AdSMOTE: A Technique for "High Proportion" Audio Augmentation

Kavisha Jayathunge, Xiaosong Yang, Richard Southern

Bournemouth University
{kjayathunge, xyang, rsouthern}@bournemouth.ac.uk

Abstract

Data augmentation is a practice that is widely used in the fields of machine and deep learning. It is used primarily for its effectiveness in reducing the generalisation gap between training and validation, as well as to artificially increase in available training data points. This is particularly relevant to audio datasets, which are usually smaller and suffer from imbalanced classes in some applications. This work presents adSMOTE (audio SMOTE), a novel sampling and augmentation strategy and also compares it to Specaugment, one of the most effective augmentation strategies for audio data. We show that our method outperforms the latter by a considerable margin when the proportion of synthetic training samples is high. We also provide source code for the complete algorithm, which can easily be integrated into an existing model, enabling the rapid development of augmentation frameworks.

1 Introduction

When gathering training data for a deep learning application, it is necessary to have a sufficiently large dataset – if it is too small, it might cause the model to over-fit to the data when training. Augmentation is the practice of artificially increasing the size of a dataset by adding new, meaningful data points to it. In the case of audio, this can be done by transforming either the raw waveform or a latent representation of a given piece of audio. The latter is the method employed by Specaugment (Park et al. 2019), the current state of the art. One commonly observed phenomenon in many audio datasets with emotion class labels is class imbalance (Poria et al. 2020; Cao et al. 2014; Livingstone and Russo 2018). This is where particular emotions are over-represented (e.g. neutral, happy), while others are under-represented (e.g. disgust, anger). These "minority classes" therefore need to be augmented at a high rate to reach parity with the other classes in the dataset. We show that Specaugment is not particularly suited to this, as it works by randomly masking regions of the audio spectrogram, which is a destructive process that removes some of the data available to the model. We hypothesise that a method that just transforms the spectrogram without masking would lead to better model performance.

The usage of very highly augmented datasets is scarce, with some text model studies performing upto 40% data augmentation (Kim et al. 2021). However, to our knowledge, the investigation of the relationship between the proportion of audio augmentation and the performance of deep learning models has not been formally conducted.

2 Related Work

Methods proposed in order to solve the problem of audio augmentation fall into two camps: ones that operate in the signal domain and ones that operate in the frequency domain. Audio augmentation can be approached in many ways and often requires a good understanding of the data and what kind of application the augmented data is to be used in. For instance, work done by Nagano et al. (Nagano et al. 2019) leverages the fact that young children sometimes exhibit a certain disfluency in speech wherein they prolong vowel sounds. This naturally occurring phenomenon was exploited to create augmented children's speech by artificially lengthening of vowel sounds.

More recently, there have been advances in Mel-spectrogram augmentation techniques. Since spectrograms are essentially 2D images, augmentation methods are often inspired by analogous techniques in image augmentation such as cropping, scaling and warping (Maguolo et al. 2021). Specaugment is an example of such an augmentation technique intended to simulate the irregularities and errors in speech recordings. It does this by randomly masking some of the frequency and time components of an audio clip's Mel-spectrogram. It also applies warping to the spectrogram. The method achieved state of the art performance on many speech recognition tasks, and has since been extended to be more computationally efficient and work on a wider variety of audio signals (Jain et al. 2021).

As mentioned earlier, Specaugment is a destructive augmentation process, which might lead to adverse effects when training a speech generation task. We therefore introduce a novel audio augmentation strategy that transforms the audio signal directly, drawing inspiration from an existing over-sampling method – Synthetic Minority Over-representation Technique (SMOTE) (Chawla et al. 2002). We then train a speech synthesis deep learning model, and show that the proposed augmentation method outperforms the current state of the art at high levels of augmentation.

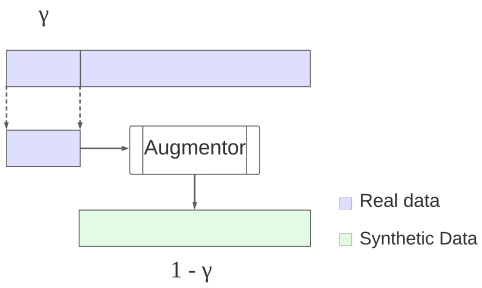

Figure 1: The original batch (top) is sub-sampled according to the gamma parameter, and then an augmentation method (either ours or Specaugment) is applied such that the proportion of synthetic samples (bottom) is $1 - \gamma$. The synthetic data are then added to the batch along with the real data that they are derived from. The final batch size is therefore the same as the original. A smaller γ corresponds to a larger fraction of synthetic data in the batch.

