INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Neural Model Reprogramming with Similarity Based Mapping for Low-Resource Spoken Command Recognition

Hao Yen[1,2] *Pin-Jui Ku*[1,2] *Chao-Han Huck Yang*[1] *Hu Hu*[1]
Sabato Marco Siniscalchi[1,3] *Pin-Yu Chen*[4] *Yu Tsao*[2]

[1]Georgia Institute of Technology, USA
[2]Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[3]Kore University of Enna, Italy and [4]IBM Research, USA

rick.yen@gatech.edu, pku9@gatech.edu, huckiyang@gatech.edu

Abstract

We propose a novel adversarial reprogramming (AR) approach for low-resource spoken command recognition (SCR) and build an AR-SCR system. The AR procedure aims at repurposing a pretrained SCR model (from the source domain) to modify the acoustic signals (from the target domain). To solve the label mismatches between source and target domains and further improve the stability of AR, we propose a novel similarity-based label mapping technique to align classes. In addition, the transfer learning (TL) technique is combined with the original AR process to improve the model adaptation capability. We evaluate the proposed AR-SCR system on three low-resource SCR datasets, including Arabic, Lithuanian, and dysarthric Mandarin speech. Experimental results show that with a pretrained acoustic model trained on a large-scale English dataset, the proposed AR-SCR system outperforms the current **state-of-the-art results** on Lithuanian and Arabic datasets, with only a limited amount of training data.

Index Terms: low-resource speech processing, adversarial reprogramming, spoken command recognition, transfer learning, label mapping.

1. Introduction

The aim of spoken command recognition (SCR) is to identify a target command out of a set of predefined candidates, based on an input utterance [1, 2]. Owing to its wide applicability to various domains, such as smart home devices [3] or crime defections [4], SCR has long been an important research topic in the speech processing field [5, 6]. Due to recent advances in deep learning (DL) algorithms, the performance of SCR systems has been significantly enhanced [7, 8, 9]. However, a common requirement to build a high-performance DL-based SCR system is to prepare a large amount of labeled training data (speech utterances and corresponding transcriptions of the commands). Such a requirement is not always realizable in real-world scenarios. In fact, it is generally favorable to build an SCR system with only a limited amount of training data. Such a scenario is often referred to as a low-resource training scenario [10].

For resource-limited speech command recognition, traditional methods leverage template matching-based approaches such as dynamic time warping (DTW) [11, 12] to look for similarities at the feature level. Recently, numerous algorithms have been developed to train DL-based systems under low-resource scenarios. A well-known category of approaches is transfer learning (TL) [13], which aims to use a small amount of training data to fine-tune a pretrained model, where the pretrained model is generally trained on a large-scale dataset. Prior arts have demonstrated the effectiveness of TL in speech processing tasks. For example, in [14], an English acoustic model (AM) pretrained on a large-scale training set is fine-tuned with limited training data to obtain a Spanish AM. Meanwhile in [15], a multilingual bottleneck feature extractor is pretrained on a large-scale training set and fine-tuned to form a keyword recognizer on a low-resourced Luganda corpus. The study of [16] collected 200 million pieces of 2-second data from Youtube to pretrain a speech embedding model to extract useful features for a downstream keyword spotting task. Although these TL approaches show promising results, the fine-tuning process (which is often done in the online mode) requires large training resources and thus is only feasible for applications where sufficient computation resources are available.

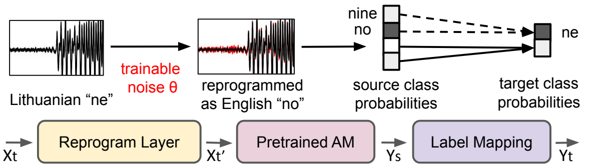

Figure 1: *Illustration of the proposed AR-SCR system. The acoustic signals of a Lithuanian command ("ne") is reprogrammed to English commands ("nine" and "no") and mapped to its final prediction with a pretrained English acoustic model.*

Another category of approaches is to adopt a pretrained model to extract representative features to facilitate efficient and effective training for the SCR systems. The pretrained model is generally trained on a large-scale dataset with either a supervised or self-supervised training manner. In [17] and [18], the SCR systems were established by adopting representative features extracted from a pretrained sound event detector and phone classifier, respectively. Both were trained on large-scale labeled data. Meanwhile, several methods adopt self-supervised models, such as PASE+ [19, 20] , wav2vec [21, 22], and wav2vec 2.0 [23], as feature extractors to build the SCR systems [24, 25, 6]. A notable drawback of this category of approaches is that an additional large-scale DL-based model (accordingly with increased hardware) is required.

