



Model Compression for DNN-based Speaker Verification Using Weight Quantization

Jingyu Li¹, Wei Liu¹, Zhaoyang Zhang¹, Jiong Wang², Tan Lee¹

¹Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong

²School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China

{lijingyu0125, louislau.1129, zhaoyangzhang}@link.cuhk.edu.hk,
jiongwang@link.cuhk.edu.cn, tanlee@ee.cuhk.edu.hk

Abstract

DNN-based speaker verification (SV) models demonstrate significant performance at relatively high computation costs. Model compression can be applied to reduce the model size for lower resource consumption. The present study exploits weight quantization to compress two widely-used SV models, namely ECAPA-TDNN and ResNet. Experimental results on VoxCeleb show that weight quantization is effective for compressing SV models. The model size can be reduced multiple times without noticeable degradation in performance. Compression of ResNet shows more robust results than ECAPA-TDNN with lower-bitwidth quantization. Analysis of the layer weights suggests that the smooth weight distribution of ResNet may be related to its better robustness. The generalization ability of the quantized model is validated via a language-mismatched SV task. Furthermore, analysis by information probing reveals that the quantized models can retain most of the speaker-relevant knowledge learned by the original models.

Index Terms: speaker verification, weight quantization, model compression, cross languages, information probe

1. Introduction

Speaker verification (SV) is a biometric authentication process verifying whether a spoken utterance is from the claimed speaker [1, 2]. Deep neural network (DNN) models have shown great success in SV [3–6] and outperformed conventional methods, e.g., I-vector [7]. Despite significant performance attained with these DNN models, it comes at the cost of higher computation complexity. SV systems are expected to be run locally on devices without the Internet [8]. In this way, users' private data are kept safe without being exposed outside the device. The data transmission delay can also be avoided. However, the demanding computation cost of current DNN-based SV systems makes them not handy to deploy on resource-constrained devices.

To reduce resource consumption, model compression is needed. The performance achieved by the compressed model should remain comparable to the original model. Various compression methods, such as weight quantization [9–11], pruning [12, 13], and knowledge distillation [14], have been studied on different tasks [15, 16]. The present study focuses on weight quantization for DNN-based speaker embedding extraction models. The default data bitwidth used in existing deep learning frameworks is 32, e.g., PyTorch [17] and TensorFlow [18]. The core operation in weight quantization is mapping full-bitwidth model weights onto a lower-bitwidth representation, which reduces the model size and enables faster processing.

To the best of our knowledge, weight quantization has not been investigated on state-of-the-art DNN SV models. ECAPA-TDNN [5] and ResNet [4, 19] are two widely-used well-

performing models for speaker embedding extraction. Many SV models have been developed based on them [6, 20, 21]. In this study, we evaluate the effect of weight quantization on these two models. The experiments are carried out on the VoxCeleb datasets [4, 22, 23], with two common learnable quantization methods: uniform quantization [24] and Powers-of-Two quantization [25]. Our preliminary experiments show that weight quantization is effective for SV model compression. With multiple times smaller sizes than the original models, the performance of quantized ECAPA and ResNet declines slightly. ResNet shows more robust results than ECAPA, especially in 4-bit quantization. To investigate the cause of such a difference, we analyze the layer-wise weight distribution of these two models. The highly zero-concentrated and sharp weight distribution of ECAPA is observed, which may explain its less robustness than ResNet using lower-bitwidth quantization. As reducing the model size, the learning ability of the quantized model is decreased compared to the original counterpart. The poorer learning ability may affect the models' generalization capability and extracted knowledge. We evaluate the generalization of quantized models using a language-mismatched SV evaluation, which is performed on CN-Celeb [26] without model fine-tuning. Inspired by [27], four information probe tasks are applied to the extracted speaker embeddings to investigate the model's learned knowledge before and after quantization.

The paper is organized as follows: Section 2 introduces the details of the quantization methods. The experimental setting and results are given in Sections 3 and 4, respectively. Finally, we give a brief conclusion in Section 5.

