ParallelChain Lab's Anti-spoofing Systems for ASVspoof 5

Thien Tran, Thanh Bui, Panagiotis Simatis

ParallelChain Lab*

Abstract

The recent rise of generative AI makes detecting audio deepfakes increasingly challenging. Deep learning techniques produce highly realistic fake audio that can deceive both humans and Automatic Speaker Verification (ASV) systems. This paper presents ParallelChain Lab's submissions to the ASVspoof 5 challenge, namely a voice anti-spoofing system and a spoofing-robust ASV system. We developed an ensemble architecture comprising models trained with various augmentation types, including waveform augmentations, mel-spectrogram augmentations, and vocoder synthesis. An extensive experimental evaluation confirms the efficacy of our systems, achieving *minDCF* of 0.2660 for the deepfake detection system and *min a-DCF* of 0.3173 for the spoofing-robust ASV system in the closed condition.

1. Introduction

Advanced deep learning techniques produce synthetic voices capable of spoofing security systems and humans alike. Furthermore, the widespread availability of Text-To-Speech (TTS) and Voice Conversion (VC) tools increases the prevalence of deepfake media. Without robust anti-spoofing measures, ASV systems are at risk of unauthorized access, compromising user trust and data integrity.

ASVspoof challenges lead the development of countermeasures against audio deepfakes. This year's iteration, ASVspoof 5, features two tracks: 1) deepfake detection and 2) Spoofing-robust Automatic Speaker Verification (SASV). Each track offers closed and open conditions, with the open condition permitting the use of external data and pre-trained models [1]. Moreover, the ASVspoof 5 database poses a greater challenge compared to previous years, with recordings captured with diverse devices and acoustic conditions, as well as incorporating state-of-the-art TTS [2, 3], VC [4], and Adversarial Attack [5]. Similar to the previous iteration, ASVspoof 5 provides four data splits for each track: *training*, *development*, and *progress* sets for model development, and *evaluation* set for the final submission.

We participate in the closed condition of track 1 and track 2 to develop innovative, data-independent techniques, and build resilient deepfake detection and SASV systems. This approach ensures that our systems can be further improved using publicly available datasets and pre-trained models. To achieve this goal, we combine findings from past competitions [6, 7, 8] with hand-crafted data augmentations designed specifically for voice anti-spoofing, as well as our expertise in training fast and robust deep learning models. Concretely, our contributions are:

- A secure deepfake detector for voice anti-spoofing.
- A robust SASV system for user authentication.
- A pipeline of augmentations for deepfake detection with limited training data.

The remainder of the paper is organized as follows. Section 2 describes our system architecture and methods used. Section 3 outlines the model training setups, experiments, and results analysis. Lastly, Section 4 concludes the paper.

2. System Overview

2.1. Track 1: Speech deepfake detection (closed condition)

2.1.1. Input features

The two most popular input features for voice anti-spoofing are raw waveform and mel-spectrogram. We choose mel-spectrogram since it can be viewed as a 2D image with a single channel. This enables us to leverage established vision-related architectures and training techniques for model optimization.

Table 1 demonstrates the different settings of the mel-spectrogram transform. The transform parameters are varied across models to increase feature diversity for later model ensembling. In addition, we take the logarithm of the mel-spectrogram to obtain log-energies, and apply mean normalization along the time axis. The last step is equivalent to gain normalization for each mel feature.

The number of mel filters used is considerably larger than the typically chosen values of 80 or smaller [7, 8]. We find this to be beneficial over using longer audio samples and a smaller hop size, suggesting that spectral resolution is important for deepfake detection. Nevertheless, using more than 160 mel filters yields diminishing

*ParallelChain Lab is a software development company with offices in Singapore, Hong Kong, and Bangkok.

Table 1: *Mel-spectrogram parameters.*

Parameter	Value
Window type	Hann, Povey [9]
Window size	400, 512
Hop size	160
FFT size	512
No. of mel filters	120, 128, 160

returns with substantially longer training time. These experiments are omitted for brevity.

2.1.2. Model architecture

For both tracks, we use a modified variant of the Residual Network (ResNet) [10]. In preliminary experimentation, we found that ResNet is not too sensitive to training hyperparameters, thus allowing us to concentrate on designing effective data augmentations.

