SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le*

Google Brain

{danielspark, williamchan, ngyuzh, chungchengc, barretzoph, cubuk, qvl}@google.com

Abstract

We present SpecAugment, a simple data augmentation method for speech recognition. SpecAugment is applied directly to the feature inputs of a neural network (i.e., filter bank coefficients). The augmentation policy consists of warping the features, masking blocks of frequency channels, and masking blocks of time steps. We apply SpecAugment on Listen, Attend and Spell networks for end-to-end speech recognition tasks. We achieve state-of-the-art performance on the LibriSpeech 960h and Swichboard 300h tasks, outperforming all prior work. On LibriSpeech, we achieve 6.8% WER on test-other without the use of a language model, and 5.8% WER with shallow fusion with a language model. This compares to the previous state-of-the-art hybrid system of 7.5% WER. For Switchboard, we achieve 7.2%/14.6% on the Switchboard/CallHome portion of the Hub5'00 test set without the use of a language model, and 6.8%/14.1% with shallow fusion, which compares to the previous state-of-the-art hybrid system at 8.3%/17.3% WER.

Index Terms: end-to-end speech recognition, data augmentation

1. Introduction

Deep Learning has been applied successfully to Automatic Speech Recognition (ASR) [1], where the main focus of research has been designing better network architectures, for example, DNNs [2], CNNs [3], RNNs [4] and end-to-end models [5, 6, 7]. However, these models tend to overfit easily and require large amounts of training data [8].

Data augmentation has been proposed as a method to generate additional training data for ASR. For example, in [9, 10], artificial data was augmented for low resource speech recognition tasks. Vocal Tract Length Normalization has been adapted for data augmentation in [11]. Noisy audio has been synthesised via superimposing clean audio with a noisy audio signal in [12]. Speed perturbation has been applied on raw audio for LVSCR tasks in [13]. The use of an acoustic room simulator has been explored in [14]. Data augmentation for keyword spotting has been studied in [15, 16]. Feature drop-outs have been employed for training multi-stream ASR systems [17]. More generally, learned augmentation techniques have explored different sequences of augmentation transformations that have achieved state-of-the-art performance in the image domain [18].

Inspired by the recent success of augmentation in the speech and vision domains, we propose SpecAugment, an augmentation method that operates on the log mel spectrogram of the input audio, rather than the raw audio itself. This method is simple and computationally cheap to apply, as it directly acts on the log mel spectrogram as if it were an image, and does not require any additional data. We are thus able to apply SpecAugment online during training. SpecAugment consists of three kinds of deformations of the log mel spectrogram. The first is time warping, a deformation of the time-series in the time direction. The other two augmentations, inspired by "Cutout", proposed in computer vision [19], are time and frequency masking, where we mask a block of consecutive time steps or mel frequency channels.

This approach while rudimentary, is remarkably effective and allows us to train end-to-end ASR networks, called Listen Attend and Spell (LAS) [6], to surpass more complicated hybrid systems, and achieve state-of-the-art results even without the use of Language Models (LMs). On LibriSpeech [20], we achieve 2.8% Word Error Rate (WER) on the test-clean set and 6.8% WER on the test-other set, without the use of an LM. Upon shallow fusion [21] with an LM trained on the LibriSpeech LM corpus, we are able to better our performance (2.5% WER on test-clean and 5.8% WER on test-other), improving the current state of the art on test-other by 22% relatively. On Switchboard 300h (LDC97S62) [22], we obtain 7.2% WER on the Switchboard portion of the Hub5'00 (LDC2002S09, LDC2003T02) test set, and 14.6% on the CallHome portion, without using an LM. Upon shallow fusion with an LM trained on the combined transcript of the Switchboard and Fisher (LDC200{4,5}T19) [23] corpora, we obtain 6.8%/14.1% WER on the Switchboard/Callhome portion.

2. Augmentation Policy

We aim to construct an augmentation policy that acts on the log mel spectrogram directly, which helps the network learn useful features. Motivated by the goal that these features should be robust to deformations in the time direction, partial loss of frequency information and partial loss of small segments of speech, we have chosen the following deformations to make up a policy:

1. Time warping is applied via the function `sparse_image_warp` of `tensorflow`. Given a log mel spectrogram with τ time steps, we view it as an image where the time axis is horizontal and the frequency axis is vertical. A random point along the horizontal line passing through the center of the image within the time steps $(W, \tau - W)$ is to be warped either to the left or right by a distance w chosen from a uniform distribution from 0 to the time warp parameter W along that line. We fix six anchor points on the boundary—the four corners and the mid-points of the vertical edges.