2. Quantization

A trainable weight in a layer is denoted as $\mathcal{W} \in \mathbb{R}^N$. In a 2D convolutional layer (CNN), $N = k \times k \times c_1 \times c_2$. In a fully-connected layer (FC), $N = c_1 \times c_2$. \mathcal{W} takes continuous values. The quantization of \mathcal{W} is expressed as

$$\widehat{\mathcal{W}} = \Gamma([\mathcal{W}, \alpha])_{\mathbf{q}(\alpha, b)}. \quad (1)$$

\mathcal{W} is first normalized by its mean and standard deviation so that it can be represented in a symmetric range. They are clipped into a range of $[-\alpha, \alpha]$ by a function denoted as $[\cdot, \alpha]$, where $\alpha (\geq 0)$ is a learnable parameter. $\Gamma(\cdot)_{\mathbf{q}}$ represents a projection function, mapping continuous values into a set of discrete values $\mathbf{q} = [\pm q_1, \pm q_2, \dots, \pm q_{\frac{n}{2}}]$. $n = 2^b$ is the number of quantization levels. b is the bitwidth and is usually predetermined. The quantization level, i.e., the values of $[q_1, q_2, \dots]$, is another critical factor determining the performance of the quantized models. Uniform and Powers-of-Two quantization are widely used [24, 25, 28–30] for constructing the quantization

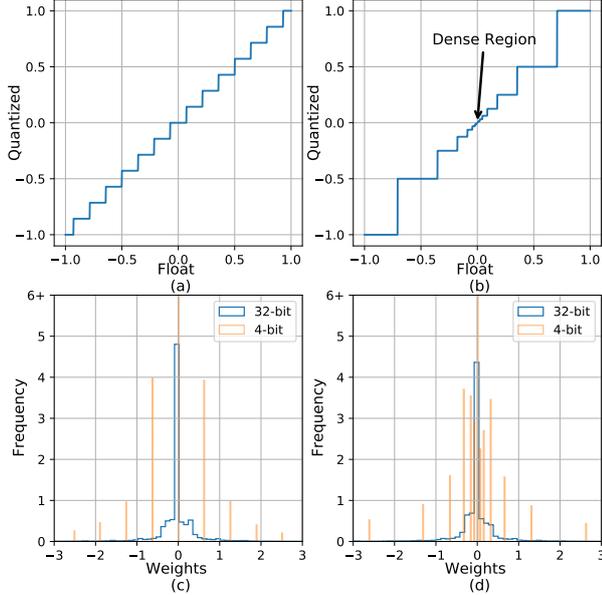


Figure 1: The quantized values of (a) uniform and (b) PoT quantization versus original float32 values. The weights distribution of 32-bit weights and 4-bit quantized by (c) uniform, (d) PoT.

level. They are friendly to both software and hardware implementations.

2.1. Uniform quantization

The quantization levels of uniform quantization are represented as

$$\mathbf{q}(\alpha, b) = [0, \frac{\pm 1}{2^{b-1}-1}, \frac{\pm 2}{2^{b-1}-1}, \dots, \pm 1] \times \alpha. \quad (2)$$

Uniform quantization generates the discrete values evenly over the interval $[-1, 1]$, as illustrated in Fig. 1(a). The length of each quantization step is constant, i.e., $\frac{1}{2^{b-1}-1}$. A smaller step length gives denser discrete levels to distinguish different values. However, the distribution of weights is usually non-uniform [12]. For instance, a layer’s weight distribution from a trained model is plotted by the blue curve in Fig. 1(c). There are more weights concentrated around the center and fewer weights around the boundaries. The orange line in Fig. 1(c) shows the distribution of uniform-quantized weights. The minor numerical differences between weights are erased if they lie in the same quantization level. Thus a large proportion of the weights around the center will be assigned to an identical value, which may dramatically decrease the diversity of weights.