The ResNet variant used originates from prior speaker verification works [11], and has been adapted for deepfake detection [7]. The first convolution layer (with stride 2) and the max pooling layer are replaced with a single stride 1 convolution layer, reducing the total stride of the model from 32 to 8. With fewer downsampling operations, the model captures more fine-grained details from the mel-spectrogram, which is crucial in identifying deepfake artifacts.

Due to time constraints, we train from scratch the small ResNet-34 architecture (i.e., 7 million parameters) for deepfake detection. Our implementation is adapted from the WeSpeaker toolkit [12].

We briefly experimented with other modern image classification networks such as Vision Transformer (ViT) [13] and ConvNeXt [14]. Although these architectures show impressive results for vision-related tasks, we find them less suitable for voice anti-spoofing. Figure 1 shows ResNet-34's superiority in terms of (i) convergence speed (lower loss at the same number of training steps), (ii) computational efficiency (faster to train), and (iii) training stability (smaller fluctuations in loss curves). This observation agrees with previous works indicating that specific architectures require tailor-made augmentations for effective learning [15, 16].

2.1.3. Data augmentation

To improve the system's robustness and generalization to unseen data, we employ a wide range of augmentation methods. Figure 2 illustrates the effects of our data augmentations on the mel-spectrogram.

Waveform augmentations. These augmentations are applied on the raw waveform before the mel-spectrogram transformation.

1. Time masking: We randomly replace time intervals with zeros. This can be seen as an extension of CutOut [17] to audio data, which helps the model be more resistant to data corruption.

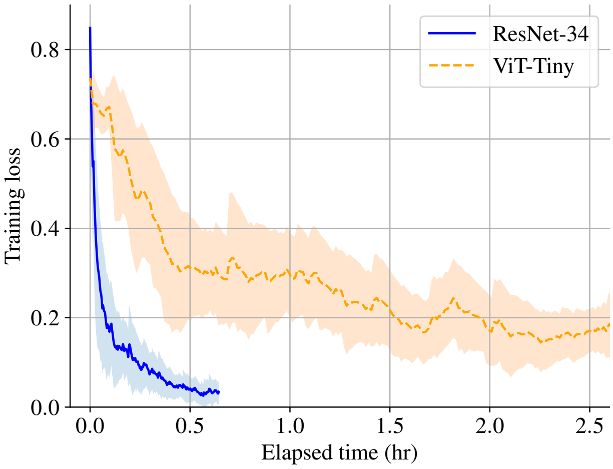

Figure 1: *Training loss (i.e., binary cross entropy) curves of ResNet-34 (7M parameters) and ViT-Tiny (6M parameters) using the same data augmentations and training hyperparameters (10k steps). The curves are smoothed with exponential moving average for clearer visualization, with the shaded region being one standard deviation away from the mean.*

2. Background noise: We sample a random signal-to-noise ratio (SNR) value from a pre-defined range, select a random noise audio, and add it to the training data with the desired SNR. We use the noise and music portions of the MUSAN corpus [18], as well as ASVspoof 5's bona-fide samples from the training split as background noise. To adhere to the challenge's rules, we exclude the vocal samples from MUSAN's music portion.

3. RawBoost [19]: Following the paper's recommendations, we apply linear and non-linear convolutive noise, and impulsive signal-dependent additive noise, but not stationary signal-independent additive noise.

4. Speed perturbation [20]: We adjust the audio speed using audio resampling with the ratio ranging from 0.9 to 1.1.

5. Codec: We encode and decode the audio with different codecs, including MP3, G.722, OGG, AAC, OPUS, Vorbis, a-law, and μ-law. For each codec, we randomly sample a target bitrate from a pre-defined range.