3 Methodology

3.1 The Gamma Parameter

Because we are interested in investigating the relationship between the proportion of synthetic data in a batch and model performance, it is necessary to parameterise this proportion – we use the parameter Gamma (γ) for this. It represents the fraction of real data points, the rest are generated by either augmentation method. This means that the lower the value of γ, the higher the proportion of synthetic data in the batch.

3.2 AdSMOTE Algorithm

Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al. 2002) was developed as a way to counter the problem of imbalanced data classes within datasets used for classification tasks (Sarakit, Theeramunkong, and Haruechaiyasak 2015). It works by interpolating between a point in the dataset and one of its nearest neighbours, and has shown to be more effective than similar techniques such as Gaussian noise, which might produce values that may lie outside the distribution of the data set (Arslan et al. 2019). Following a similar strategy, we propose a novel augmentation strategy that is applicable to audio data.

In this work, we extract two features from each audio clip: the average fundamental frequency $f0$ of the speaker in the clip, and the root mean square rms of the time series signal s. These features are given by:

$$f_0 = \frac{1}{N}\sum_{i=1}^{N}\pi(s) \tag{1}$$

$$rms = \sqrt{\frac{1}{L}\sum_{i=1}^{L}s^2} \tag{2}$$

where L is the length of the audio signal in samples and $\pi(\cdot)$ refers to the probabilistic fundamental frequency extraction algorithm (pYIN) (Mauch and Dixon 2014; de Cheveigne and Kawahara 2002). This algorithm generates N instantaneous fundamental frequency values (window length $93ms$, non-overlapping) which are averaged to get f_0. Doing this for each audio clip a dataset allows us to build a 2D feature space for this dataset, where each point corresponds to a unique audio clip – this feature space is then made available to the model at run-time.

Algorithm 1: A high-level overview of the proposed augmentation procedure. Arrays are in bold.

inputs : Input batch of signals **batch**, pre-cached feature space **fspace**, desired proportion of non-synthetic data γ, number of nearest neighbours to calculate k
output: Batch of augmented signals **aug_batch**

aug_batch $\leftarrow [Empty]$
$batch\_size \leftarrow$ `length(`**batch**`)`
$N_{real} \leftarrow$ `round(`$\gamma \times batch\_size$`)`
`// populate real data in aug batch`
for $idx \in [0, N_{real})$ **do**
  `append(`**aug_batch**, **batch**$[idx]$`)`
end
$idx \leftarrow 0$
while `length(`**aug_batch**`)` $< batch\_size$ **do**
  $sig \leftarrow$ **batch**$[idx]$
  $p_{src\_f0} \leftarrow$ `getFFreq(`sig`)`
  $p_{src\_rms} \leftarrow$ `getRMS(`sig`)`
  $p_{src} \leftarrow [p_{src\_f0}, p_{src\_rms}]$
  nns $\leftarrow$ `getNeighbours(`p_{src}, **fspace**, k`)`
  if $k \geq 3$ **then**
    `// need >=3 points for polygon`
    hull $\leftarrow$ `getConvHull(`**nns**`)`
    samples $\leftarrow$ `sampleHull(`**hull**, $N_{samples}$`)`
  else
    `// basic SMOTE`
    $p_{nn} \leftarrow$ **nns**$[0]$
    $p_{synth} \leftarrow$ `interpolate(`p_{src}, p_{nn}`)`
    samples $\leftarrow [p_{synth}]$
  end
  for $p_{tgt\_f0}, p_{tgt\_rms} \in$ **samples do**
    $aug\_sig \leftarrow$ `pitchShift(`$sig, \frac{p_{tgt\_f0}}{p_{src\_f0}}$`)`
    $aug\_sig \leftarrow$ `volumeShift(`$sig, \frac{p_{tgt\_rms}}{p_{src\_rms}}$`)`
    `append(`**aug_batch**, aug_sig`)`
  end
  $idx \leftarrow (idx + 1) \bmod N_{real}$
end
`// truncate to batch size`
aug_batch $\leftarrow$ `trunc(`**aug_batch**, $batch\_size$`)`
return aug_batch

As explained in Section 3.1, only a certain portion of the input batch is transformed. For each one of these, if only 1