2.2. Powers-of-Two quantization

The quantization levels of Powers-of-Two (PoT) quantization are represented as

$$\mathbf{q}(\alpha, b) = [0, \pm 2^{-2^{b-1}+2}, \dots, \pm 2^{-1}, \pm 1] \times \alpha. \quad (3)$$

PoT provides non-identical discrete levels, unlike uniform quantization, as shown in Fig. 1(b). The quantization levels around the center are denser. The denser levels provide a more refined quantization description of weight parameters, as shown by the orange lines in Fig. 1(d). Sparser quantization levels are proposed near the boundary. If only a few weights lie around the boundary, as in Fig. 1(d), discretizing the weights may not significantly affect the quantization error ($\|\mathcal{W} - \widehat{\mathcal{W}}\|^2$).

2.3. Parameter update

The quantization error ($\|\mathcal{W} - \widehat{\mathcal{W}}\|^2$) depends heavily on the value of α . A large α covers a wide range of weight values, and the quantization levels are distributed sparsely. A smaller value of α gives denser discrete levels, and more weights will be clipped to $-\alpha$ or α , resulting in higher quantization errors. The distributions of weights in different layers are not identical, and a constant clipping value may not be optimal for quantizing \mathcal{W} . In this paper, α is made learnable as in [9]. Each layer is assigned an α , and the value of α is updated during model training by backward propagation. $[\cdot, \alpha]$ and $\Gamma(\cdot)_q$ are two discrete operations, where the gradients cannot be calculated directly. Following [10], the gradients for α and \mathcal{W} are estimated by Straight-Through Estimator (STE) [31] as

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \frac{\partial \mathcal{L}}{\partial \widehat{\mathcal{W}}} \frac{\partial \widehat{\mathcal{W}}}{\partial \alpha}, \quad \frac{\partial \widehat{\mathcal{W}}}{\partial \alpha} = \begin{cases} \text{sign}(\mathcal{W}) & \text{if } |\mathcal{W}| > \alpha \\ \frac{\widehat{\mathcal{W}}}{\alpha} - \frac{\mathcal{W}}{\alpha} & \text{if } |\mathcal{W}| \leq \alpha \end{cases}, \quad (4)$$

$$\frac{\partial \mathcal{L}}{\partial \mathcal{W}} = \frac{\partial \mathcal{L}}{\partial \widehat{\mathcal{W}}} \frac{\partial \widehat{\mathcal{W}}}{\partial \mathcal{W}}, \quad \frac{\partial \widehat{\mathcal{W}}}{\partial \mathcal{W}} = 1. \quad (5)$$

where \mathcal{L} is the training loss. In each training step, the full-precision \mathcal{W} is first converted into $\widehat{\mathcal{W}}$ with discrete values. $\widehat{\mathcal{W}}$ is used for the computational process, such as convolution or matrix multiplication. Thus the calculation of $\frac{\partial \mathcal{L}}{\partial \widehat{\mathcal{W}}}$ can be performed as the conventional gradient computation. After training, the \mathcal{W} is quantized to $\widehat{\mathcal{W}}$ and stored. In the inference, the model with weight $\widehat{\mathcal{W}}$ is used to perform the forward process, which requires less device memory and computation cost.

3. Experimental Set-up

3.1. Datasets

The datasets utilized in the experiments are VoxCeleb 1&2 (denoted as Vox.1 & Vox.2) [4, 22, 23]. Speech data in VoxCeleb are mostly in English. The development set of Vox.2 consists of 5,994 speakers and is used for model training. Vox.1 is used to evaluate the performance of models. Three test sets are constructed using data from Vox.1 and denoted as Original, Easy, and Hard sets, respectively.

In addition, the test set of CN-Celeb [26] is used to evaluate the models’ performance in another language. CN-Celeb is a large-scale Chinese dataset consisting of about 1,000 speakers.

