



Breaking Resource Barriers in Speech Emotion Recognition via Data Distillation

Yi Chang¹, Zhao Ren², Zhonghao Zhao³, Thanh Tam Nguyen⁴, Kun Qian³, Tanja Schultz², Björn W. Schuller^{1,5}

¹GLAM – the Group on Language, Audio, & Music, Imperial College London, United Kingdom

²Cognitive Systems Lab, University of Bremen, Germany

³School of Medical Technology, Beijing Institute of Technology, China

⁴Griffith University, Australia

⁵Chair of Health Informatics, Technical University of Munich, Germany

y.chang20@imperial.ac.uk, zren@uni-bremen.de

Abstract

Speech emotion recognition (SER) plays a crucial role in human-computer interaction. The emergence of edge devices in the Internet of Things (IoT) presents challenges in constructing intricate deep learning models due to constraints in memory and computational resources. Moreover, emotional speech data often contains private information, raising concerns about privacy leakage during the deployment of SER models. To address these challenges, we propose a data distillation framework to facilitate efficient development of SER models in IoT applications using a synthesised, smaller, and distilled dataset. Our experiments demonstrate that the distilled dataset can be effectively utilised to train SER models with fixed initialisation, achieving performances comparable to those developed using the original full emotional speech dataset.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

Speech, a distinctive human capability, facilitates the conveyance of a wide range of emotional nuances [1] (e. g., happiness, sadness, anger, fear, etc.). Identifying emotions from speech is a meaningful and challenging task [2, 3], as internal emotional states are often hidden in speech and difficult for humans to recognise [4]. Over the past decades, speech emotion recognition (SER) aims to automatically identify emotional states from speech signals through signal processing and artificial intelligence approaches. An escalating number of human-computer interaction applications, such as intelligent call centres [5] and computer games [6], are evidencing a burgeoning need for SER.

With the advancement of computing resources, numerous deep learning (DL) methodologies have been successfully applied to the task of SER [7–10], including convolutional neural networks (CNNs) with the input of time-frequency representations [8, 9] and end-to-end models (e. g., wav2vec 2.0 [11], HuBERT [12], and WavLM [13]) fed with raw audio samples. Currently, the integration of DL techniques for SER into the Internet of Things (IoT) has spurred the development of numerous innovative personalised applications, such as smart homes [14], health states monitoring [15], and intelligent transportation systems [16]. In this context, applying DL to IoT frameworks shows two major challenges. First, the constraints of DL models imposed by memory consumption continue to impede their advancement on edge devices, which often have limited onboard resources [17]. Second, considering that speech data encompasses a wide array of personal information [18], recent research findings have demonstrated the potential vulnerability of DL algorithms to privacy breaches [19, 20].

To address the aforementioned two challenges, data distilla-

tion emerges as a promising solution by extracting representative information to form a refined, smaller dataset from a larger one. Specifically, data distillation generates a compact dataset synthesised from the entirety of the original data, enabling that models trained on the smaller dataset perform on par with those on the complete one [21]. Additionally, leveraging synthesised smaller-scale datasets for subsequent model training further diminishes the likelihood of privacy leakage [22]. Data distillation has emerged as a vibrant research area in the field of machine learning with diverse applications (e. g., image classification [23]). However, the application of data distillation remains limited in SER. To the best of our knowledge, this is the first attempt to construct a data distillation framework specifically for SER.

In this paper, we present a data distillation framework to synthesise a compact dataset for SER by constructing a similar model parameters with fewer iterations [23]. The main contributions of this work can be summarised as follows: First, the proposed approach can effectively reduce the dataset size to approximately 15 % of the original one, conserving memory space while maintaining performance. Second, the required training iterations on the distilled dataset are less than those on the original data. Third, our framework has potential in protecting user identification against malicious exploitation in the SER domain. The code will be available online upon the paper’s acceptance.

