# Study of Data Augmentation with Various Distortions

**Xiwen Chen***, **Sunghwan Baek***, **Matthew Saenz***
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA

## Abstract

Data augmentation is a crucial source of potential improvement in speech recognition tasks, which can enlarge the dataset, improve scalability to real-life applications, and result in potentially more robust models. In real-world scenarios, various categories of distortion exist, which lead to great challenges to speech recognition models trained on clean data. Motivated by this reasoning, we performed two experiments utilizing existing data augmentation methods provided by Torchaudio (Yang et al. 2022), which were applied to the LibriSpeech 100-hour data set, and trained using the Branchformer architecture (Panayotov et al. 2015). In the first experiment, distortions were applied incrementally, and the relative improvement of each step was compared to determine the effectiveness of each data augmentation. In the second, distortions were added independently of one another, and the performance of each model relative to the baseline was compared to determine the effectiveness of each data augmentation method. This experiment was then extended by training an ASR model using data augmented through speed perturbation and the addition of environmental noise, which produced the best-performing model of our project, with an average character error rate (CER) of 5.2% and average word error rate (WER) of 11.1%, in comparison to our baseline of 7.0% average CER and 14.8% average WER.

## Introduction

In order to improve the model's robustness, data augmentation is widely applied in ASR training (Ko et al. 2015; Kanda, Takeda, and Obuchi 2013; Oneata and Cucu 2022). Compared with feature-based augmentation methods that utilize transformations, flips, time warping, or masking that operate on audio features, we focus on distortion methods including Room Impulse Response (RIR), adding background noise, filtering, and codec (Oneata and Cucu 2022). This comes from the motivation that although feature-based augmentation increases the training data size, most resulting data points are not realistic, and can never happen in real-life scenarios (Ribas, Vincent, and Calvo 2016).

*Equal contributions



Ribas et al. studied the speech distortion in real-life scenarios, and provided preliminary guidelines towards designing experimental setup (Ribas, Vincent, and Calvo 2016). However, it remains to be explored how state-of-the-art speech models improve with distorted audio data. Based on a popular Branchformer model, we apply multiple audio distortions that simulate the real-world scenarios on the training data, and analyze the results compared with baseline (Peng et al. 2022).

## Related studies

A number of data augmentation methods have been proposed to improve the performance of ASR tasks, of varying conceptual and computational complexity. Ko et. al. (2015) investigated speed perturbation as a data augmentation technique by creating a training set consisting of the audio at its original speed, as well as at 90% and 110% speed. As part of their novel ASR architecture, Hannun et. al (2014) explored data augmentation through noise addition (Hannun et al. 2014). In order to prevent the model from learning the pattern of a single noise track, unique noise tracks totalling the length of the training data were used for this augmentation. In testing on noisy speech, the noisy trained model achieved a 22.6% WER over the clean trained model's 28.7% WER, a 6.1% absolute improvement (Hannun et al. 2014). Ko et. al. (2017) showcases the reverberation distortion method. In this method, training data is convolved with a Room Impulse Response (RIR), which encodes information about acoustic characteristics of a room or an environment (Ko et al. 2017a).

One data augmentation method of particular note is SpecAugment. Initially proposed in (Park et al. 2019), this computationally inexpensive method operates on the log mel spectrogram of the input audio, rather than the raw audio itself. Three types of deformation are applied to the log mel spectrogram; a deformation of the time-series in the time direction, consecutive time block masking and consecutive frequency block masking (Park et al. 2019).

## Problem Formulation

The task we are focusing on is automatic speech recognition, and we will apply multiple distortion methods for data augmentation. The ASR task is formulated as follows. Given

an audio signal $S = (\mathbf{s}_1, \ldots, \mathbf{s}_T)$, we would like to output the corresponding word sequence $W = (w_1, \ldots, w_N)$, where $T$ and $N$ are audio length and sequence length, respectively. End-to-end models fit best in our purpose, and we will use Branchformer as the baseline (Peng et al. 2022; Watanabe et al. 2018).