Adversarial reprogramming (AR), as an alternative model adaptation technique, has been confirmed to provide satisfactory results in numerous machine learning tasks [26, 27]. In [28], AR adopts a trainable layer to generate additive noise (e.g., as additional information) on input acoustic signals (source-domain data) to guide an AM to recognize electrocardiography (ECG) signals (target-domain data). Along with the success of neural reprogramming [28, 29, 30], our study investigates whether AR can be applied to domain adaptation and accordingly building an SCR system in a low-resource training scenario. Different from [31, 32], which perform acoustic model adaptation by passing input through a neural net-

10.21437/Interspeech.2023-1086

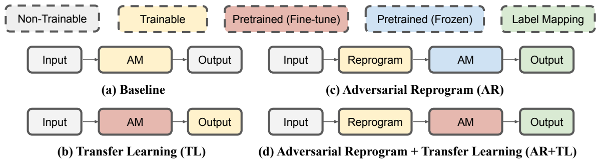

Figure 2: *Frameworks studied in this work. The "AM" block refers to the acoustic model. In (a), the baseline system is trained from scratch on the target domain data. In (b), AM is pretrained and then fine-tuned on the target domain data. In (c), AM is pretrained on the source domain and then fixed; an adversarial reprogram (AR) layer is then placed before the pretrained AM model to modify the input signals. In (d), we combine AR and TL to train the adversarial reprogram layer and fine-tune AM simultaneously.*

work layer for transformation, AR aims to add a trainable noise directly to the input sequence without modifying the model. Fig. 1 shows the design concept of the proposed AR-SCR system, which consists of a reprogram layer and a pretrained AM. The reprogram layer first generates trainable noises, θ, to modify the original signals before passing them into the pretrained AM. The AM will then output class probabilities corresponding to the source classes. A label mapping technique is adopted to map the probabilities of source classes to the target class by aggregating probabilities over the assigned source labels. Based on the aggregated probabilities, the reprogram layer is further trained to generate noises and thus modify the input signals, so that the pretrained AM can be repurposed to perform recognition in the target task.

The proposed AR-SCR system adopts two additional techniques to further improve the model adaptation capability: (1) a novel similarity-based label mapping strategy that aims to align the target and source classes and (2) a fine-tuning process that adjusts the AM with the AR-generated signals. Experiment results on three low-resource SCR datasets, including Arabic, Lithuanian, and dysarthric Mandarin speech command datasets, demonstrate that the proposed AR-SCR system can yield better performance than other state-of-the-art methods. In summary, the major contribution of the present work is twofold: 1) This is the first study that investigates the applicability of AR to low-resource SCR tasks with promising results. 2) We verify that AR has the flexibility to combine with the TL technique to achieve a better model adaptation performance.

2. The Proposed AR-SCR System

2.1. AR-SCR System

Fig. 2 illustrates the overflow of an SCR system with AR and TL model adaptation techniques. In Fig. 2 (a), the AM is directly trained by data from the target-domain task. When the training data from the target domain is limited, the model cannot be trained well and thus may result in unsatisfactory recognition performance. In Fig. 2 (b), the TL technique is applied to the pretrained AM to establish a new SCR system that matches the target domain. In Fig. 2 (c), the pretrained AM is fixed, and an adversarial reprogram layer is trained to transform the input signals to reduce the distance between the target and source distributions. In Fig. 2 (d), the adversarial reprogram layer is treated as a front-end processor, and the TL technique is applied to further fine-tune AM with the reprogrammed signals. We expect the combination of AR (as a front-end processing) and TL (as a back-end processing) can reach better model adaptation capability due to their complementary abilities.