3.2. Model training and evaluation

Experiments are performed on ECAPA-TDNN with 512 and 1024 channels, ResNet34 with 32 channels. The training process has two stages. The model is first trained in full precision, i.e., 32-bit in PyTorch. The model is trained to predict the input speaker identity using the AAM-softmax [32] as the classification loss. The margin and scale of AAM-softmax are set to be 0.2 and 30. The Adam optimizer [33] with a weight decay of $2e-5$ is utilized. The learning rate is initialized as 0.001 and decayed by a ratio of 0.1 at the 20th and 32nd epoch, respectively. The whole training process consists of 40 epochs. The batch size is set to 128. A 2-second long segment is cropped from each input speech randomly. The model input is 64-dimensional *log* Mel-filterbanks (FBanks) transformed from the raw waveform of each segment using the hamming window. The window size is 25ms, and the hop length equals 10ms. Background sound addition [34] and audio reverberation [35] are utilized for data augmentation in training.

In the second stage, the quantized model is fine-tuned from the full-precision model for 20 epochs. The learning rate is de-

Table 1: Performances of the models. Vox.O, Vox.E, Vox.H are short for the Original, Easy and Hard test sets of Vox.I

Model	Parameters (M)	MACs (Giga)	Compression	Bitwidth (bits)	Size (MBytes)	EER(%)		
						Vox.O	Vox.E	Vox.H
ECAPA-TDNN-512	5.95	0.96	-	32	23.87	1.07	1.28	2.43
			Uni	8	6.09	1.20	1.46	2.75
			PoT	8	6.09	1.44	1.69	3.11
			Uni	4	3.13	1.56	1.78	3.29
ECAPA-TDNN-1024	14.38	2.56	-	32	57.61	0.92	1.15	2.28
			Uni	8	14.6	1.26	1.45	2.73
			PoT	8	14.6	1.41	1.59	2.99
			Uni	4	27.63	1.10	1.15	2.08
ResNet34	6.9	3.67	-	32	27.63	1.10	1.15	2.08
			Uni	8	6.96	1.05	1.11	2.02
			PoT	8	6.96	1.28	1.34	2.41
			Uni	4	3.52	1.22	1.29	2.32
ECAPA-TDNN-256	3.11	0.43	-	32	12.5	1.43	1.59	2.87
			distill(512)	32	12.5	1.29	1.53	2.88
			distill(1024)	32	12.5	1.30	1.53	2.85
ResNet18	4.37	1.78	-	32	17.51	1.45	1.44	2.53
			distill	32	17.51	1.33	1.39	2.48

Table 2: Performances of the models on CN-Celeb

Model	Compression	EER(%)
ECAPA-TDNN-512	-	17.61
	Uni,8-bit	17.50
	Uni,4-bit	16.12
ResNet34	-	11.85
	Uni,8-bit	11.75
	Uni,4-bit	11.93
ECAPA-TDNN-256	distill(512)	18.13
ResNet18	distill	12.32

cayed at the 10th and 16th epochs. Reducing the model size decreases the learning ability. Therefore, data augmentation is not used for training the quantized models. The rest of the training settings remain the same as the first stage.

For evaluation, each utterance is divided into 4-second long segments, with a 1-second overlap between two adjacent segments. The cosine similarity values between the test utterance segments and the enrollment segments are averaged as the verification score. Adaptive s-norm (AS-norm) [36] is applied to calibrate the scores. The performance is reported in terms of equal error rate (EER).

4. Results and Analysis

4.1. Learnable quantization

The results with different models on the three test sets of Vox.I are shown in Table 1. The computation complexity is measured in terms of the required number of multiply-accumulate operations (MACs). 8-bit quantization reduces the model size by approximately 4 times. The performance of ECAPA-TDNN-512 relatively declines by 15% on all test sets using uniform quantization. ECAPA-1024 outperforms ECAPA-512 with full precision but gives close results with 8-bit uniform quantiza-

tion. ResNet gives more robust results than ECAPA. The performance of ResNet34 is almost not affected with uniform quantization. PoT shows slightly poorer results than uniform on all models and test sets. The EER of PoT’s output is 15% to 20% higher than the uniform’s.