To apply distortion to the audio signal, we will utilize the torchaudio library, and apply one or more distortions sequentially to mimic the real-world scenarios (Yang et al. 2022). Namely, we define the distortion function as

$$F = f_1 \circ f_2 \circ \cdots \circ f_l : S \mapsto S' = (\mathbf{s}_1, \ldots, \mathbf{s}_{T'}),$$

where $l$ is the number of distortion functions we use in the sequence. Then the augmented signals will be used for training. We will experiment on multiple combinations of distortion functions, including RIR, background noise, filtering, and codec provided by torchaudio (Yang et al. 2022).

# Method

## SpecAugment

SpecAugmentation enhances the model's ability to generalize from varied audio inputs by modifying the spectrogram of audio signals. It employs two primary techniques: frequency and time masking. Frequency masking alters a portion of the frequencies in the spectrogram, while time masking modifies a segment of the time domain. These manipulations help reduce the model's overfitting of the training data and improve its ability to handle unseen audio variations.

## Speed Perturbation

Speed Perturbation, a part of the ESPnet's ASR algorithm, enhances the model's adaptability to different speech tempos without affecting the pitch. This is achieved by resampling the audio to adjust the playback speed. This technique is beneficial for preparing the model to understand speech from speakers who talk at varying speeds, further enhancing the versatility of the ASR system.

## Noise

Noise Augmentation involves artificially adding background noise to the audio waveform tensor. This technique aims to enhance the robustness of ASR models in noisy environments by simulating real-world scenarios. The process includes selecting a noise level and adding it to the original audio signal, ensuring the model can effectively process and understand speech in various noise types.

In our project, we used two types of noises.

1. Random noise sampled from a normal distribution and amplified with a uniformly sampled factor ranging from 0.1 to 0.3. The random noise is then added to the original signal.
2. We downloaded background noise from Columbia Noise sample, which is a set of background noise from babble, airport, restaurant, exhibition, street, car, subway and train (col ).

Table 1: Scene Configuration

| Scene | Impulse | Noise (SNR) | Effect | Codec |
|---|---|---|---|---|
| Airport | - | airport (20) | - | - |
| Confrence | - | exhibition (20) | echo | - |
| Phone | - | babble (25) | lowpass, echo | g722 |
| Restaurant | house | restaurant (20) | lowpass, echo | pcm_mulaw |
| Street | - | street (20) | lowpass, echo | - |

## Reverberation

Reverberation Augmentation is designed to improve model performance in echoic environments. It simulates different acoustic spaces, such as large halls or small rooms, by convolving the original audio signal with Room Impulse Response (RIR) waveforms. This method helps ASR models to better understand and interpret speech in environments with varying reverberation characteristics. We used the openSLR room impulses and noise database for the first part of our experiment, and room impulses recorded in different space characteristics provided by Sonic Palimpsest Impulse Response library (Ko et al. 2017b; son ).

## Simulated Scenes

As a step forward, we would like to simulate common scenes that might happen in real-life scenarios using the distortion methods combined. Based on previous experiments, we used noise audios corresponding each scene and chose possible room impulse. Then we manually chose other parameters including SNR for adding noise, predefined effect including echoing, and codec type provided by torchaudio. For instance, we used no impulse, street noise with SNR of 20, low-pass filtering effect and no codec to simulate the street scene. The scene configurations used in our experiment are listed in Table 1.

# Experiments

## Dataset

The dataset utilized in this study is the LibriSpeech, a meticulously curated speech corpus derived from English audiobooks(Panayotov et al. 2015). The dataset, sampled at 16kHz, comprises a comprehensive 100 hours of clean speech data, which will be the focal point of our experiments and analysis in this project. Subsequently, we attempt to enhance the robustness of the model to various environmental conditions by employing torchaudio to incorporate room reverberation and background noise(Yang et al. 2022). The dataset statistics are shown in Table 2, as is provided in (Panayotov et al. 2015).

## Baseline

We use Branchformer as the baseline (Peng et al. 2022). It is a flexible, interpretable and customizable encoder alternative to Conformer consisting of two parallel branches to capture local and global context information for end-to-end ASR(Gulati et al. 2020).