2. Related Work

Our work addresses memory constraints on edge devices by reducing model parameters. For instance, model pruning can effectively reduce model size and computational complexity by strategically removing some model parameters [24]. Parameter quantisation reduces the storage space of a model by decreasing the number of bits needed for each parameter [25]. Model distillation [26] is a method that transfers knowledge from a complex teacher model to a lightweight student model, making it well-suited for edge computing and IoT applications [26]. Compared to the aforementioned methods for model size reduction, data distillation entails transferring knowledge from a large-scale dataset to a smaller one, enabling models trained on the compact dataset to match the performance of those trained on the original dataset [21]. Data distillation has been widely researched within ML (e. g., continual learning [27], federated learning [28], and neural architecture search [29]). Most existing research focuses on image datasets, like MNIST [30] and CIFAR [31]. Studies [32] and [33] have sought to apply dataset distillation techniques in the domains of speech recognition and question-answering systems, respectively. Inspired by these studies, data distillation [23] is investigated herein to reduce data size for SER in our work.

To protect users’ privacy in the context of IoT, federated

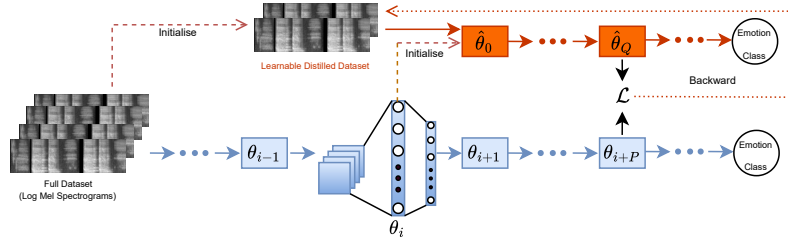


Figure 1: Overview of the proposed data distillation framework for SER. The blue arrows indicate the teacher networks’ training on the original, complete dataset, whereas the red arrows display the student networks’ training on the learnable distilled dataset.

learning aims to keep users’ data on local clients (i. e., edge devices) while only uploading model parameters to the cloud server. This approach helps to safeguard privacy by minimising exposure during client-server communication as well as in the cloud, though it requires the exchange of complex model parameters [22]. Data distillation can protect users’ privacy by generating dataset summaries from few number of data samples [22]. Moreover, data distillation has been shown to be effective in differential privacy [34]. Furthermore, integrating federated learning with data distillation, which involves uploading synthesized data rather than model parameters, not only reduces the communication load between clients and the server but has also been shown to outperform traditional federated learning with model synchronisation [35, 36].

3. Methodology

Based on the original, complete and full SER dataset D_{org} , dataset distillation in this work aims at generating a smaller but effective, synthetic training dataset D_{syn} . As depicted as in Figure 1, log Mel spectrograms extracted from the audio samples in D_{org} are fed into a convolutional neural network (CNN) for emotion classification, where the snapshot of model parameters after each iteration are stored into a *teacher trajectory*. When conducting the dataset distillation, we first initialise the trainable D_{syn} by randomly selecting a certain number of log Mel spectrograms per class (IPC) from D_{org} . Second, a *student trajectory* is initialised with a randomly chosen iteration (i. e., i in Figure 1) from a *teacher trajectory*. Afterwards, the *student trajectory* is trained on the D_{syn} for several iterations (i. e., Q in Figure 1). During the dataset distillation, the values of the log Mel spectrograms are updated according to the distance between θ_{i+P} and $\hat{\theta}_Q$, where P is a pre-defined number of epochs along the *teacher trajectory*. In this way, the knowledge of all emotional information in the full dataset is transferred from the *teacher trajectory* to the generated D_{syn} . In Section 3.1 and 3.2, we introduce the SER models applied for detailed trajectories construction. In Section 3.3, the loss function in the training of D_{syn} is described.

3.1. Teacher Trajectories

In this work, the trajectory means a sequence of a model’s parameters. When training the CNN models on D_{org} , we store the snapshot parameters of CNN models after each epoch as θ_i , where i represents the current iteration number. A *teacher trajectory* is defined as $\Theta = \{\theta_1, \theta_2, \dots, \theta_e\}$, where e denotes the number total iterations. In this work, we generate a set of *teacher trajectories* $\mathcal{T} = \{\Theta_1, \Theta_2, \dots, \Theta_t\}$ for robustness.

