M. Ravanelli, P. Svaizer, and M. Omologo, "Realistic multi-microphone data simulation for distant speech recognition," *arXiv preprint arXiv:1711.09470*, 2017.

[35] M. Ravanelli and M. Omologo, "Contaminated speech training methods for robust dnn-hmm distant speech recognition," *arXiv preprint arXiv:1710.03538*, 2017.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[37] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 2013.

[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.