Table 2: LibriSpeech-100 Hours

| Subset | Hours | Per-speaker Minutes |
|---|---|---|
| train-clean-100 | 496.7 | 30 |
| dev-clean | 5.4 | 8 |
| dev-other | 5.3 | 8 |
| test-clean | 5.4 | 10 |
| test-other | 5.1 | 10 |

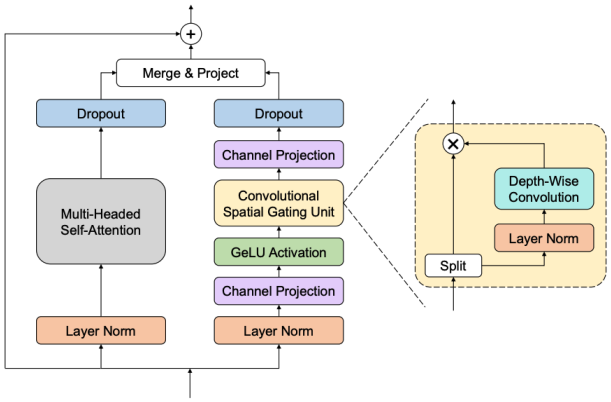


Figure 1: Branchformer encoder block.

The basic architecture of Branchformer encoder is shown in Figure 1 (Peng et al. 2022). It consists of two branches sharing the same Mel feature input, and then extracts global and local context information using multi-head self-attention or cgMLP (MLP with convolutional gating) (Sakuma, Komatsu, and Scheibler 2021) followed by a dropout layer. The output of the two branches are merged via concatenation or weighted average.

## Experimental Setup

Our experiment contains two directions to explore the effect of applying distortions in different schemes. The pipelines for these experiments are shown in Figure 2 and Figure 3, respectively. We used ESPNet for our experiments, and applied distortion to form customized datasets before further data preprocessing stages provided by ESPNet (Watanabe et al. 2018). There are two methods for constructing our data, corresponding to each of our two experiments.

1. The pipeline is shown in Figure 2. We embark on the audio data distortion process with a multi-stage approach meticulously designed to enhance robustness. The initial stage introduces noise, creating a baseline distortion layer replicating common real-world audio disturbances. This foundational noise addition sets the stage for the subsequent application of SpecAugment. This step involves manipulating the audio's spectrogram, challenging our model against variations in both frequency and time domains. During this phase, we meticulously optimize parameters, experimenting with different settings to find the most effective combination for the noise and SpecAugment mix.

   After achieving the optimal result with this combination, we strategically introduce small room reverberation. This is carefully blended at a 15% level between the noise and SpecAugment stages. The rationale behind this intermediate addition of reverberation – a prevalent environmental element – is to ensure its interaction with the basic noise-altered audio before it undergoes further spectral and temporal modifications. This approach ensures that the model is trained on audio with foundational noise characteristics and is enriched with realistic environmental effects.

   The final stage in our process is Speed Perturbation, which is implemented through ESPnet's built-in function. This stage is critical for simulating different speech rates, ensuring the model's adaptability across various speaking tempos. Through this structured progression – starting with a simple noise introduction, then strategically blending in reverberation, then sophisticated spectral and temporal alterations with SpecAugment, and culminating in speed perturbation – we achieve comprehensive conditioning of the audio data.

2. As is shown in Figure 3, we independently test with individual types of distortions, including noise, effects provided by torch audio, codec, and reverberation. Each of the single distortions consists of a set of distortion functions with different parameters. In the data augmentation step, we randomly sampled one of the configurations for each audio sample. As a step forward, we combine all these distortions with manually tuned parameters to simulate audio signals that resemble real-life scenarios. Based on the best-performing combination from the previous steps, we add the speed perturbation provided by ESPNet.

As mentioned above, we used Branchformer as our baseline architecture, and trained the model using our augmented data together with the original data for 70 epochs on single V100 GPU with 16G memory. The optimization is Adam with learning rate of 0.002 and a decay of $10^{-6}$. The initial 15000 training steps are used for warmup.