In considering deployment on edge devices, log Mel spectrograms are preferred over raw audio samples due to their lower

storage resource requirements [37]. Furthermore, CNNs have demonstrated their proficiency in extracting effective representations from log Mel spectrograms [38, 39]. Therefore, we evaluate the effectiveness of our proposed framework for SER using three classic CNN models: CNN-6, ResNet-9, and VGG-15. These CNN models are also suitable for deployment on devices with limited computing resources [37]. The detailed model architectures are described in Section 4.2.

3.2. Student Trajectories

Similarly, a *student trajectory* denotes also a sequence of model’s parameters trained on the D_{syn} . For faster distillation process and robustness, at each distillation step, we first randomly choose one *teacher trajectory* Θ_j from \mathcal{T} and then randomly chose the i -th snapshot of a model’s parameters θ_i from Θ_j . However, due to the diminishing informational value (i. e., model parameter changes) in later iterations during the training, we set an upper bound e^+ for the value i in the θ_i for the *student trajectory* initialisation.

At each distillation step, following the student trajectory’s initialisation, there are in total of Q numbers of updates based on the classification loss of the synthetic data. Specifically, considering the memory constraints, the synthetic dataset is partitioned into batches, and gradient descent updates are performed at each time according to the cross-entropy loss \mathcal{L}' calculated from the current batch of synthetic data, indicated as follows:

$$\hat{\theta}_{n+1} = \hat{\theta}_n - \alpha \nabla \mathcal{L}'((b_n); \hat{\theta}_n), \quad (1)$$

where $n \in \{0, 1, \dots, Q - 1\}$ indicates the current iteration number and b_n is the sampled batch of synthetic data, and α is a trainable learning rate to automatically adjust the magnitude of iterations P and Q for the student trajectories at each distillation step. After a certain number of distillation steps, the final distilled data D_{syn} is generated.

3.3. Training of Distilled Data

After Q updates to the student trajectory, with the snapshot of parameters θ_{i+P} retrieved from Θ and the parameters $\hat{\theta}_Q$, the distilled log Mel spectrograms are updated according to the normalised squared l -norm distance as follows:

$$\mathcal{L} = \frac{\|\hat{\theta}_Q - \theta_{i+P}\|_2^2}{\|\theta_i - \theta_{i+P}\|_l^2}, \quad (2)$$

where $Q \ll P$, showing the efficiency of the distillation process. This loss function directly encourages a similar student trajectory $\hat{\theta}_Q$ to the teacher trajectory θ_{i+P} trained on the D_{org} in the parameter space with much less iteration steps. The normalisation in the above loss function ensures the continuity and effectiveness of the learning signal across different training stages by

maintaining the relative scale of updates and preventing the potentially diminishing gradient issue, especially in the later training epochs [23].

4. Experiments and Results

4.1. Dataset

DEMoS. The Database of Elicited Mood in Speech (DEMoS) [40] is an Italian speech corpus consisting of 7.7 hours of audio recordings from 68 participants (45 males and 23 females). This dataset not only includes 332 neutral speech samples but also features a collection of 9,365 audio samples categorised into seven emotional states: anger, disgust, fear, guilt, happiness, sadness, and surprise. Same to prior studies [37, 41, 42] on the DEMoS dataset, this work excludes the minority neutral class, focusing on the 9,365 emotional speech recordings. These 9,365 samples are then allocated into training (40%), validation (30%), and test (30%) sets, adhering to a speaker-independent approach. The detailed data distribution of the DEMoS dataset is outlined in Table 1.