2.2. Acoustic Signal Reprogramming

The concept of AR was first introduced in [26], and its aim was to determine a trainable input transformation function $\mathcal{H}$ to repurpose a pretrained model from the source domain to a target task. The authors in [26] showed that by the AR process, a pretrained ImageNet model trained on the image classification task can solve a square-counting task with high accuracy. A later study [33] demonstrated that a reliable classification system can be established using AR and a black-box pretrained model with scarce data and limited resources. Meanwhile, the Voice2Series method [28] is proposed to transfer time series data (e.g., ECG or Earthquake) x_t, as the target domain, $\mathcal{X}_{\mathcal{T}} \subseteq \mathbb{R}^{d_{\mathcal{T}}}$ from the source domain $\mathcal{X}_{\mathcal{S}} \subseteq \mathbb{R}^{d_s}$, where dimension $d_{\mathcal{T}} < d_{\mathcal{S}}$. For these AR approaches, a reprogrammed sample x'_t can be formulated as:

$$x'_t = \mathcal{H}(x_t; \theta) := \mathrm{Pad}(x_t) + \underbrace{M \odot \theta}_{\delta}, \tag{1}$$

where $\mathrm{Pad}(x_t)$ generates a zero-padded time series of dimension $d_{\mathcal{S}}$. The binary mask $M \in \{0,1\}^{d_{\mathcal{S}}}$ indicates the indexes that are not occupied and reprogrammable. $\theta \in \mathbb{R}^{d_{\mathcal{S}}}$ is a set of trainable parameters for aligning source and target domain data distributions. The term δ denotes the trainable additive input transformation for reprogramming. In the original AR method, the target sequence must be shorter than the source counterpart. To overcome this limitation, we design to add trainable noises to the whole sequence, and thus Eq. (1) becomes

$$x'_t = \mathcal{H}(x_t; \theta) := x_t + \theta. \tag{2}$$

In this work, we focus on applying AR as a model adaptation technique to effectively fine-tune a pretrained model.

2.3. Pretrained Acoustic Model

In the Voice2Series study [28], the authors have compared several well-known AMs as pretrained models and provided the first theoretical justifications by optimal transport for reprogramming general time-series signals to acoustic signals. Based on the provided justifications, in this study, we established AM with two layers of fully convolutional neural networks, followed by two bidirectional recurrent neural networks, which are then combined with an attention layer. To train the model, we use the Google Speech Commands dataset [34], which is a large-scale collection of spoken command words, containing 105,829 utterances of 35 words from 2,618 speakers; all the utterances are recorded in a 16 kHz sampling-rate format. The pretrained AM has 0.2M parameters and yields a 96.90% recognition accuracy rate on the testing set of Google Speech Commands.

2.4. Similarity Label Mapping

As illustrated in Fig. 1, a label mapping function is adopted to map the probabilities of the source class to that of the target class. The results in [28] show that a many-to-one label mapping strategy (randomly mapping multiple classes from the source task to an arbitrary target class) yields a better performance as compared to the one-to-one mapping strategy. In this study, we attempt to improve the random mapping process with a similarity mapping that considers the relationships of data in the source and target domains. In [35], the structural relationships between acoustic scene classes are explored and utilized to address the domain mismatch issue, while the authors in [36] proposed an unsupervised learning method that maps

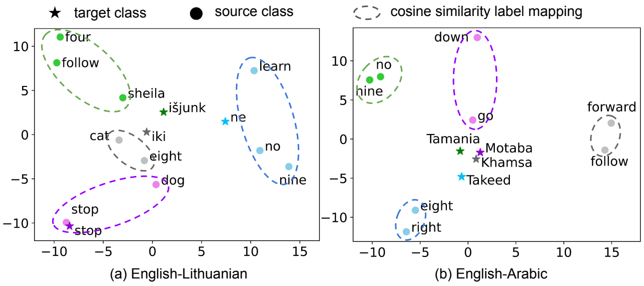

Figure 3: *PCA plots of average representations of several source-target pairs for (a) English-Lithuanian and (b) English-Arabic datasets. A target class (star point) is mapped to two or three source classes (circle points) with higher cosine similarity (marked with same colors).*