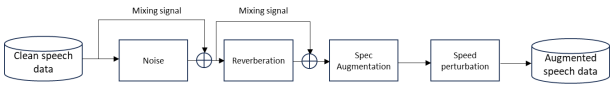


Figure 2: Sequential speech data augmentation pipeline for the first experiment.

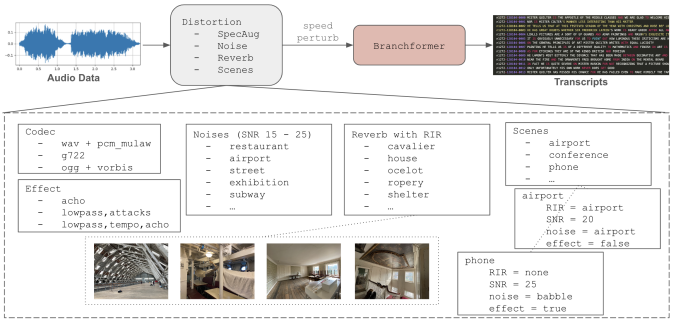


Figure 3: Distortion pipeline for the second experiment.

Table 3: Experiment results (CER and WER). The *comb* notation means the distortions are chosen randomly from a pool of different schemes.

| **Method** | **CER** | | | | **WER** | | | |
|---|---|---|---|---|---|---|---|---|
| | dev-clean | dev-other | test-clean | test-other | dev-clean | dev-other | test-clean | test-other |
| Branchformer | 3.1 | 10.8 | 3.2 | 10.8 | 7.9 | 21.3 | 8.2 | 21.8 |
| *+[sp]* | 2.6 | 8.3 | 2.6 | 8.3 | 6.3 | 16.8 | 6.5 | 17.1 |
| *+[sp, specaug]* | 2.5 | 8.0 | 2.6 | 8.0 | 6.2 | **16.0** | 6.5 | 16.3 |
| *+[sp, specaug, noise]* | 2.5 | 8.1 | 2.5 | 7.9 | **6.1** | 16.2 | **6.3** | 16.1 |
| *+[sp, specaug, noise, reverb]* | **2.5** | **7.9** | **2.5** | **7.9** | 6.2 | 16.1 | 6.4 | **16.1** |
| *+[effects(comb)]* | 2.9 | 9.2 | 3.0 | 9.3 | 7.1 | 18.4 | 7.4 | 18.9 |
| *+[reverb(comb)]* | 3.5 | 10.2 | 3.4 | 10.1 | 8.1 | 19.7 | 8.2 | 20.0 |
| *+[noises(comb)]* | 2.8 | 9.1 | 2.8 | 9.2 | 6.9 | 18.2 | 7.0 | 18.6 |
| *+[scenes(comb)]* | 3.0 | 9.3 | 3.1 | 9.4 | 7.2 | 18.4 | 7.6 | 18.9 |
| *+[sp, noise(comb)]* | **2.5** | **7.9** | **2.4** | **7.8** | **6.1** | **15.8** | **6.3** | **16.1** |

## Results and Discussion

The results for word error rate (WER) and character error rate (CER) for each split of develop and test data split are shown in Table 3. The top section of the table shows the results of our first experiment, where data augmentations were sequentially added to the model, while the bottom section shows the results of our second experiment, where data augmentations were independently added to the model.