Table 1: *Emotion distribution of the DEMoS dataset.*

#	Train	Val	Test	Σ	F / M
Speaker	27	25	16	68	23 / 45
Anger	586	531	360	1,477	400 / 729
Disgust	666	608	404	1,678	596 / 1,082
Fear	461	404	291	1,156	524 / 871
Guilt	453	395	281	1,129	415 / 741
Happiness	561	471	363	1,395	516 / 961
Sadness	606	543	381	1,530	349 / 651
Surprise	396	358	246	1,000	532 / 998
Σ	3,729	3,310	2,326	9,365	3,332 / 6,033

4.2. Experimental Settings

Evaluation Metrics. The Unweighted Average Recall (UAR), alongside accuracy (i. e., weighted average recall), serves as the primary evaluation metric in alignment with previous studies on the DEMoS [37, 42] corpus, aiming to mitigate the issue of class imbalance.

Models Architecture. The CNN-6 model consists of four convolutional layers with output channels of 64, 128, 256, and 512, each with a kernel size of 3×3 . Each convolutional layer is followed by a local max pooling layer with a kernel size of 2×2 . The VGG-15 model features five convolutional blocks: the first two blocks each contain two convolutional layers, while the latter three blocks each comprise three convolutional layers. The number of output channels in these blocks is set as 64, 128, 256, 512, and 512, with all convolutional layers utilising 3×3 kernels. Following each convolutional block, a max pooling layer with a 2×2 kernel is employed. The ResNet-9 model initiates with a convolutional layer of 64 output channels with a 7×7 kernel and a stride of 2, followed by a max pooling layer with a 3×3 kernel. Subsequent to this are three ResNet blocks with output channel numbers 128, 256, and 512. These blocks leverage ‘shortcut connections’ that combine identity mappings with the outputs from two consecutive 3×3 convolutional layers [38]. For classification purposes, a global max pooling layer is utilised, followed by two fully connected (FC) layers to discern the contribution of each time-frequency bin. In this way, CNN-6, ResNet-9, and VGG-15 contain 4.44 M, 4.96 M, and 14.86 M parameters, respectively.

Implementation Details. In this work, all audio samples in the DEMoS dataset are re-sampled into 16 kHz for faster processing. Moreover, we unify the duration of all audio samples to

match that of the longest clip by trimming excessive portions and self-repeating segments of shorter ones. We extract log Mel spectrograms from DEMoS audio samples as features for models development. Specifically, the sliding window, overlap, and Mel bins are set as 512, 256, and 64 time frames, respectively. As a result, the obtained log Mel spectrograms are shaped as (373, 64), with the 373 representing the time axis and 64 describing the number of Mel frequency bins. The developing of the models employs an Adam optimiser with an initial learning rate of $1e-3$ and stops after 50 epochs. The learning rate decreases by 30% after every 5 epochs.

In our distillation setup, we define the number of teacher trajectories as $N_{\Theta} = 5$ to ensure robustness against random initialisation of the student trajectories. We empirically set the maximum starting iteration e^+ to 30 epochs when randomly selecting a parameter snapshot from Θ . For the initialisation of trainable distilled data, we sample $\{1, 5, 10, 50, 100, 150\}$ log Mel spectrograms per class (IPC). The iteration number Q applied to the student trajectory at each distillation step is fixed at 20, while P is set to 3 epochs of iterations. The synthetic data is updated using an Stochastic Gradient Descent (SGD) optimiser with an initial learning rate set to 1000. Concurrently, the learning rate α , used in the gradient updates (as defined in Equation 1), starts at $1e-3$ and is also adjusted using an SGD optimiser, with an initial learning rate of $1e-3$. The l is set as 2 in Equation 2. The entire distillation process is iterated for 60 cycles empirically. During the distillation process, the synthetic data is utilised every 10 distillation steps to develop three CNN models introduced in Section 3.1 from scratch, which are subsequently evaluated on the validation and test datasets of DEMoS. Given the reduced size of the synthetic dataset, the total number of training epochs is reduced to 30, while all other settings remain consistent with the development of the teacher trajectories.