6. Audio shuffling: We divide a spoof audio sample into small segments of 0.1 seconds each, and shuffle them. This method teaches the model speech

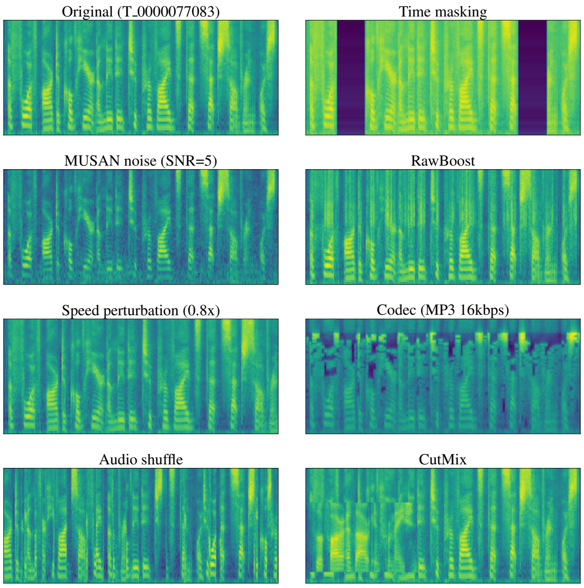

Figure 2: *Visualization of different augmentation techniques on the mel-spectrogram. Some augmentations are exaggerated for better visualization.*

consistency without explicit supervision. This is motivated by the fact that shuffled audio sounds gibberish to humans, but appears indistinguishable in its mel-spectrogram (see Figure 2).

Mel-spectrogram augmentation. We apply CutMix [21] to the mel-spectrograms of the spoof samples. This technique increases deepfake diversity, allowing a single sample to contain multiple attacks. We also experimented with SpecAugment [22] but it did not yield improvements.

Vocoder synthesis. Inspired by [6], we generate new spoof data using a variety of traditional and neural vocoders. Since most modern voice synthesis systems rely on vocoders as their last step to convert spectrograms to audio waveforms, the ability to distinguish vocoder artifacts enhances the model's performance. We select the following vocoders:

1. Traditional vocoders (not learned): Griffin-Lim [23] and WORLD [24]. We directly apply these vocoders on bona-fide samples.

2. Neural vocoders (learned): HiFiGAN [25] and ParallelWaveGAN [26]. We train these vocoders from scratch on bona-fide data and infer them on spectrogram of spoof data. We use the implementation from Coqui TTS[1].

Label ambiguity. Certain augmentations are applied exclusively to spoof samples to avoid label ambiguity. In simple terms, augmented bona-fide samples can no longer be considered genuine, such as those altered by audio shuffling and mel-spectrogram CutMix. Additionally, neural vocoders may produce audio that is too similar to the real data that they are trained on (possibly a case of overfitting), thus confusing the deepfake detection model. By training neural vocoders on the bona-fide subset and inferring on the deepfake subset, we ensure that the generated audio is indeed spoof.

Online and Offline augmentation. Depending on the processing speed of each method, we apply augmentations *online* (on-the-fly during traning) or *offline* (preprocess before training). Online processing allows infinite generation of augmented data, while the offline counterpart only produces a single version of altered samples. Only speed perturbation, audio shuffling, and vocoder synthesis are done offline due to their heavy computational requirements.

2.1.4. Post-processing

Test-time augmentation (TTA). TTA is a popular method to boost system performance in many computer vision tasks [27]. We adapt this technique for speech deepfake detection. For each audio sample, we make several random four-second crops, and score them independently. The crops may overlap. The final score for the sample is obtained by averaging the individual scores.

2.1.5. Ensembling

Model ensembling is a common technique to improve a system's generalization on unseen data. To this end, we train several models with different input features, data augmentations, and training hyperparameters, and select the 10 best performers. To combine the model scores, we experiment with two methods: linear regression and score averaging.

We only consider linear regression as a learning-based fusion approach to avoid overfitting. However, even with such a simple model, we consistently observe degraded performance on the progress set when training the fusion model on the training or development set. We postulate a domain gap between various splits of the ASVspoof 5 dataset. Consequently, we use a simple weighted average of individual model scores as our final system prediction.

2.2. Track 2: Spoofing-robust Automatic Speaker Verification (closed condition)

Our SASV system consists of two sub-systems: deepfake detection and speaker verification. Since the competition permits the use of the VoxCeleb2 [28] in track 2, we considered fine-tuning existing speaker embedding models

[1] `https://github.com/coqui-ai/TTS`

pre-trained on VoxCeleb2 [2], in addition to training models from scratch.