sentences in two different languages into the same latent space for the Machine Translation task. Inspired by the prior arts, we propose to investigate the similarity of the labels between the source and target domains and determine the optimal many-to-one label mapping strategy. To compute the similarity, we first feed source and target data to the pretrained AM and calculate the average representations of all classes and compute the cosine similarity between each of them. Based on the similarity of classes, each target class is mapped to two or three source classes in our AR-SCR system. Fig. 3 (a) and (b), respectively, show the PCA plots of representation vectors for the English-Lithuanian and English-Arabic datasets. Interestingly, we can observe that command words with similar acoustic characteristics are mapped to the same target word. For example, in Fig. 3 (a), the source English classes ”nine, no, learn” are mapped to the target Lithuanian class ”ne”; whereas in Fig. 3 (b), the source English classes ”right, eight” are mapped to the target Arabic class ”Takeed”. In our experiment, the AR system with similarity mapping strategy outperforms the one with random mapping [28], improving the average testing accuracy by 2.1%, 13.4%, and 9.2% on the Arabic, Lithuanian, and Mandarin datasets respectively.

3. Experiments

3.1. Speech Commands Dataset

We use the Google Speech Commands [34] as the source domain data to pretrain our AM. Three low-resource SCR datasets, including the Lithuanian, dysarthric Mandarin, and Arabic speech command datasets, are used as the target domain tasks.

Lithuanian Speech Commands: The content of the Lithuanian speech commands dataset [25] is created by translating 20 keywords from Google Speech Commands [34] into the Lithuanian language. The dataset consists of recordings from 28 speakers, with each speaker pronouncing 20 words on a mobile phone. We follow the setting in [25] and [37] and chose 15 target classes: 13 command words, 1 unknown word, and 1 silent class. The resulting dataset consists of 326 recordings for training, 75 for validation, and 88 for testing.

Dysarthric Mandarin Speech Commands: The dataset contains 19 Mandarin commands, each uttered 10 times from 3 dysarthric patients [38], with a 16kHz sampling rate. These 19 commands include 10 action commands and 9 digits, which are designed to allow dysarthric patients to control web browsers via speech. By removing long recordings, we select 13 short commands with durations around one second in our experiments. We follow the setting in [38] to split the whole dataset into 70% and 30% to form training and testing tests.

Table 1: *Testing results (average and std of the accuracy scores) of three conditions (3, 10, and 20 training samples) for the Lithuanian speech commands dataset.*

Limit	System	Avg. Acc. (%)	Rel. Imp. (%)	Std.
3	DTW	45.6	–	–
	Baseline	33.8	–	±8.82
	AR	25.1	-25.7	±**2.27**
	TL	52.8	56.2	±7.69
	AR+TL	**58.9**	**72.8**	±4.84
10	DTW	50.0	–	–
	Baseline	63.7	–	±5.35
	AR	34.4	-46.0	±**4.1**
	TL	80.7	26.7	±5.65
	AR+TL	**81.9**	**28.6**	±4.44
20	DTW	64.5	–	–
	Baseline	70.3	–	±6.8
	AR	34.6	-51.4	±3.0
	TL	86.8	23.5	±2.92
	AR+TL	**88.6**	**26**	±**1.7**

Arabic Speech Commands: The Arabic speech commands dataset [39] consists of 16 commands, including 6 control words and 10 digits (0 through 9), and each command has 100 samples, amounting to 1600 utterances in total. 40 speakers are involved in preparing the dataset. The speech utterances are recorded at a sampling rate of 48 kHz and then converted into 16 kHz in our experiments. We follow the setting in [39] and split the dataset into 80% for training and 20% for testing. In addition, we randomly excerpt 20% of the training data to form a validation set.