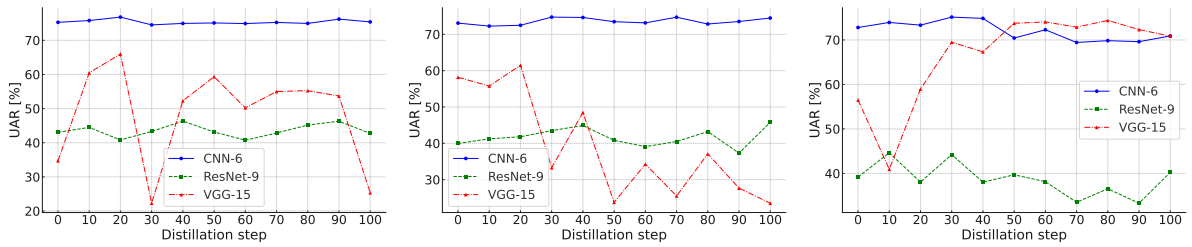
4.3. Results and Analysis

Models developed on full dataset. To ensure the efficacy of teachers networks in data distillation, we compare our performance with those of all state-of-the-art (SOTA) approaches on the DEMoS dataset, as shown in Table 2. CNN-6 achieves best validation UAR and best test UAR. Moreover, our approach outperforms most prior works on DEMoS dataset. Please note that the data split in [42] is different from our work and the work in [42] applies a more advanced wav2vec 2.0 model.

Table 2: *Comparison of the performances [UAR %] on the DEMoS dataset between our models (lower lines) and the state-of-the-art (SOTA). We present the mean \pm standard deviation of accuracy and UAR values.*

Models	Validation Dataset		Test Dataset	
	Accuracy	UAR	Accuracy	UAR
<i>Model self-distillation [42]</i>	-	91.8	-	91.4
<i>VGG-16 with Adv [38]</i>	-	87.5	-	86.7
<i>VGG-15 with NSL [37]</i>	74.9	70.0	85.9	78.9
CNN-6	88.9 \pm 0.7	88.8 \pm 0.9	88.3 \pm 0.6	88.3 \pm 0.6
ResNet-9	74.8 \pm 0.9	74.6 \pm 1.0	81.5 \pm 0.4	81.1 \pm 0.4
VGG-15	86.9 \pm 1.4	86.8 \pm 1.3	87.1 \pm 1.1	87.0 \pm 1.1

Models developed on the distilled dataset. Considering the memory constraints on IoT devices, we experiment with various sizes of synthetic datasets. While a higher IPC can enhance model performance, it also necessitates a larger dataset. Thus, we set the IPC values to a maximum of 150 to ensure the distilled



(a) Trained on data distilled with CNN-6. (b) Trained on data distilled with ResNet-9. (c) Trained on data distilled with VGG-15.
Figure 2: Comparison of the performance (UAR [%]) of the models on the DEMoS validation datasets, when IPC is set as 150.

datasets remain sufficiently small. Specifically, certain numbers of Log Mel spectrograms per class (IPC) are randomly chosen from the real original dataset and updated throughout the distillation process, with IPC values set as $\{1, 5, 10, 50, 100, 150\}$. These IPC values represents 0.1%, 0.5%, 1.0%, 5.0%, 9.9%, 14.9% of the training plus validation data samples and 0.2%, 0.9%, 1.9%, 9.4%, 18.8%, 28.2% of the training samples.

For each IPC, we apply the DEMoS validation dataset to test the effectiveness of the distilled data after every 10 distillation steps, up to 100. Figure 2 presents the results when $IPC = 150$. First, the performance of CNN-6 remains stable across the distillation steps, regardless of whether the synthetic data is guided by CNN-6, ResNet-9, or VGG-15, which may be attributed to its relatively simpler architecture and the adequacy of 150 IPC for training CNN-6. Second, VGG-15 shows improved performance when trained on synthetic data distilled by VGG-15 itself, likely due to its more complex architecture. Third, as evidenced by Figure 2, apart from the case with $IPC = 100$, we observe a distillation step of 60 appears optimal. Further distillation leads to no improvement or even a decrease in performance, possibly indicative of overfitting.