2. Frequency masking is applied so that f consecutive mel frequency channels $[f_0, f_0 + f)$ are masked, where f is first chosen from a uniform distribution from 0 to the

*Work done as a member of the Google AI Residency Program.

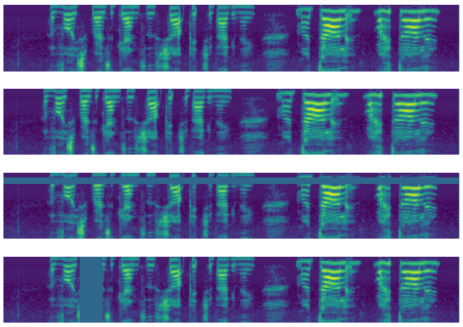

Figure 1: *Augmentations applied to the base input, given at the top. From top to bottom, the figures depict the log mel spectrogram of the base input with no augmentation, time warp, frequency masking and time masking applied.*

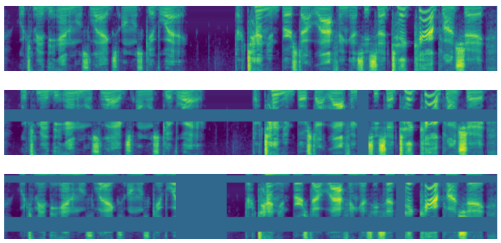

Figure 2: *Augmentation policies applied to the base input. From top to bottom, the figures depict the log mel spectrogram of the base input with policies None, LB and LD applied.*

frequency mask parameter F, and f_0 is chosen from $[0, \nu - f)$. ν is the number of mel frequency channels.

3. Time masking is applied so that t consecutive time steps $[t_0, t_0 + t)$ are masked, where t is first chosen from a uniform distribution from 0 to the time mask parameter T, and t_0 is chosen from $[0, \tau - t)$.

    - We introduce an upper bound on the time mask so that a time mask cannot be wider than p times the number of time steps.

Figure 1 shows examples of the individual augmentations applied to a single input. The log mel spectrograms are normalized to have zero mean value, and thus setting the masked value to zero is equivalent to setting it to the mean value.

We can consider policies where multiple frequency and time masks are applied. The multiple masks may overlap. In this work, we mainly consider a series of hand-crafted policies, LibriSpeech basic (LB), LibriSpeech double (LD), Switchboard mild (SM) and Switchboard strong (SS) whose parameters are summarized in Table 1. In Figure 2, we show an example of a log mel spectrogram augmented with policies LB and LD.

Table 1: *Augmentation parameters for policies. m_F and m_T denote the number of frequency and time masks applied.*

Policy	W	F	m_F	T	p	m_T
None	0	0	-	0	-	-
LB	80	27	1	100	1.0	1
LD	80	27	2	100	1.0	2
SM	40	15	2	70	0.2	2
SS	40	27	2	70	0.2	2

3. Model

We use Listen, Attend and Spell (LAS) networks [6] for our ASR tasks. These models, being end-to-end, are simple to train and have the added benefit of having well-documented benchmarks [24, 25] that we are able to build upon to get our results. In this section, we review LAS networks and introduce some notation to parameterize them. We also introduce the learning rate schedules we use to train the networks, as they turn out to be an important factor in determining performance. We end with reviewing shallow fusion [21], which we have used to incorporate language models for further gains in performance.

3.1. LAS Network Architectures

We use Listen, Attend and Spell (LAS) networks [6] for end-to-end ASR studied in [25], for which we use the notation LAS-d-w. The input log mel spectrogram is passed in to a 2-layer Convolutional Neural Network (CNN) with max-pooling and stride of 2. The output of the CNN is passes through an encoder consisting of d stacked bi-directional LSTMs with cell size w to yield a series of attention vectors. The attention vectors are fed into a 2-layer RNN decoder of cell dimension w, which yields the tokens for the transcript. The text is tokenized using a Word Piece Model (WPM) [26] of vocabulary size 16k for LibriSpeech and 1k for Switchboard. The WPM for LibriSpeech 960h is constructed using the training set transcripts. For the Switchboard 300h task, transcripts from the training set are combined with those of the Fisher corpus to construct the WPM. The final transcripts are obtained by a beam search with beam size 8. For comparison with [25], we note that their "large model" in our notation is LAS-4-1024.