2.2.1. Deepfake detection sub-system

We reuse the best performing models from track 1 as deepfake detection models for track 2. Additionally, we fine-tune ResNet-152 and ResNet-293 pre-trained on VoxCeleb2 with a speaker verification objective for deepfake detection. The fine-tuning procedure closely follows the methods described in Subsection 2.1. The large and diverse VoxCeleb2 corpus used in the pre-training stage helped these models learn discriminative speech features, enabling them to generalize better than models trained solely on ASVspoof 5.

In total, the deepfake detection of track 2 utilizes four ResNet-34 models trained from scratch in track 1, one ResNet-152 and one ResNet-293 fine-tuned from a VoxCeleb2 checkpoint.

2.2.2. ASV sub-system

We experimented with domain adaptation for the pre-trained ResNet models. This was done by fine-tuning them on ASVspoof 5 with the objective of speaker verification (i.e., CosFace [29]). However, fine-tuning resulted in worse ASV performance on the development set. We suspect that the lack of speaker diversity (400 unique speakers in ASVspoof 5 training set compared to 5,994 unique speakers in VoxCeleb2) accounts for the drop in accuracy. Consequently, we directly use the VoxCeleb2 pre-trained ResNet-221 and ResNet-293 for our speaker verification sub-system.

2.2.3. SASV score fusion

To obtain a single deepfake detection score, also called countermeasure (CM) score in the competition, we average scores from individual models. We do the same procedure to obtain a single ASV score.

Directly taking the average of ASV and CM scores is impractical due to their different range of values. ASV score varies from -1 to 1 as it is the cosine similarity between the embeddings of the speaker and trial utterance, while CM score, being the logit of bona-fide probability, has an unbounded range of $(-\infty, \infty)$.

To tackle this problem, we model SASV score to be a linear combination of ASV and CM scores. Since there are only two learnable weights, the risk of overfitting is minimal. COBYQA algorithm [30] is used to learn the linear weights while minimizing *min a-DCF* on the development set.

3. Experiments

3.1. Datasets

The ASVspoof 5 database [1] is built based on the MLS English dataset [31]. The training split consists of 18,797 bona-fide and 163,560 spoof samples, coming from eight different attack types, while the development split contains 31,334 genuine and 109,616 deepfake samples, aggregated across eight voice synthesis systems. There is no overlap of attack types across different splits.

For developing our deepfake detection system, we use the ASVspoof 5 training split for training and the development split for validation. The progress split is further used for the final model selection, based on the competition's online evaluation during the progress phase. The primary metric for each track (i.e., *minDCF* and *min a-DCF* for track 1 and track 2 respectively) is used for ranking the models.

3.2. Training hyperparameters

We train our models with AdamW optimizer [32], learning rate $3.0e^{-4}$, weight decay $1.0e^{-2}$, binary cross entropy loss, batch size 64. Since there are significantly more spoof samples than bona-fide ones (roughly 9:1), we address class imbalance by upsampling genuine instances (i.e., making them appear five times more frequently in the training set). For fine-tuning pre-trained models, we use a lower learning rate of $3.0e^{-5}$ and weight decay of $1.0e^{-4}$.

Apart from the common training configuration above, we apply different training hyperparameters to encourage diversity in model ensembling. Firstly, we create mel-spectrograms with a variety of parameters (see Table 1). In addition, we range the number of training steps from 200,000 to 500,000 (effectively 50-120 epochs). Finally, we use cosine decay [33] and constant learning schedule, both of which have linear warm-up from zero. For the latter schedule, we additionally apply Exponential Moving Average (EMA) of model weights to produce checkpoints [34].