After fixing the total number of distillation steps to 60, we examine the impact of IPC on the efficacy of the synthetic data used for training from scratch. In Figure 3, there is a discernible trend showing that the performance of models trained on the synthetic distilled data improves with the increase of IPC. Notably, CNN-6 exhibits a more pronounced and rapid improvement, with considerable gains observed post an IPC threshold of 10. Upon reaching an IPC of 150, the performance of VGG-15 closely aligns with that of CNN-6 and their UARs exceeds 70%, outperforming one SOTA work [37] in Table 2. An IPC of 150 means we train the model with 28.2% of the original validation dataset. A comparable performance on the distilled data with $IPC = 150$ has verified the effectiveness of the proposed approach, which is another reason we do not increase the IPC further.

In Table 3, we present the model performances trained on distilled data guided by their respective model architectures as well as others. With IPC set at 150, the train dataset constitutes 28.2% and the combined train plus validation set accounts for 14.9% of the original corresponding dataset. On the validation dataset, CNN-6 achieves a UAR of 88.8% when trained on the full train dataset. Interestingly, even when trained on merely 28.2% of the data samples, its UAR remains above 72.0%, peaking at 74.9% when guided by its own architecture during the data distillation process. Similarly, ResNet-9 maintains a consistent validation UAR of around 40.0%. VGG-15, on the other hand, exhibits more variation in validation UAR when distilled datasets are guided by different models, achieving its peak at 74.1% UAR under the guidance of its own architecture. This performance significantly surpasses the work presented in [37] ($p < 0.01$ in a one-tailed z-test) and is comparable to ResNet-9’s performance trained on the full train dataset. Moreover, when

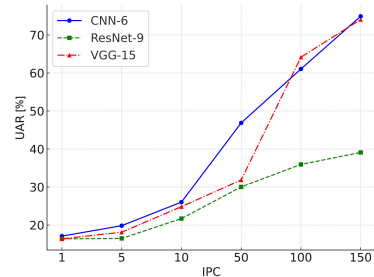


Figure 3: Analysis of the image per class (IPC) on the DEMoS validation dataset when the distillation step is fixed to 60. The model type in data distillation and evaluation remains same.

trained on the distilled train plus validation dataset, representing a mere 14.9% of the data samples, the test UAR experiences a decrease compared to the validation UAR. For instance, CNN-6, when trained with data distilled under ResNet-9’s guidance, yields a highest test UAR of 67.1%. Additionally, transferability of the distilled dataset has been observed. Specifically, CNN-6 and ResNet-9 perform similarly regardless of the guiding model for the distilled dataset, whereas VGG-15, when trained on data distilled by its own architectural style, outperforms the other models. This could be attributed to the inherent complexity and representational capacity of VGG-15, which may benefit more distinctly from data distilled within its architectural paradigm.

Table 3: Model performances [Accuracy / UAR %] on the DEMoS dataset. The models in the first column are used to guide the data distillation, while the models in the first row indicate those developed on the corresponding distilled data. $IPC = 150$ means merely 28.2% of the training data samples and 14.9% of training plus validation data samples.

IPC=150	CNN-6		ResNet-9		VGG-15	
	Val	Test	Val	Test	Val	Test
CNN-6	74.8 / 74.9	65.1 / 65.1	39.7 / 40.7	39.9 / 39.9	51.8 / 50.2	50.8 / 51.3
ResNet-9	72.8 / 73.2	67.2 / 67.1	39.4 / 39.1	41.3 / 40.5	32.9 / 34.3	28.4 / 29.2
VGG-15	72.0 / 72.3	65.6 / 65.9	38.9 / 38.1	38.3 / 37.4	74.2 / 74.1	61.4 / 61.2

5. Conclusions and Future Work

In this work, we proposed a data distillation framework for Speech Emotion Recognition that leverages knowledge transfer from pre-trained teacher trajectories to generate a compact, synthetic dataset. Our experimental results demonstrated that the framework could achieve comparable performance to models trained on the full dataset, with the highest test unweighted average recall reaching 67.1% using only 14.9% of the data samples. Additionally, we observed notable transferability in the distilled dataset. In future work, we will investigate the application of more complex models (e. g., wav2vec 2.0) to guide the data distillation process. We also intend to conduct an in-depth analysis of the characteristics of the generated distilled data to understand its underpinnings and potential for SER.