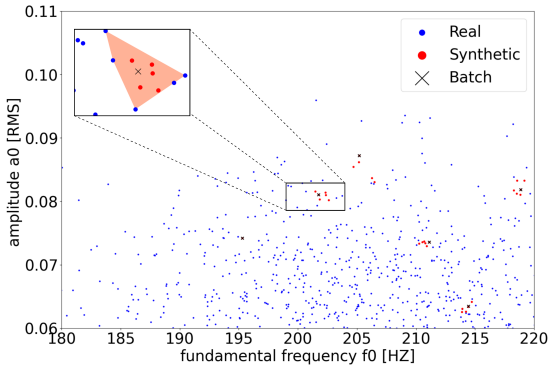

Figure 2: Visualisation of the adSMOTE algorithm as applied to one batch (batch size = 32, $\gamma = 0.25$, $k = 5$) of the LJ-Speech dataset, superimposed upon the rest of the dataset. The inset plot shows how the convex hull of the nearest neighbours of the batch point is sampled to generate the synthetic points.

nearest neighbour is considered, the synthetic point is generated by random uniform interpolation between the batch point and its nearest neighbour. If however the number of nearest neighbours is ≥ 3, we can consider this collection of points to be a polygon, inside which we uniformly sample to get the new synthetic data points – refer to Figure 2. In the case where $k = 2$, it is not possible to do the polygon sampling because we won't have at least 3 vertices to create a polygon. In this case, we could use the batch point itself as the 3rd vertex, essentially setting $k = 3$. For the purposes of this study, we set the number of samples to be 5.

We then use the Sound Exchange (Bagwell 2015) software (which provides the `pitchShift` and `volumeShift` procedures in Algorithm 1) to turn the source signals into target ones, for which the distance between the source and target is calculated as follows:

$$d_{vol} = \left(\frac{rms_{target}}{rms_{source}} \right) \tag{3}$$

$$d_{pitch} = 1200 \log_2 \left(\frac{f_{0\_target}}{f_{0\_source}} \right) \tag{4}$$

where d_{vol} is the volume ratio of the target signal to the source signal, and d_{pitch} is the pitch difference from the source signal to the target signal in cents, a unit used to measure the distance between two frequencies – we use the formula given by Ellis (Ellis 1885) in Equation 4.

4 Experiments

The experiments all use validation losses as the metrics for comparison, and we augment only the training batch. In addition to validation performance, the normalised generalisation gap is also calculated – this can be thought of as the gap between the training and validation: a measure of how well the model is able to generalise to unseen examples. This

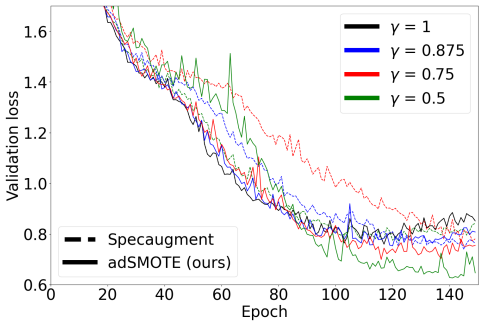

Figure 3: The validation loss curves for Specaugment and adSMOTE with $k = 1$. It is observed that the difference between our method and Specaugment is more prominent when the proportion of synthetic data is higher, i.e. γ is lower.

can then be divided by the train metric to get the normalised generalisation gap g:

$$g = \left| \frac{M_v - M_t}{M_t} \right| \tag{5}$$

where M_t refers to the value of the training metric (validation loss in this case) and M_v refers to the validation metric. A lower g therefore indicates the model has learned a good representation of the data as there isn't much difference in performance between training and validation.

For the purposes of these experiments, a Specaugment implementation derived from (Caceres 2019) was used with $W = 10$, $T = 40$, $F = 30$. These parameters were chosen following a hyperparameter search in which we determined the best configuration for this application. As in the original Specaugment paper, we use $N_{masks} = 2$ for both frequency and time domains. All plots are the result of averaging the metrics over 2 runs.

4.1 Model and Dataset

The objective of these experiments is to investigate how the proportion of synthetic samples affects the performance of Glow-TTS (Kim et al. 2020), a text-to-speech application. This flow-based model is robust to long input texts and scales well to multi-speaker tasks, and models of this kind are being extensively and actively researched (Cong et al. 2021; Casanova et al. 2021; Miao et al. 2020; Biliński et al. 2022). It was selected in this study in the hopes that our findings are generally applicable to the field of speech generation.

The dataset used to train the text-to-speech model was the LJSpeech dataset (Ito and Johnson 2017), an English-language, single-speaker collection of short audio clips read by a female speaker in a neutral tone.