To evaluate the model performance under higher compression rate, 4-bit quantization is applied to quantize ECAPA-512 and ResNet34. The quantization reduces the models’ size by 8 times. The EERs of 4-bit quantized ECAPA-512 are increased by about 50% compared with the full-precision model. The performance degradation of ResNet34 is 10% to 20%, significantly milder than that of ECAPA-512. ResNet34 has a model size similar to ECAPA-512, but its MACs are much larger than ECAPA’s, which may contribute to its minor performance degradation.

We compare the quantized models with models compressed by knowledge distillation. The distillation is performed on ECAPA-TDNN-256 and ResNet18. ECAPA-256 is constructed by using 256 convolution channels. As shown in Table 1, the distilled models give around 10% lower EER than the model trained from scratch on all test sets. The 8-bit quantized ECAPA structure uses a half size and gives a close performance to the distilled ECAPA-256. The quantized ResNet uses a smaller size and achieves superior performance to its distilled variant.

4.2. Weight analysis

The above results have shown that uniform quantization is more suitable than PoT on these two SV models. Here, 8-bit uniform-quantized models are utilized as representatives in weight analysis. Several layers’ weight distribution and statistics information are given in Fig. 2. The last convolution layers of Block 1 to 3 of ResNet34 are given in (a)-(c), respectively. For ECAPA structures, the last convolution layers of Block 1 to 3 are shown. The weights are clipped by the corresponding α and scaled into [-1, 1] for better visualization. “Error” denotes the average quantization error of the corresponding layer.

Among the three models, the weight distributions of

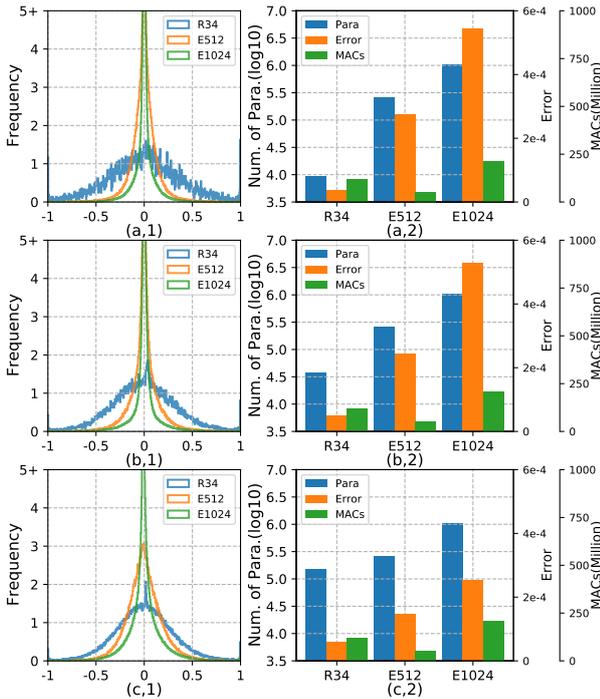


Figure 2: The weight distributions are shown in (a,1) to (c,1). R34 stands for ResNet34, E512 for ECAPA-512, and E1024 for ECAPA-1024. (a,2) to (c,2) gives the number of parameters, averaged quantization error, and MACs of layers.

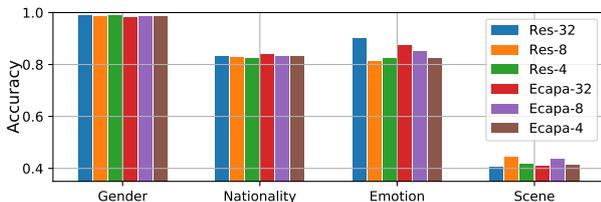


Figure 3: Accuracy of probing tasks for models with different bitwidths. Res-32 is short for ResNet34 with 32-bit, Ecapa-8 for ECAPA-512 with 8-bit, etc.

the ECAPA networks are highly concentrated, and those of ResNet34 show smooth bell shapes. The spiky-shape distribution may lead to larger quantization error in ECAPA when uniform quantization is applied, as explained in Section 2.1. It can also be observed that the average quantization error is positively correlated with the number of parameters in a layer. Thus quantizing layers with a large number of parameters is challenging. The parameters are evenly distributed among layers in the same block of ResNet34, whereas the parameters concentrate densely on a block’s first and last convolution layers in the ECAPA structure. The dense layers of ECAPA may explain its larger quantization performance decline compared to ResNet. This suggests a good strategy for model design is to avoid concentrating a large number of parameters on several layers.