6. References

- [1] T. M. Wani *et al.*, “A comprehensive review of speech emotion recognition systems,” *IEEE access*, vol. 9, pp. 47 795–47 814, 2021.
- [2] B. W. Schuller, “Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends,” *Communications of the ACM*, vol. 61, no. 5, pp. 90–99, 2018.
- [3] Y. Chang, Z. Ren, Z. Zhang, X. Jing, K. Qian, X. Shao, B. Hu, T. Schultz, and B. W. Schuller, “STAA-Net: A Sparse and Transferable Adversarial Attack for Speech Emotion Recognition,” *arXiv preprint:2402.01227*, pp. 1–9, 2024.
- [4] A. Al-Talabani, H. Sellahewa, and S. A. Jassim, “Emotion recognition from speech: tools and challenges,” in *Proc. MMIPSA*, vol. 9497, 2015, pp. 193–200.
- [5] M. Płaza, R. Kazała, Z. Koruba, M. Kozłowski, M. Lucińska, K. Sitek, and J. Spyrka, “Emotion recognition method for call/contact centre systems,” *Applied Sciences*, vol. 12, no. 21, pp. 1–24, 2022.
- [6] M. Song, Z. Yang, A. Baird, E. Parada-Cabaleiro, Z. Zhang, Z. Zhao, and B. Schuller, “Audiovisual analysis for recognising frustration during Game-Play: Introducing the multimodal game frustration database,” in *Proc. ACII*, 2019, pp. 517–523.
- [7] S. Madanian *et al.*, “Speech emotion recognition using machine learning – A systematic review,” *Intelligent systems with applications*, pp. 200 266 (1–25), 2023.
- [8] J. Zhao, X. Mao, and L. Chen, “Speech emotion recognition using deep 1D & 2D CNN LSTM networks,” *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.
- [9] H. Meng, T. Yan, F. Yuan, and H. Wei, “Speech emotion recognition from 3D log-Mel spectrograms with deep learning network,” *IEEE Access*, vol. 7, pp. 125 868–125 881, 2019.
- [10] Z. Liu, X. Kang, and F. Ren, “Dual-TBNet: Improving the robustness of speech features via dual-transformer-bilstm for speech emotion recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2193–2203, 2023.
- [11] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NIPS*, 2020, pp. 12 449–12 460.
- [12] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [13] S. Chen *et al.*, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [14] R. Chatterjee, S. Mazumdar, R. S. Sherratt, R. Halder, T. Maitra, and D. Giri, “Real-Time Speech Emotion Analysis for Smart Home Assistants,” *IEEE Transactions on Consumer Electronics*, vol. 67, no. 1, pp. 68–76, 2021.
- [15] S. Latif *et al.*, “Speech technology for healthcare: Opportunities, challenges, and state of the art,” *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 342–356, 2021.
- [16] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, and T. Y. Al-Naffouri, “Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications,” *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 016–11 040, 2021.
- [17] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yi, H. Zhang, and Y. Zhang, “Deep learning on computational-resource-limited platforms: A survey,” *Mobile Information Systems*, vol. 2020, pp. 1–19, 2020.
- [18] A. Hashem, M. Arif, and M. Alghamdi, “Speech emotion recognition approaches: A systematic review,” *Speech Communication*, pp. 102 974 (1–29), 2023.
- [19] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, “A survey of machine and deep learning methods for internet of things (IoT) security,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.
- [20] H. Yin, L. Chen, W. Wang, X. Du, Q. V. H. Nguyen, and X. Zhou, “Mobi-sage: A sparse additive generative model for mobile app recommendation,” in *Proc. ICDE*, 2017, pp. 75–78.