4.2 Gamma Comparison

We initially look at how γ affects the validation loss of the model with $k = 1$ by running experiments on a range of γ values $(1, 0.875, 0.75, 0.5)$. These preliminary results can be seen in Figure 3, where it is noted that our method (solid

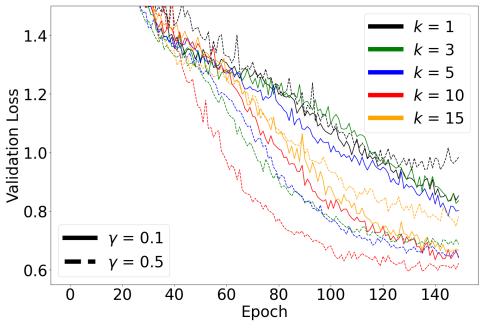

Figure 4: A comparison of model performance over a range of k values, showing that the validation loss keeps decreasing as k is increased until $k = 10$ (red line, lowest loss value in the graph). After this point (i.e. $k = 15$), validation loss increases. We therefore take $k = 10$ to be our best performing variant.

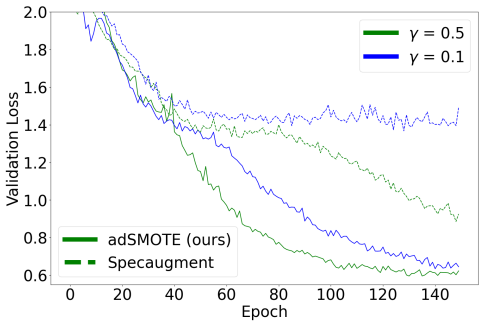

Figure 5: A comparison of our best-performing adSMOTE variant ($k = 10$) and Specaugment, across low values of γ, showing that ours significantly outperforms the latter under these conditions.

line) does not show much of a difference in model performance over Specaugment (dotted line) for larger values of γ. However, we can see at $\gamma = 0.5$ that adSMOTE performs much better than the other augmentation technique.

Motivated by this finding, we continue in this direction by using smaller values of γ for the future experiments.

4.3 Using Multiple Nearest Neighbours

We also parameterise the number of nearest neighbours, k, that are considered for each point that needs to be transformed. In this experiment, we compare our method against itself, with $k = 1, 3, 5, 10, 15$.

Figure 4 shows that $k = 10$ is our best performing variant, and we use this to compare against Specaugment in another series of experiments that are summarised in Figure 5. Here, we see that for each γ, our method outperforms Specaugment by a considerable margin.

Since one of the main motivations behind augmentation is to make the model robust to over-fitting, we also calculate the normalised generalisation gap as per Equation 5. These curves are presented in Figure 6 show that the generalisation gap increases rapidly with no augmentation at all. This effect

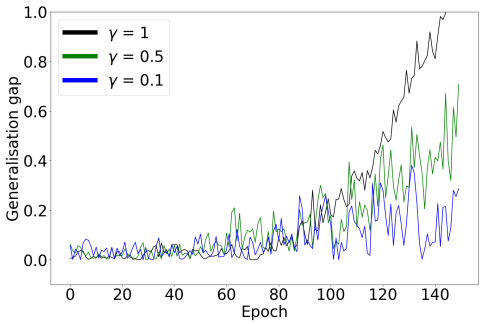

Figure 6: Plot showing the normalised generalisation gap between the training and validation metrics of adSMOTE with $k = 10$. We see that our method's generalisation gap gets lower as γ decreases, i.e. as the fraction of synthetic points in the data batch increases.

is reduced when augmentation is applied, and our method's generalisation gap is lower when the proportion of synthetic data is increased.

5 Limitations

One disadvantage of this method is the need for a pre-calculated feature space, as described in Section 3.2. We maintain this feature space so that items in the batch can be compared to other points in the dataset, which means that this feature space needs to be calculated once for each dataset. This is in contrast to most other augmentation techniques, which work without the use of such pre-cached data.

6 Conclusion

We have presented adSMOTE, a novel non-destructive audio augmentation technique which synthesises audio data by sampling the neighbourhood of the original point. The increase in performance over Specaugment, particularly as applied to speech generation, could be explained by the fact that we leave the full information content of the data intact. By uniformly sampling within the convex hull of the neighbouring points, it effectively interpolates on a 2D plane, drawing synthetic points that are close to the original distribution of the data.

We show that when a large portion of the training batch of a speech generation model is made up of synthetic samples, adSMOTE starts to significantly outperform Specaugment, a state-of-the-art augmentation method.

It is in principle generalisable to a higher dimensional feature space, although that was not explored in this work. It would also be beneficial to study this effect on other datasets – currently, we only test on the LJSpeech dataset. Further experiments on different speech datasets with multiple speakers would be helpful in determining whether adSMOTE can generally be applied to audio augmentation.

References

Arslan, M.; Guzel, M.; Demirci, M.; and Ozdemir, S. 2019. SMOTE and Gaussian Noise Based Sensor Data Augmen-

tation. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 1–5.