3.2. Experimental Results

For each of the three datasets, we compare the SCR results of the four systems, as shown in Fig. 2. Additionally, we consider a dynamic time warping method [40], denoted as DTW, for comparison. The DTW algorithm uses a learned warping window and a nearest neighbor classifier on input waveforms to perform speech command recognition. The results of the DL-based baseline system (denoted as Baseline), stand for an AM trained from scratch on the training set of the three SCR datasets. The results of the systems with AR and TL model adaptations (Fig. 2 (b) and (c)) are denoted as AR and TL, respectively. The results of combining the AR and TL methods referred to Fig. 2 (d) are denoted as AR+TL. In our preliminary experiments, we tested the performance of several different setups. We have compared the domains for signal reprogramming by directly modifying the input waveforms or modifying the input spectral features. Meanwhile, we evaluated different label mapping techniques, i.e., one-to-one versus many-to-one, as well as random mapping versus similarity mapping. Empirically, all the best results we obtained use the many-to-one similarity mapping technique. In the following discussions, we only report the results of the best setups. For all the experiments in this study, we tested each SCR system 10 times and the average accuracy and standard deviation (std.) values are reported in each table.

Table 1 lists the testing results of the Lithuanian speech commands dataset. We follow the setting in [25] to conduct experiments using different amounts of training data and report the results of three conditions: each keyword has 3, 10, and 20 training samples. From Table 1, we observe that TL consistently outperforms DTW and Baseline for the three conditions. Notably, although AR alone cannot attain improved per-

Table 2: *Testing results (average and std of the accuracy scores) of the dysarthric Mandarin dataset.*

System	Avg. Acc. (%)	Rel. Imp. (%)	Std.
DTW	48.7	–	–
Baseline	64.0	–	±16.43
AR	33.2	-48.12	±2.88
TL	78.1	22.03	±3.18
AR+TL	**82.3**	**28.59**	**±2.56**

formance, AR+TL can yield average accuracy of 58.9%, 81.9%, and 88.6% for 3, 10, and 20 training samples conditions, respectively, which are notably better than TL with the average accuracy of 52.8%, 80.7%, and 86.7%. The best results for AR+TL are obtained by modifying the input waveforms with a reprogramming layer. Moreover, the std values in Table 1 show that AR can improve performance stability when combined with the TL technique. Fig. 4 also compares our best AR+TL system with the state-of-the-art SCR system [25], where the best-1 accuracy results are reported for the 3, 10, and 20 conditions. From the figure, AR+TL consistently outperforms the state-of-the-art wav2vec SCR system [25] on the Lithuanian speech commands dataset.

Table 2 shows the results of the dysarthric Mandarin dataset. A similar observation can be obtained as those from Table 1 with input waveforms reprogramming. First, TL yields an improved performance as compared to DTW and Baseline. Next, although AR alone can not provide better results than Baseline, AR+TL achieves the best performance among the four systems. As compared to Baseline, AR+TL yields notable improvements of 28.59% (from 64.0 to 82.3). Moreover, the low std value of ±2.56 suggests that AR can improve the stability of the SCR systems.

We also conducted experiments on the Arabic speech commands dataset. Empirically, we found that modifying input spectral features with the reprogramming layer gives the best performance. From the testing results, we first note that the three adaptation systems (AR, TL, and AR+TL) all yield improved accuracy rates with lower std as compared to Baseline. AR alone can achieve 95.8% accuracy with only 16k trainable parameters, which is far fewer than the Baseline system (200.4k). Next, we note that AR+TL can achieve the highest accuracy of 98.9%, which outperforms individual AR (94.8%) and TL (98.6%). Based on our literature survey, AR+TL also outperforms the state-of-the-art SCR system using LSTM [39] (98.13%) on this Arabic speech commands dataset.

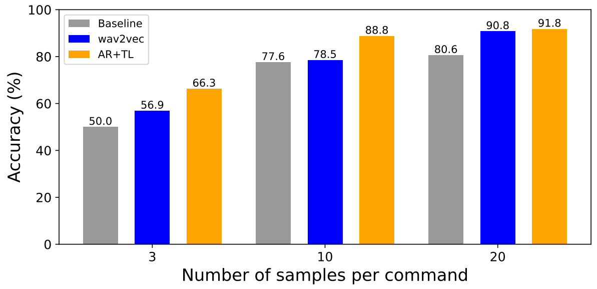

Figure 4: *The best-1 accuracy values of Baseline, the state-of-the-art wav2vec model [25], and the proposed AR+TL system on the Lithuanian speech command dataset. We follow the same evaluation metrics in [25] and only reported the best-1 test accuracy values.*

Table 3: *Testing results (average and std of the accuracy scores) of two conditions (3, and 10 training samples) for the Lithuanian speech commands dataset.*