- [21] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, “Dataset distillation,” *arXiv preprint:1811.10959*, pp. 1–14, 2018.
- [22] N. Sachdeva and J. McAuley, “Data distillation: A survey,” in *Proc. TMLR*, 2023, pp. 1–21.
- [23] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu, “Dataset distillation by matching training trajectories,” in *Proc. CVPR*, 2022, pp. 4750–4759.
- [24] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in *Proc. NIPS*, vol. 28, 2015, pp. 1–9.
- [25] 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 *Proc. LPCV*, 2022, pp. 291–326.
- [26] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” pp. 1–9, 2014.
- [27] X. Li, S. Wang, J. Sun, and Z. Xu, “Memory efficient data-free distillation for continual learning,” *Pattern Recognition*, vol. 144, p. 109875, 2023.
- [28] I. Sucholutsky and M. Schonlau, “Soft-label dataset distillation and text dataset distillation,” in *Proc. IJCNN*, 2021, pp. 1–8.
- [29] T. Nguyen, R. Novak, L. Xiao, and J. Lee, “Dataset distillation with infinitely wide convolutional networks,” in *Proc. NIPS*, vol. 34, 2021, pp. 5186–5198.
- [30] L. Deng, “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [31] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” *Master’s thesis, University of Toronto*, pp. 1–60, 2009.
- [32] P. Mohapatra, A. Pandey, B. Islam, and Q. Zhu, “Speech disfluency detection with contextual representation and data distillation,” in *Proc. IASA*, 2022, pp. 19–24.
- [33] C. You, N. Chen, F. Liu, D. Yang, and Y. Zou, “Towards data distillation for end-to-end spoken conversational question answering,” in *Proc. ICLR*, 2021, pp. 1–13.
- [34] T. Dong, B. Zhao, and L. Lyu, “Privacy for free: How does dataset condensation help privacy?” in *Proc. ICML*, 2022, pp. 5378–5396.
- [35] S. Hu, J. Goetz, K. Malik, H. Zhan, Z. Liu, and Y. Liu, “Fedsynth: Gradient compression via synthetic data in federated learning,” *arXiv preprint:2204.01273*, pp. 1–8, 2022.
- [36] P. Liu, X. Yu, and J. T. Zhou, “Meta knowledge condensation for federated learning,” *arXiv preprint:2209.14851*, pp. 1–14, 2022.
- [37] Y. Chang, Z. Ren, T. T. Nguyen, K. Qian, and B. W. Schuller, “Knowledge transfer for on-device speech emotion recognition with neural structured learning,” in *Proc. ICASSP*, 2023, pp. 1–5.
- [38] Z. Ren, A. Baird, J. Han, Z. Zhang, and B. W. Schuller, “Generating and protecting against adversarial attacks for deep speech-based emotion recognition models,” in *Proc. ICASSP*, 2020, pp. 7184–7188.
- [39] Y. Chang, X. Jing, Z. Ren, and B. W. Schuller, “CovNet: A transfer learning framework for automatic COVID-19 detection from crowd-sourced cough sounds,” *Frontiers in Digital Health*, vol. 3, no. 799067, pp. 1–11, 2022.
- [40] E. Parada-Cabaleiro, G. Costantini, A. Batliner, M. Schmitt, and B. Schuller, “DEMoS: An Italian emotional speech corpus,” *Language Resources and Evaluation*, pp. 1–43, Feb. 2019.
- [41] Z. Ren, J. Han, N. Cummins, and B. Schuller, “Enhancing transferability of black-box adversarial attacks via lifelong learning for speech emotion recognition models,” in *Proc. INTERSPEECH*, 2020, pp. 496–500.
- [42] Z. Ren, T. T. Nguyen, Y. Chang, and B. W. Schuller, “Fast yet effective speech emotion recognition with self-distillation,” in *Proc. ICASSP*, 2023, pp. 1–5.