Bagwell, C. 2015. SoX - Sound eXchange | HomePage. http://sox.sourceforge.net/. Accessed: 2022-04-12.

Biliński, P.; Merritt, T.; Ezzerg, A.; Pokora, K.; Cygert, S.; Yanagisawa, K.; Barra-Chicote, R.; and Korzekwa, D. 2022. Creating New Voices using Normalizing Flows. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2022-September, 2958–2962. ISSN: 2308-457X.

Caceres, Z. 2019. SpecAugment with Pytorch. https://github.com/zcaceres/spec_augment. Accessed: 2022-04-15.

Cao, H.; Cooper, D. G.; Keutmann, M. K.; Gur, R. C.; Nenkova, A.; and Verma, R. 2014. CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, 5(4): 377–390. Publisher: Institute of Electrical and Electronics Engineers Inc.

Casanova, E.; Shulby, C.; Gölge, E.; Müller, N.; de Oliveira, F.; Junior, A.; da Silva Soares, A.; Aluisio, S.; and Ponti, M. 2021. SC-GlowTTS: An efficient zero-shot multi-speaker text-to-speech model. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 5, 3546–3550. ISBN 978-1-71383-690-2. ISSN: 2308-457X.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357. ArXiv: 1106.1813.

Cong, J.; Yang, S.; Xie, L.; and Su, D. 2021. Glow-WaveGAN: Learning speech representations from gan-based variational auto-encoder for high fidelity flow-based speech synthesis. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 5, 3241–3245. ISBN 978-1-71383-690-2. ISSN: 2308-457X.

de Cheveigne, A.; and Kawahara, H. 2002. YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, 111(4): 14.

Ellis, A. J. 1885. *On the Musical Scales of Various Nations*. Journal of the Society of arts. Google-Books-ID: sNtDAAAAYAAJ.

Ito, K.; and Johnson, L. 2017. The LJ Speech Dataset. https://keithito.com/LJ-Speech-Dataset/. Accessed: 2022-03-28.

Jain, A.; Samala, P. R.; Mittal, D.; Jyoti, P.; and Singh, M. 2021. SpliceOut: A Simple and Efficient Audio Augmentation Method. ArXiv: 2110.00046.

Kim, H.; Jeong, J.; Kim, K.-M.; Lee, D.; Lee, H. D.; Seo, D.; Han, J.; Park, D. W.; Heo, J. A.; and Kim, R. Y. 2021. Intent-based Product Collections for E-commerce using Pretrained Language Models. In *2021 International Conference on Data Mining Workshops (ICDMW)*, 228–237. ISSN: 2375-9259.

Kim, J.; Kim, S.; Kong, J.; and Yoon, S. 2020. Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. *34th Conference on Neural Information Processing Systems*. ArXiv: 2005.11129.

Livingstone, S. R.; and Russo, F. A. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north American english. *PLoS ONE*, 13(5): e0196391. Publisher: Public Library of Science.

Maguolo, G.; Paci, M.; Nanni, L.; and Bonan, L. 2021. Audiogmenter: a MATLAB toolbox for audio data augmentation. *Applied Computing and Informatics*. ArXiv: 1912.05472 Publisher: Emerald Group Holdings Ltd.

Mauch, M.; and Dixon, S. 2014. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 659–663. Florence, Italy: IEEE. ISBN 978-1-4799-2893-4.

Miao, C.; Liang, S.; Chen, M.; Ma, J.; Wang, S.; and Xiao, J. 2020. Flow-TTS: A non-autoregressive network for text to speech based on flow. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 2020-May, 7209–7213. ISBN 978-1-5090-6631-5. ISSN: 1520-6149.

Nagano, T.; Fukuda, T.; Suzuki, M.; and Kurata, G. 2019. Data Augmentation Based on Vowel Stretch for Improving Children's Speech Recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings*, 502–508. Institute of Electrical and Electronics Engineers Inc. ISBN 978-1-72810-306-8.

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 *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2613–2617. ArXiv: 1904.08779 ISSN: 19909772.

Poria, S.; Hazarika, D.; Majumder, N.; Naik, G.; Cambria, E.; and Mihalcea, R. 2020. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 527–536. Association for Computational Linguistics (ACL). ISBN 978-1-950737-48-2. ArXiv: 1810.02508.

Sarakit, P.; Theeramunkong, T.; and Haruechaiyasak, C. 2015. Improving emotion classification in imbalanced YouTube dataset using SMOTE algorithm. In *ICAICTA 2015 - 2015 International Conference on Advanced Informatics: Concepts, Theory and Applications*. ISBN 978-1-4673-8143-7.