1. In the first experiment, we saw the largest single-augmentation improvement from the application of speed perturbation, which produced an average relative improvement of 20.3% in CER and 20.9% in WER over baseline. The subsequent incremental improvements over the previous model were smaller, with the final addition of reverberation actually seeing an loss of performance in WER over the previous model. The addition of SpecAugment resulted in an average relative improvement of only 2.77% in CER and 2.76% in WER over speed perturbation only, the addition of background noise resulted in an average relative improvement of 0.96% in CER and 1.17% in WER over speed perturbation and SpecAugment, and the addition of reverberation resulted in an average relative improvement of 0.62% in CER and an average relative performance reduction of 0.65% in WER. The final model had an average relative improvement of 23.7% in CER and 23.5% in WER and an average absolute improvement of 1.8% CER and 3.6% CER over baseline. The greatest performance gains were seen in the test-other dataset, where CER was reduced by 2.9% and WER by 5.7%.
2. Of the four data augmentations tested independently in experiment 2, the addition of environmental noise produced the greatest improvement over baseline, with an average relative improvement of 13.2% in CER and 14.1% in WER. Effects produced the next greatest improvement, with an average relative improvement of 10.4% in CER and 11.7% in WER over baseline. Simulated Scenes produced the third best performance with an average relative improvement of 8.30% in CER and 10.8% in WER. Reverberation had the worst performance of the experiment, with an average relative performance loss of 1.78% in CER and a low average relative improvement of 3.31% in WER. With these results we then applied speed perturbation, the data augmentation with the greatest contribution to performance improvement in experiment 1, alongside the addition of environmental noise, the best performing distortion of the four data augmentations in experiment 2, to produce a model with performance slightly exceeding the performance of the final incremental model of experiment 1. This model had an average relative improvement of 24.7% in CER and 24.5% in WER and an average absolute improvement of 1.8% CER and 3.8% WER over baseline.

## Conclusion

In this project, we performed two experiments to explore the effect of applying data augmentations to training data in different schemes on ASR model performance. Both experiments used Branchformer as the baseline architecture and were trained using the same set of hyperparameters.

In the first experiment, the most effective strategy emerged when we integrated all four augmentation techniques—noise addition, reverberation, SpecAugmentation, and speed perturbation. This combination was not arbitrary; it was predicated on the hypothesis that each method targets distinct characteristics of audio signal variability. By fine-tuning each method for optimal parameter efficiency, we developed a model that excels in handling the intricate challenges of real-world audio scenarios. This holistic approach didn't just enhance the model's robustness—it also laid down a strategic framework for future enhancements in speech recognition technology

In the second, multiple distortions were experimented in both independent and combined schemes, and were manually chosen to simulate real-life scenarios. We found that the most effective data augmentation studied in this experiment was the addition of environmental noise. This setting was then extended by training an ASR model using data augmented through speed perturbation and the addition of environmental noise, which produced the best-performing model of our project.

## References


Columbia noise samples. https://www.ee.columbia.edu/~dpwe/sounds/noise/.

Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; and Ng, A. Y. 2014. Deep speech: Scaling up end-to-end speech recognition.

Kanda, N.; Takeda, R.; and Obuchi, Y. 2013. Elastic spectral distortion for low resource speech recognition with deep neural networks. In *2013 IEEE workshop on automatic speech recognition and understanding*, 309–314. IEEE.

Ko, T.; Peddinti, V.; Povey, D.; and Khudanpur, S. 2015. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*.

Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M. L.; and Khudanpur, S. 2017a. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5220–5224.

Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M. L.; and Khudanpur, S. 2017b. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5220–5224. IEEE.

Oneata, D., and Cucu, H. 2022. Improving multimodal speech recognition by data augmentation and speech representations. *arXiv preprint arXiv:2204.13206*.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 5206–5210. IEEE.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*, interspeech 2019. ISCA.

Peng, Y.; Dalmia, S.; Lane, I.; and Watanabe, S. 2022. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *International Conference on Machine Learning*, 17627–17643. PMLR.

Ribas, D.; Vincent, E.; and Calvo, J. R. 2016. A study of speech distortion conditions in real scenarios for speech processing applications. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, 13–20. IEEE.

Sakuma, J.; Komatsu, T.; and Scheibler, R. 2021. Mlp-based architecture with variable length input for automatic speech recognition.

Sonic palimpsest impulse responses. https://research.kent.ac.uk/sonic-palimpsest/impulse-responses/.

Watanabe, S.; Hori, H.; Karita, S.; et al. 2018. ESPnet: End-to-end speech processing toolkit. In *Interspeech*.

Yang, Y.-Y.; Hira, M.; Ni, Z.; Astafurov, A.; Chen, C.; Puhrsch, C.; Pollack, D.; Genzel, D.; Greenberg, D.; Yang, E. Z.; et al. 2022. Torchaudio: Building blocks for audio and speech processing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6982–6986. IEEE.