3.2. Learning Rate Schedules

The learning rate schedule turns out to be an important factor in determining the performance of ASR networks, especially so when augmentation is present. Here, we introduce training schedules that serve two purposes. First, we use these schedules to verify that a longer schedule improves the final performance of the network, even more so with augmentation (Table 2). Second, based on this, we introduce very long schedules that are used to maximize the performance of the networks.

We use a learning rate schedule in which we ramp-up, hold, then exponentially decay the learning rate until it reaches $1/100$ of its maximum value. The learning rate is kept constant beyond this point. This schedule is parameterized by three time stamps (s_r, s_i, s_f) – the step s_r where the ramp-up (from zero learning rate) is complete, the step s_i where exponential decay starts, and the step s_f where the exponential decay stops.

There are two more factors that introduce time scales in our experiment. First, we turn on a variational weight noise [27] of standard deviation 0.075 at step s_{noise} and keep it constant throughout training. Weight noise is introduced in the step interval (s_r, s_i), i.e., during the high plateau of the learning rate.

Second, we introduce uniform label smoothing [28] with uncertainty 0.1, i.e., the correct class label is assigned the confidence 0.9, while the confidence of the other labels are increased accordingly. As is commented on again later on, label smoothing can destabilize training for smaller learning rates, and we sometimes choose to turn it on only at the beginning of training and off when the learning rate starts to decay.

The two basic schedules we use, are given as the following:

1. B(asic): $(s_r, s_{\text{noise}}, s_i, s_f)$ = (0.5k, 10k, 20k, 80k)
2. D(ouble): $(s_r, s_{\text{noise}}, s_i, s_f)$ = (1k, 20k, 40k, 160k)

As discussed further in section 5, we can improve the performance of the trained network by using a longer schedule. We thus introduce the following schedule:

3. L(ong): $(s_r, s_{\text{noise}}, s_i, s_f)$ = (1k, 20k, 140k, 320k)

which we use to train the largest model to improve performance. When using schedule L, label smoothing with uncertainty 0.1 is introduced for time steps $< s_i = 140$k for LibriSpeech 960h, and is subsequently turned off. For Switchboard 300h, label smoothing is turned on throughout training.

3.3. Shallow Fusion with Language Models

While we are able to get state-of-the-art results with augmentation, we can get further improvements by using a language model. We thus incorporate an RNN language model by shallow fusion for both tasks. In shallow fusion, the "next token" $\mathbf{y}^*$ in the decoding process is determined by

$$\mathbf{y}^* = \underset{\mathbf{y}}{\text{argmax}} \left(\log P(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y}) \right) , \qquad (1)$$

i.e., by jointly scoring the token using the base ASR model and the language model. We also use a coverage penalty c [29].

For LibriSpeech, we use a two-layer RNN with embedding dimension 1024 used in [25] for the LM, which is trained on the LibriSpeech LM corpus. We use identical fusion parameters ($\lambda = 0.35$ and $c = 0.05$) used in [25] throughout.

For Switchboard, we use a two-layer RNN with embedding dimension 256, which is trained on the combined transcripts of the Fisher and Switchboard datasets. We find the fusion parameters via grid search by measuring performance on RT-03 (LDC2007S10). We discuss the fusion parameters used in individual experiments in section 4.2.

4. Experiments

In this section, we describe our experiments on LibriSpeech and Switchboard with SpecAugment. We report state-of-the-art results that out-perform heavily engineered hybrid systems.

4.1. LibriSpeech 960h

For LibriSpeech, we use the same setup as [25], where we use 80-dimensional filter banks with delta and delta-delta acceleration, and a 16k word piece model [26].

The three networks LAS-4-1024, LAS-6-1024 and LAS-6-1280 are trained on LibriSpeech 960h with a combination of augmentation policies (None, LB, LD) and training schedules (B/D). Label smoothing was not applied in these experiments. The experiments were run with peak learning rate of 0.001 and batch size of 512, on 32 Google Cloud TPU chips for 7 days. Other than the augmentation policies and learning rate schedules, all other hyperparameters were fixed, and no additional tuning was applied. We report test set numbers validated by the dev-other set in Table 2. We see that augmentation consistently improves the performance of the trained network, and that the benefit of a larger network and a longer learning rate schedule is more apparent with harsher augmentation.

We take the largest network, LAS-6-1280, and use schedule L (with training time $\sim$ 24 days) and policy LD to train the network to maximize performance. We turn label smoothing on for time steps $<$ 140k as noted before. The test set performance is reported by evaluating the checkpoint with best dev-other performance. State of the art performance is achieved by the LAS-6-1280 model, even without a language model. We can incorporate an LM using shallow fusion to further improve performance. The results are presented in Table 3.