4.3. Evaluation on CN-Celeb

The models trained on VoxCeleb are tested on CN-Celeb without fine-tuning to evaluate their generalization ability in a different language. The results are summarized in Table 2. The 8-bit quantized model shows superior performance to the full-

precision model on ECAPA-512, and the 4-bit model gives the lowest EER. ResNet34 outperforms ECAPA-512 with or without the quantization, indicating it has a better generalization ability. The 8-bit quantized ResNet34 also outperforms its original model. There is a slight performance drop on the 4-bit ResNet34. [10] found that quantization may contribute to weight regularization, which can alleviate the overfitting problem of the full-precision model. This effect may increase the model generalization in cross-language SV. Performance decline is observed in the distillation models, showing that their generalization ability is worse than quantized models’.

4.4. Information probe

Information probing is conducted on the speaker embeddings extracted by the original and quantized ResNet34 and ECAPA-512. Four classification tasks are applied, i.e., gender, nationality, emotion, and scene. Gender and nationality are two speaker-dependent information. They are shown to be highly relevant to the speakers’ variation [37]. The gender and nationality classification tasks are trained on Vox.2 and evaluated on Vox.1. Emotions are speaker-independent. Different emotions are varied from various pitches, loudness, etc [38]. Previous works [39, 40] have shown that speaker embeddings contain significant emotional information. SAVEE [41] is used for emotion classification. Scene information is expected to be speaker-irrelevant. Scene classification is performed on TAU Urban Acoustic Scenes 2020 Mobile [42]. A multi-layer perceptron with two FC layers and ReLU activation is utilized as the classifier for all tasks.

The results are summarized in Fig. 3. The accuracy of gender classification is over 98% for all models, showing that the quantization does not lose this information. Nationality and emotion are two factors related to the speaking styles of the input speech. The classifiers achieve accuracy higher than 80% on both tasks for all models, while the accuracy is decreased slightly in the quantized models compared to the full-size models. The results indicate that the embeddings capture sufficient knowledge to distinguish different speaking styles, and a small fraction of styling information is lost in the quantization. The scene classification accuracy is observably lower than other tasks. However, the quantized models show higher accuracy in this task than the original models. The scene sounds may have biases for different speakers. The quantized models can rely on biased environment information for distinguishing speakers, but this effect will cause more errors in SV under noisy environments. The full-size model has a stronger learning ability, which may depress the disturbance from scene information.

5. Conclusions

The method of weight quantization is investigated on compressing DNN-based SV models. The experiments are carried out on two commonly used SV model structures: ECAPA and ResNet. The experiments show that the SV model size can be compressed by multiple times with a slight performance decline. The ResNet gives robust performance after quantization. The performance of ECAPA is affected more by low-bitwidth quantization than ResNet. The quantized models perform comparable or better than the full-precision models in a cross-language SV evaluation, indicating that the model’s generalization ability is preserved after model quantization. The quantized models retain speaker-relevant knowledge and obtain additional information with respect to the environmental sound.