3.3. Results

3.3.1. Track 1: Speech deepfake detection (closed condition)

Single model comparisons. Table 2 shows our best models on the progress set and their main training hyperparameters. X1-X10 are ResNet-34 models trained from scratch for track 1, while Y1 and Y2 are fine-tuned ResNet-152 and ResNet-293 for track 2. Comparing the X models, we observe that metrics on the development set do not correlate strongly with progress set results. For

[2] `https://github.com/wenet-e2e/wespeaker/blob/master/docs/pretrained.md`

Table 2: *Performance of our deepfake detection models. X1-X10 are used in track 1 and 2, while Y1-Y2 are exclusively used in track 2. Only the main differences between models are highlighted. For X1-X10, the* ***best*** *and the second best results are stylized.*

		Data augmentation			Training params.		Progress set		Development set	
Model	Arch.	TimeMask	Noise[†]	Speed	n_mels	n_steps	minDCF	EER (%)	minDCF	EER (%)
X1	R34		N		120	200k	0.0741	2.67	0.0907	3.88
X2	R34		N, M	✓	120	480k	0.0678	2.34	0.1140	4.57
X3	R34	✓	N		120	200k	0.0754	2.72	0.1057	4.26
X4	R34	✓	N, M	✓	120	320k	0.0743	2.60	0.0859	3.26
X5	R34	✓	N, M, S		128	200k	0.0623	2.28	0.1008	4.20
X6	R34	✓	N, M, S		128	280k	0.0728	2.70	0.1232	5.13
X7	R34	✓	N, M, S	✓	120	400k	0.0757	2.62	0.0939	3.66
X8	R34	✓	N, M, S	✓	128	220k	**0.0569**	**2.20**	0.1068	4.49
X9	R34	✓	N, M, S	✓	160	200k	0.0614	2.21	0.0971	4.16
X10[*]	R34	✓	N, M, S	✓	160	200k	0.0739	2.62	**0.0795**	**3.23**
Y1	R152[**]	✓	N, M, S	✓	80	20k	0.0561	2.02	0.0331	1.35
Y2	R293[**]	✓	N, M, S		80	25k	0.0582	2.10	0.0831	3.43

[†] *Background noise: [N]oise and [M]usic from MUSAN, [S]peech from ASVspoof 5 bona-fide.*
[*] *X10 is trained with SGD optimizer.*
[**] *These models are fine-tuned from VoxCeleb2 checkpoints.*

example, the best model on the progress set (i.e., X8) performs significantly worse than the best model on the development set (i.e., X10). This discrepancy may be due to the lack of generalization to unseen data. Therefore, we rely on heavy augmentations and model ensembling for robust predictions on the evaluation set.

Model ensembling. Based on the results from Table 2, we assign a weight of 0.125 for the top four models on the progress set (i.e., X8, X9, X5, X2), and a weight of 0.083 to the rest (i.e., X6, X10, X1, X4, X3, X7). This strategy ensures that the better-performing models contribute slightly more to the final predictions.

Table 3: *Deepfake detection results on the evaluation set*

Model	**minDCF**	**EER (%)**
ASVspoof 5 baseline 1	0.8270	36.04
ASVspoof 5 baseline 2	0.7110	29.12
Ours	**0.2660**	**9.18**

Evaluation set results. Table 3 shows the results of our model ensemble for track 1 (i.e., deepfake detection). We achieve *minDCF* of 0.2660 and *EER* of 9.18%, around three times better than the competition baselines, showing the effectiveness of our data augmentation strategy and ensembling.

3.3.2. Track 2: Spoofing-robust Automatic Speaker Verification (closed condition)

Deepfake detection sub-system. Comparing the fine-tuned models (i.e., Y1 and Y2) with the best trained-from-scratch ResNet-34 (i.e., X8), although they are comparable on the progress set, Y1 achieves significantly lower *minDCF* on the development set. Unsurprisingly, fine-tuning pre-trained models is beneficial. To obtain the final CM score, we average the scores of Y1, Y2, and the top four ResNet-34 models (i.e., X8, X9, X5, X2).

Table 4: *SASV results on the evaluation set*

Model	**min a-DCF**
ASVspoof5 baseline	0.6810
Ours	**0.3173**

Evaluation set results. Table 4 shows our SASV system achieves *min a-DCF* of 0.3173, which is more than two times lower than the competition baseline.