Limit	System	Avg. Acc. (%)	Rel. Imp. (%)	Std.
3	Baseline+Aug	39.5	-	±8.65
	TL+Aug	65.9	66.8	±4.52
	AR+TL+Aug	**70.2**	**77.7**	**±2.72**
10	Baseline+Aug	65.3	-	±8.80
	TL+Aug	81.7	25.1	±2.64
	AR+TL+Aug	**87.4**	**33.8**	**±2.63**

3.3. Ablation Study: Combining Adversarial Reprogramming with Data Augmentation

As mentioned in Section 2, AR could be easily combined with TL and achieve the highest performance. It is thus very natural to assume that we can combine AR with other low-resource training techniques and further improve the performance of the whole system. To verify whether AR has such flexibility, we combine AR with Data Augmentation (Aug), another standard approach when training DL-based systems with limited amount of data. In our experiments, we utilize SpecAugment [41], a common data augmentation method in the speech processing field which randomly masks input features by zero along both time and frequency axes. We dynamically perform augmentation to generate additional data during training. We compare the results by combining Aug with the Baseline, TL, and AR+TL systems mentioned above. We respectively denoted the three systems as Baseline+Aug, TL+Aug, and AR+TL+Aug.

Table 3 shows the testing results of the Lithuanian Speech command dataset when each keyword has 3 or 10 training samples. Compared with the results in Table 1, data augmentation can improve and stabilize the performance of the AR and TL systems by a large margin. Similar results can be observed as those from Table 1. First, TL+Aug improves the performance by 66.8% and 25.1% as compared to Baseline+Aug. Then, AR+TL+Aug again achieves the highest average accuracy of 70.2% and 87.4% among the three systems, which is notably better than TL+Aug (65.9% and 81.7%) and Baseline+Aug (39.5% and 65.3%), under the conditions of 3 and 10 training samples. Moreover, the lower std values also show that AR combined with data augmentation can improve the stability of the systems. The results show that AR could be successfully combined with data augmentation schemes and further improves the SCR system under low-resource scenarios.

4. Conclusion

In this paper, we introduce an AR approach to establish an SCR system with a very limited amount of training data. Experimental results show that the proposed AR-SCR system can yield better performance as compared with state-of-the-art SCR methods on the Lithuanian and Arabic speech command datasets [39, 25]. The results also demonstrate that AR can effectively improve the accuracy over the baseline with few trainable parameters on the Arabic dataset. Meanwhile, we also show AR has great flexibility to combine with different low-resource training techniques, such as transfer learning and data augmentation. In our future work, we will explore to further improving the AR-SCR performance by combining it with self-supervised learning methods. Meanwhile, we will open our source code and all hyperparameters with pre-trained models.

5. References

[1] T. Zeppenfeld and A. Waibel, "A hybrid neural network, dynamic programming word spotter," in *Proc. ICASSP*, 1992.

[2] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Proc. ICASSP*, 1989.

[3] S. Bajpai and D. Radha, "Smart phone as a controlling device for smart home using speech recognition," in *Proc. ICCSP*, 2019.

[4] H. Kavya and V. Karjigi, "Sensitive keyword spotting for crime analysis," in *Proc. NCCSN*, 2014.

[5] T. Dinushika, L. Kavmini, P. Abeyawardhana, U. Thayasivam, and S. Jayasena, "Speech command classification system for sinhala language based on automatic speech recognition," in *Proc. IALP*, 2019.

[6] D. Seo, H.-S. Oh, and Y. Jung, "Wav2KWS: Transfer learning from speech representations for keyword spotting," *IEEE Access*, pp. 1–1, 2021.

[7] D. C. de Andrade, S. Leo, M. Viana, and C. Bernkopf, "A neural attention model for speech command recognition," *ArXiv*, vol. abs/1808.08929, 2018.

[8] S. Majumdar and B. Ginsburg, "Matchboxnet: 1d time-channel separable convolutional neural network architecture for speech commands recognition," in *Proc. Interspeech*, 2020.

[9] R. Vygon and N. Mikhaylovskiy, "Learning efficient representations for keyword spotting with triplet loss," in *Proc. SPECOM*, 2021.