6. References

- [1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.
- [2] Z. Wu, N. Evans, T. Kinnunen *et al.*, "Spoofing and countermeasures for speaker verification: A survey," *speech communication*, vol. 66, pp. 130–153, 2015.
- [3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.
- [4] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Interspeech*, 2017, pp. 2616–2620.
- [5] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech*, 2020, pp. 3830–3834.
- [6] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating Frequency Translational Invariance in TDNNs and Frequency Positional Information in 2D ResNets to Enhance Speaker Verification," in *Interspeech*, 2021, pp. 2302–2306.
- [7] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.
- [8] J. Kim, S. Chang, and N. Kwak, "PQK: model compression via pruning, quantization, and knowledge distillation," in *Interspeech*, 2021, pp. 4568–4572.
- [9] J. Choi, Z. Wang, S. Venkataramani *et al.*, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.
- [10] Y. Li, X. Dong, and W. Wang, "Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks," in *ICLR*, 2020.
- [11] Z. Zhang, W. Shao, J. Gu, X. Wang, and P. Luo, "Differentiable dynamic quantization with mixed precision and adaptive resolution," in *ICML*, 2021, pp. 12 546–12 556.
- [12] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *ICLR*, 2016.
- [13] H. Li, A. Kadav, I. Durdanovic *et al.*, "Pruning filters for efficient convnets," in *ICLR*, 2017.
- [14] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *ICLR*, 2018.
- [15] A. D. A. Salvi and R. C. Barros, "Model compression in object detection," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
- [16] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, 2018.
- [17] A. Paszke, S. Gross, F. Massa *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035.
- [18] M. Abadi, A. Agarwal, P. Barham *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [19] K. He, X. Zhang, S. Ren *et al.*, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [20] T. Liu, R. K. Das, K. A. Lee, and H. Li, "MFA: TDNN with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *ICASSP*, 2022, pp. 7517–7521.
- [21] J. Li and T. Lee, "Text-independent speaker verification with dual attention network," in *Interspeech*, 2020, pp. 956–960.
- [22] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018, pp. 1086–1090.
- [23] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Science and Language*, 2019.
- [24] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave gaussian quantization," in *CVPR*, 2017, pp. 5918–5926.
- [25] A. Zhou, A. Yao, Y. Guo *et al.*, "Incremental network quantization: Towards lossless cnns with low-precision weights," in *ICLR*, 2017.
- [26] Y. Fan, J. Kang, L. Li *et al.*, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP*, 2020, pp. 7604–7608.
- [27] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 726–733.
- [28] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, "Differentiable soft quantization: Bridging full-precision and low-bit neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4852–4861.
- [29] D. Miyashita, E. H. Lee, and B. Murmann, "Convolutional neural networks using logarithmic data representation," *arXiv preprint arXiv:1603.01025*, 2016.
- [30] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*, 2022, pp. 291–326.
- [31] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
- [32] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
- [34] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.
- [35] T. Ko, V. Peddinti, D. Povey *et al.*, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017, pp. 5220–5224.
- [36] P. Matejka, O. Novotný, O. Plchot *et al.*, "Analysis of score normalization in multilingual speaker recognition," in *Interspeech*, 2017, pp. 1567–1571.
- [37] C. Luu, S. Renals, and P. Bell, "Investigating the contribution of speaker attributes to speaker separability using disentangled speaker representations," in *Interspeech*, 2022, pp. 610–614.
- [38] A. Lausen and K. Hammerschmidt, "Emotion recognition and confidence ratings predicted by vocal stimulus type and prosodic parameters," *Humanities and Social Sciences Communications*, vol. 7, no. 1, pp. 1–17, 2020.
- [39] J. Williams and S. King, "Disentangling style factors from speaker representations," in *Interspeech*, 2019, pp. 3945–3949.
- [40] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "x-vectors meet emotions: A study on dependencies between emotion and speaker recognition," in *ICASSP*, 2020, pp. 7169–7173.
- [41] P. Jackson and S. Haq, "Surrey audio-visual expressed emotion (savee) database," *University of Surrey: Guildford, UK*, 2014.
- [42] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: Generalization across devices and low complexity solutions," in *Proceedings of 5th the Workshop on Detection and Classification of Acoustic Scenes and Events 2020 (DCASE 2020), Tokyo, Japan (full virtual)*, 2020, pp. 56–60.