4. Conclusion

This paper presents ParallelChain Lab's submissions for the ASVspoof 5 challenge, focusing on deepfake detection and SASV systems. Our deepfake detection utilizes a modified ResNet architecture and a plethora of data augmentation techniques, while the SASV system combines pre-trained models with our deepfake detection system. Our final results, with a *minDCF* of 0.2660 for deepfake detection and a *min a-DCF* of 0.3173 for SASV in the closed condition, outperform the ASVSpoof 5 baselines and demonstrate their effectiveness against sophisticated deepfake attacks.

5. References

[1] Xin Wang, Héctor Delgado, Hemlata Tak, Jee-weon Jung, Hye-jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi, "ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale," in *ASVspoof Workshop 2024 (accepted)*, 2024.

[2] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov, "Grad-TTS: A Diffusion Probabilistic Model for Text-to-Speech," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 8599–8608, PMLR.

[3] Jaehyeon Kim, Jungil Kong, and Juhee Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 5530–5540, PMLR.

[4] Sang gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon, "BigVGAN: A Universal Neural Vocoder with Large-Scale Training," in *The Eleventh International Conference on Learning Representations*, 2023.

[5] Michele Panariello, Wanying Ge, Hemlata Tak, Massimiliano Todisco, and Nicholas Evans, "Malafide: a novel adversarial convolutive noise attack against deepfake and spoofing detection systems," in *Proc. INTERSPEECH 2023*, 2023, pp. 2868–2872.

[6] Haochen Wu, Zhuhai Li, Luzhen Xu, Zhentao Zhang, Wenting Zhao, Bin Gu, Yang Ai, Yexin Lu, Jie Zhang, Zhenhua Ling, et al., "The USTC-NERCSLIP System for the track 1.2 of Audio Deepfake Detection (ADD 2023) Challenge.," in *DADA@ IJCAI*, 2023, pp. 119–124.

[7] Alexander Alenin, Nikita Torgashov, Anton Okhotnikov, Rostislav Makarov, and Ivan Yakovlev, "A Subnetwork Approach for Spoofing Aware Speaker Verification," in *Proc. Interspeech 2022*, 2022, pp. 2888–2892.

[8] Anton Tomilov, Aleksei Svishchev, Marina Volkova, Artem Chirkovskiy, Alexander Kondratev, and Galina Lavrentyeva, "STC Antispoofing Systems for the ASVspoof2021 Challenge," in *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021, pp. 61–67.

[9] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] Daniel Garcia-Romero, Greg Sell, and Alan Mccree, "MagNetO: X-vector Magnitude Estimation Network plus Offset for Improved Speaker Recognition," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2020)*, 2020, pp. 1–8.

[12] Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[14] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, "A ConvNet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11976–11986.

[15] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer, "How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers," *Transactions on Machine Learning Research*, 2022.

[16] Hugo Touvron, Matthieu Cord, and Hervé Jégou, "DeiT III: Revenge of the ViT," in *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow,

Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, Eds., Cham, 2022, pp. 516–533, Springer Nature Switzerland.

[17] Terrance DeVries and Graham W Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[18] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[19] Hemlata Tak, Madhu Kamble, Jose Patino, Massimiliano Todisco, and Nicholas Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6382–6386.

[20] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech 2015*, 2015, pp. 3586–3589.

[21] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[22] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[23] D. Griffin and Jae Lim, "Signal estimation from modified short-time Fourier transform," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983, vol. 8, pp. 804–807.

[24] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[25] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 17022–17033, Curran Associates, Inc.

[26] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[27] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag, "Better aggregation in test-time augmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1214–1223.

[28] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *INTERSPEECH*, 2018.

[29] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "CosFace: Large Margin Cosine Loss for Deep Face Recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[30] T. M. Ragonneau, *Model-Based Derivative-Free Optimization Methods and Software*, Ph.D. thesis, Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China, 2022.

[31] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert, "MLS: A Large-Scale Multilingual Dataset for Speech Research," in *Proc. Interspeech 2020*, 2020, pp. 2757–2761.

[32] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[33] Ilya Loshchilov and Frank Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in *International Conference on Learning Representations*, 2017.

[34] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," in *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, Ricardo Silva, Amir Globerson, and Amir Globerson, Eds. 2018, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pp. 876–885, Association For Uncertainty in Artificial Intelligence (AUAI).