Table 2: *LibriSpeech test WER (%) evaluated for varying networks, schedules and policies. First row from [25].*

Network	Sch	Pol	No LM		With LM	
			clean	other	clean	other
LAS-4-1024 [25]	B	-	4.7	13.4	3.6	10.3
LAS-4-1024	B	LB	3.7	10.0	3.4	8.3
	B	LD	3.6	9.2	2.8	7.5
	D	-	4.4	13.3	3.5	10.4
	D	LB	3.4	9.2	2.7	7.3
	D	LD	3.4	8.3	2.8	6.8
LAS-6-1024	D	-	4.5	13.1	3.6	10.3
	D	LB	3.4	8.6	2.6	6.7
	D	LD	3.2	8.0	2.6	6.5
LAS-6-1280	D	-	4.3	12.9	3.5	10.5
	D	LB	3.4	8.7	2.8	7.1
	D	LD	3.2	7.7	2.7	6.5

Table 3: *LibriSpeech 960h WERs (%).*

Method	No LM		With LM	
	clean	other	clean	other
HMM				
Panayotov et al., (2015) [20]			5.51	13.97
Povey et al., (2016) [30]			4.28	
Han et al., (2017) [31]			3.51	8.58
Yang et al. (2018) [32]			2.97	7.50
CTC/ASG				
Collobert et al., (2016) [33]	7.2			
Liptchinsky et al., (2017) [34]	6.7	20.8	4.8	14.5
Zhou et al., (2018) [35]			5.42	14.70
Zeghidour et al., (2018) [36]			3.44	11.24
Li et al., (2019) [37]	3.86	11.95	2.95	8.79
LAS				
Zeyer et al., (2018) [24]	4.87	15.39	3.82	12.76
Zeyer et al., (2018) [38]	4.70	15.20		
Irie et al., (2019) [25]	4.7	13.4	3.6	10.3
Sabour et al., (2019) [39]	4.5	13.3		
Our Work				
LAS	4.1	12.5	3.2	9.8
LAS + SpecAugment	**2.8**	**6.8**	**2.5**	**5.8**

4.2. Switchboard 300h

For Switchboard 300h, we use the Kaldi [40] "s5c" recipe to process our data, but we adapt the recipe to use 80-dimensional filter banks with delta and delta-delta acceleration. We use a 1k WPM [26] to tokenize the output, constructed using the combined vocabulary of the Switchboard and Fisher transcripts.

We train LAS-4-1024 with policies (None, SM, SS) and schedule B. As before, we set the peak learning rate to 0.001 and total batch size to 512, and train using 32 Google Cloud TPU chips. Here the experiments are run with and without label smoothing. Not having a canonical development set, we choose to evaluate the checkpoint at the end point of the training schedule, which we choose to be 100k steps for schedule B. We note that the training curve relaxes after the decay schedule is completed (step s_f), and the performance of the network does not vary much. The performance of various augmentation poli-

cies with and without label smoothing for Switchboard 300h is shown in Table 4. We see that label smoothing and augmentation have an additive effect for this corpus.

Table 4: *Switchboard 300h WER (%) evaluated for LAS-4-1024 trained with schedule B with varying augmentation and Label Smoothing (LS) policies. No LMs have been used.*

Policy	LS	SWBD	CH
-	×	12.1	22.6
	○	11.2	21.6
SM	×	9.5	18.8
	○	8.5	16.1
SS	×	9.7	18.2
	○	8.6	16.3

As with LibriSpeech 960h, we train LAS-6-1280 on the Switchboard 300h training set with schedule L (training time $\sim$ 24 days) to get state of the art performance. In this case, we find that turning label smoothing on throughout training benefits the final performance. We report the performance at the end of training time at 340k steps. We present our results in the context of other work in Table 5. We also apply shallow fusion with an LM trained on Fisher-Switchboard, whose fusion parameters are obtained by evaluating performance on the RT-03 corpus. Unlike the case for LibriSpeech, the fusion parameters do not transfer well between networks trained differently—the three entries in Table 5 were obtained by using fusion parameters $(\lambda, c) = (0.3, 0.05)$, $(0.2, 0.0125)$ and $(0.1, 0.025)$ respectively.