[10] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, p. 85–100, 01 2014.

[11] I. Szoke, M. Skacel, L. Burget, and J. Cernocky, "Copingwith channel mismatch in query-by-example - but quesst 2014," in *Proc. ICASSP*, 04 2015.

[12] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping to massive dataset," in *Proc. PKDD*, 1999.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[14] E. Tetariy, Y. Bar-Yosef, V. Silber-Varod, M. Gishri, R. Alon-Lavi, V. Aharonson, and A. Moyal, "Cross-language phoneme mapping for phonetic search keyword spotting in continuous speech of under-resourced languages," *Artificial Intelligence Research*, vol. 4, no. 2, 2015.

[15] R. Menon, H. Kamper, E. van der Westhuizen, J. Quinn, and T. Niesler, "Feature exploration for almost zero-resource asr-free keyword spotting using a multilingual bottleneck extractor and correspondence autoencoders," in *Proc. Interspeech*, 2019.

[16] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi, "Training keyword spotters with limited and synthesized speech data," in *Proc. ICASSP*, 2020.

[17] B. McMahan and D. Rao, "Listening to the world improves speech command recognition," in *AAAI Conference on Artificial Intelligence*, 2017.

[18] S. Myer and V. S. Tomar, "Efficient keyword spotting using time delay neural networks," in *Proc. Interspeech*, 2018.

[19] S. Pascual, M. Ravanelli, J. Serra, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Proc. Interspeech*, 2019.

[20] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *Proc. ICASSP*, 2020.

[21] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, 2019.

[22] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. ICLR*, 2019.

[23] A. Baevski, H. Zhou, A. rahman Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.

[24] A. Mittal, S. Bharadwaj, S. Khare, S. Chemmengath, K. Sankaranarayanan, and B. Kingsbury, "Representation based meta-learning for few-shot spoken intent recognition," in *Proc. Interspeech*, 2021.

[25] A. Kolesau and D. Šešok, "Unsupervised pre-training for voice activation," *Applied Sciences*, 2020.

[26] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," in *Proc. ICLR*, 2019.

[27] M. Englert and R. Lazic, "Adversarial reprogramming revisited," in *Proc. NeurIPS*, 2022.

[28] C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, "Voice2Series: Reprogramming acoustic models for time series classification," in *Proc. ICML*, 2021.

[29] C.-H. H. Yang, B. Li, Y. Zhang, N. Chen, R. Prabhavalkar, T. N. Sainath, and T. Strohman, "From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition," in *Proc. ICASSP*, 2023.

[30] H.-H. Chen and A. Lerch, "Music instrument classification reprogrammed," in *Proc. MMM*, 2022.

[31] J. Neto, L. Almeida, M. Hochberg, C. Martins, N. L., S. Renals, and T. Robinson, "Speaker-adaptation for hybrid hmm-ann continuous speech recognition system," in *Proc. EUROSPEECH*, 1995.

[32] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. SLT*, 2012.

[33] Y.-Y. Tsai, P.-Y. Chen, and T.-Y. Ho, "Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources," in *Proc. ICML*, 2020.

[34] P. Warden, "Speech Commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[35] H. Hu, S. M. Siniscalchi, Y. Wang, and C.-H. Lee, "Relational teacher student learning with neural label embedding for device adaptation in acoustic scene classification," in *Proc. Interspeech*, 2020.

[36] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *Proc. ICLR*, 2017.

[37] A. Kolesau and D. Šešok, "Voice activation for low-resource languages," *Applied Sciences*, vol. 11, no. 14, 2021.

[38] Y.-Y. Lin, W.-Z. Zheng, W. C. Chu, J.-Y. Han, Y.-H. Hung, G.-M. Ho, C.-Y. Chang, and Y.-H. Lai, "A speech command control-based recognition system for dysarthric patients based on deep learning technology," *Applied Sciences*, vol. 11, no. 6, 2021.

[39] L. Benamer and O. Alkishriwo, "Database for arabic speech commands recognition," in *Proc. CEST*, 2020.

[40] T. Rakthanmanon, B. J. L. Campana, A. A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. KDD*, 2012.

[41] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019.