Table 5: *Switchboard 300h WERs (%).*

Method	No LM		With LM	
	SWBD	CH	SWBD	CH
HMM				
Veselý et al., (2013) [41]			12.9	24.5
Povey et al., (2016) [30]			9.6	19.3
Hadian et al., (2018) [42]			9.3	18.9
Zeyer et al., (2018) [24]			8.3	17.3
CTC				
Zweig et al., (2017) [43]	24.7	37.1	14.0	25.3
Audhkhasi et al., (2018) [44]	20.8	30.4		
Audhkhasi et al., (2018) [45]	14.6	23.6		
LAS				
Lu et al., (2016) [46]	26.8	48.2	25.8	46.0
Toshniwal et al., (2017) [47]	23.1	40.8		
Zeyer et al., (2018) [24]	13.1	26.1	11.8	25.7
Weng et al., (2018) [48]	12.2	23.3		
Zeyer et al., (2018) [38]	11.9	23.7	11.0	23.1
Our Work				
LAS	11.2	21.6	10.9	19.4
LAS + SpecAugment (SM)	**7.2**	14.6	**6.8**	14.1
LAS + SpecAugment (SS)	7.3	**14.4**	7.1	**14.0**

5. Discussion

Time warping contributes, but is not a major factor in improving performance. In Table 6, we present three training results for which time warping, time masking and frequency masking have been turned off, respectively. We see that the effect time warping, while small, is still existent. Time warping, being the most expensive as well as the least influential of the augmentations discussed in this work, should be the first augmentation to be dropped given any budgetary limitations.

Table 6: *Test set WER (%) evaluated without LM for network LAS-4-1024 trained with schedule B.*

W	F	m_F	T	p	m_T	test-other	test
80	27	1	100	1.0	1	10.0	3.7
0	27	1	100	1.0	1	10.1	3.8
80	0	-	100	1.0	1	11.0	4.0
80	27	1	0	-	-	10.9	4.1

Label smoothing introduces instability to training. We have noticed that the proportion of unstable training runs increases for LibriSpeech when label smoothing is applied with augmentation. This becomes more conspicuous while learning rate is being decayed, thus our introduction of a label smoothing schedule for training LibriSpeech, where labels are only smoothed in the initial phases of the learning rate schedule.

Augmentation converts an over-fitting problem into an under-fitting problem. As can be observed from the training curves of the networks in Figure 3, the networks during training not only under-fit the loss and WER on the augmented training set, but also on the training set itself when trained on augmented data. This is in stark contrast to the usual situation where networks tend to over-fit to the training data. This is the major benefit of training with augmentation, as explained below.

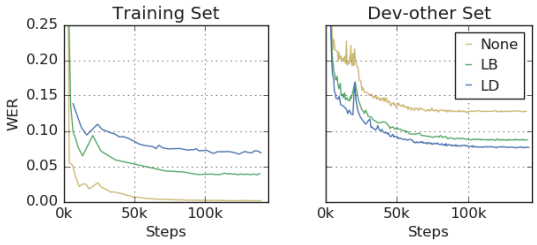

Figure 3: *LAS-6-1280 on LibriSpeech with schedule D.*

Common methods of addressing under-fitting yield improvements. We were able to make significant gains in performance by standard approaches to alleviate under-fitting—making larger networks and training longer. The current reported performance was obtained by the recursive process of applying a harsh augmentation policy, and then making wider, deeper networks and training them with longer schedules to address the under-fitting.

Remark on related works. We note that an augmentation similar to frequency masking has been studied in the context of CNN acoustic models in [49]. There, blocks of adjacent frequencies are pre-grouped into bins, which are randomly zeroed-out per minibatch. On the other hand, both the size and position of the frequency masks in SpecAugment are chosen stochastically, and differ for every input in the minibatch. More ideas for structurally omitting frequency data of spectrograms have been discussed in [50].

6. Conclusions

SpecAugment greatly improves the performance of ASR networks. We are able to obtain state-of-the-art results on the LibriSpeech 960h and Switchboard 300h tasks on end-to-end LAS

networks by augmenting the training set using simple handcrafted policies, surpassing the performance of hybrid systems even without the aid of a language model. SpecAugment converts ASR from an over-fitting to an under-fitting problem, and we were able to gain performance by using bigger networks and training longer.

Acknowledgements: We would like to thank Yuan Cao, Ciprian Chelba, Kazuki Irie, Ye Jia, Anjuli Kannan, Patrick Nguyen, Vijay Peddinti, Rohit Prabhavalkar, Yonghui Wu and Shuyuan Zhang for useful discussions. We also thank György Kovács for introducing us to the works [49, 50].

7. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, Jan 2012.

[3] T. Sainath, A. rahman Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep Convolutional Neural Networks for LVCSR," in *ICASSP*, 2013.

[4] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *ICASSP*, 2013.

[5] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *ICML*, 2014.

[6] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition," in *ICASSP*, 2016.

[7] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-End Attention-based Large Vocabulary Speech Recognition," in *ICASSP*, 2016.

[8] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art Speech Recognition With Sequence-to-Sequence Models," in *ICASSP*, 2018.

[9] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *ASRU*, 2013.

[10] A. Ragni, K. M. Knill, S. P. Rath, and M. J. F. Gales, "Data augmentation for low resource languages," in *INTERSPEECH*, 2014.

[11] N. Jaitly and G. Hinton, "Vocal Tract Length Perturbation (VTLP) improves speech recognition," in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[12] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng, "Deep Speech: Scaling up end-to-end speech recognition," in *arXiv*, 2014.

[13] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for Speech Recognition," in *INTERSPEECH*, 2015.

[14] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home," in *INTERSPEECH*, 2017.

[15] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *ICASSP*, 2015.

[16] A. Raju, S. Panchapagesan, X. Liu, A. Mandal, and N. Strom, "Data Augmentation for Robust Keyword Spotting under Playback Interference," in *arXiv*, 2018.

[17] S. H. Mallidi and H. Hermansky, "Novel neural network based fusion for Multistream ASR," in *ICASSP*, 2016.

[18] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," in *CVPR*, 2019.

[19] T. DeVries and G. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," in *arXiv*, 2017.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[21] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," in *arxiv*, 2015.

[22] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *ICASSP*, 1992.

[23] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text," in *LREC*, 2004.

[24] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *INTERSPEECH*, 2018.

[25] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "Model Unit Exploration for Sequence-to-Sequence Speech Recognition," in *arXiv*, 2019.

[26] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *ICASSP*, 2012.

[27] A. Graves, "Practical Variational Inference for Neural Networks," in *NIPS*, 2011.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[29] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *INTERSPEECH*, 2017.

[30] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *INTERSPEECH*, 2016.

[31] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane, "The CAPIO 2017 Conversational Speech Recognition System," in *arXiv*, 2017.

[32] X. Yang, J. Li, and X. Zhou, "A novel pyramidal-FSMN architecture with lattice-free MMI for speech recognition," in *arXiv*, 2018.

[33] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2Letter: an End-to-End ConvNet-based Speech Recognition System," in *arXiv*, 2016.

[34] V. Liptchinsky, G. Synnaeve, and R. Collobert, "Letter-Based Speech Recognition with Gated ConvNets," in *arXiv*, 2017.

[35] Y. Zhou, C. Xiong, and R. Socher, "Improving End-to-End Speech Recognition with Policy Learning," in *ICASSP*, 2018.

[36] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully Convolutional Speech Recognition," in *arXiv*, 2018.

[37] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An End-to-End Convolutional Neural Acoustic Model," in *arXiv*, 2019.

[38] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A comprehensive analysis on attention models," in *NIPS: Workshop IRASL*, 2018.

[39] S. Sabour, W. Chan, and M. Norouzi, "Optimal Completion Distillation for Sequence Learning," in *ICLR*, 2019.

[40] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *ASRU*, 2011.

[41] K. Vesely, A. Ghoshal, L. Burger, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH*, 2013.

[42] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *INTERSPEECH*, 2018.

[43] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in All-Neural Speech Recognition," in *ICASSP*, 2017.

[44] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct Acoustics-to-Word Models for English Conversational Speech Recognition," in *INTERSPEECH*, 2018.

[45] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *ICASSP*, 2018.

[46] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *ICASSP*, 2016.

[47] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask Learning with Low-Level Auxiliary Tasks for Encoder-Decoder Based Speech Recognition," in *INTERSPEECH*, 2017.

[48] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, "Improving Attention Based Sequence-to-Sequence Models for End-to-End English Conversational Speech Recognition," in *INTERSPEECH*, 2018.

[49] G. Kovács, L. Tóth, D. Van Compernolle, and S. Ganapathy, "Increasing the robustness of cnn acoustic models using autoregressive moving average spectrogram features and channel dropout," *Pattern Recognition Letters*, vol. 100, pp. 44–50, 2017.

[50] L. Tóth, G. Kovács, and D. Van Compernolle, "A perceptually inspired data augmentation method for noise robust cnn acoustic models," in *SPECOM*, 2018.