Sum-Product Networks for Robust Automatic Speaker Identification

Aaron Nicolson and Kuldip K. Paliwal

Signal Processing Laboratory, Griffith University, Brisbane, Queensland, Australia, 4111

aaron.nicolson@griffithuni.edu.au, k.paliwal@griffith.edu.au

Abstract

We introduce sum-product networks (SPNs) for robust speech processing through a simple robust automatic speaker identification (ASI) task. SPNs are deep probabilistic graphical models capable of answering multiple probabilistic queries. We show that SPNs are able to remain robust by using the marginal probability density function (PDF) of the spectral features that reliably represent speech. Though current SPN toolkits and learning algorithms are in their infancy, we aim to show that SPNs have the potential to become a useful tool for robust speech processing in the future. SPN speaker models are evaluated here on real-world non-stationary and coloured noise sources at multiple signal-to-noise ratio (SNR) levels. In terms of ASI accuracy, we find that SPN speaker models are more robust than two recent convolutional neural network (CNN)-based ASI systems. Additionally, SPN speaker models consist of significantly fewer parameters than their CNN-based counterparts. The results indicate that SPN speaker models could be a robust, parameter-efficient alternative for ASI. Additionally, this work demonstrates that SPNs have potential in related tasks, such as robust automatic speech recognition (ASR) and automatic speaker verification (ASV).

Availability: The SPN ASI system is available at `https://github.com/anicolson/SPN-ASI`.

Index Terms: sum-product networks (SPN), marginalisation, missing-feature approach, robust automatic speaker identification.

1. Introduction

The task of a text-independent automatic speaker identification (ASI) system is to identify a speaker from a given voice recording, irrespective of its linguistic content. This is accomplished by modelling the voice characteristics of each speaker after an enrolment phase [1]. Common applications of ASI include the selection of a speaker-dependent acoustic model for an automatic speech recognition (ASR) system [2] and speaker segmentation — an important pre-processing step for speaker diarisation [3]. The realisation of each application is dependent upon a high-performance ASI system. The first widely adopted ASI system utilised Gaussian mixture model (GMM) speaker models [4].

One obstacle that prevented the commercial introduction of GMM speaker models was their poor performance in the presence of noise [5], spurring the investigation of robust approaches [6]. A noteworthy approach was the missing-feature approach, which is underpinned by evidence that speech is intelligible to humans even after it has undergone substantial spectral masking [7]. *Marginalisation*, as proposed by Cook *et al.* [8], has been the most prominent missing-feature approach in the literature [9], and is able to significantly increases the robustness of a GMM speaker model [10]. For marginalisation, the marginal probability density function (PDF) is obtained by integrating over the components of the feature vector that have been classified as unreliable representations of speech [11]. Classification is thus performed on a partial instantiation of a given feature vector, consisting of only the components that reliably represent speech.

Recently, ASI and automatic speaker verification (ASV) systems employing deep neural networks (DNNs) have demonstrated a higher performance than GMM and i-vector-based systems [12]. One example is the x-vector system, which utilises pooling and a DNN trained to discriminate between speakers to map speech to a fixed-size embedding [13]. Convolutional neural networks (CNNs) have also been employed [14]. SincNet is a CNN that employs parametrised sinc functions to predefine a bank of band-pass filters for its first layer [15]. Another example proposed by Xie *et al.* [16] utilises a 'thin' residual CNN (referred to as Xie2019 henceforth). It also includes dictionary-based NetVLAD [17] and GhostVLAD [18] layers for feature aggregation. Despite their high performance on clean speech, modern ASI systems are still susceptible to performance degradation in the presence of noise [19]. Additionally, DNNs are not probabilistic models and cannot employ classifier-compensation missing-feature approaches, such as marginalisation. Currently, the most popular approach to increase the robustness of a DNN-based system is to use a front-end to pre-process the noisy speech [20, 21].

In 2011, Poon *et al.* [22] proposed a deep tractable probabilistic graphical model called the sum-product network (SPN). An SPN can be described as a deep neural network (DNN) restricted to using sum and product operators. When viewed as a probabilistic graphical model, an SPN can be described as a rooted directed acyclic graph with distributions as leaves. SPNs have clear semantics; each node represents an unnormalised joint probability distribution over a set of variables. As they can answer marginal inference queries, SPNs lend themselves well to marginalisation. One disadvantage is that structure and weight learning algorithms for SPNs, as well as libraries, are currently undeveloped, as highlighted by Jaini *et al.* [23]. However, the long-term outlook of SPNs is positive. New SPN toolkits are being developed, such as LibSPN [24], that take advantage of modern machine learning toolkits, such as TensorFlow [25]. Additionally, recently proposed SPN architectures developed for temporal (dynamic SPNs [26]) and spatial representations (deep generalised convolutional SPNs (DGC-SPNs) [27]) have shown promising results.

We propose SPNs and marginalisation for robust ASI. We first formulate marginalisation for SPNs. We then investigate SPNs and marginalisation on a simple robust ASI task. The structure of each SPN speaker model is learned using LearnSPN [28]. SPN speaker models are evaluated against GMM speaker models [29], SincNet [15], and Xie2019 [16]. The SPN and GMM speaker models employ marginalisation, whilst SincNet and Xie2019 employ the long short-term memory ideal ratio mask (LSTM-IRM) estimator by Chen *et al.* [30] as a front-

end. SPN speaker models are evaluated using multiple conditions, including real-world non-stationary and coloured noise sources and multiple signal-to-noise ratio (SNR) levels. From the presented results, we aim to demonstrate the following: 1) SPN speaker models are more robust than GMM speaker models when marginalisation is and is not used, 2) SPN speaker models utilising marginalisation have the potential to be more robust than recent CNN-based ASI systems that employ a front-end technique, and 3) SPNs and marginalisation have potential in related robust speech processing tasks, such as robust ASR and ASV.

2. SPN speaker models

2.1. Features

For marginalisation, a frequency-domain representation is required. Hence, we employ the log-spectral subband energies (LSSEs) of the clean speech power spectral density (PSD) estimate as features for the SPN and GMM speaker models. The LSSEs are computed from the single-sided PSD estimate:[1]

$$\mathrm{X}_b = \log \sum_{k=0}^{N_d/2} h_{b,k} \hat{P}_k, \quad 0 \leq b \leq B-1, \tag{1}$$

where N_d denotes the time-frame duration in discrete-time samples, k denotes the discrete-frequency bin, $\hat{P}_k$, for all k, denotes the PSD estimate for a given time-frame, and $h_{b,k}$, for all k, denotes the b^{th} filter of a bank of B triangular-shaped critical band filters spaced uniformly on the mel-scale. The PSD is estimated from the short-time Fourier transform (STFT) of the clean speech using the periodogram method, as in [10].

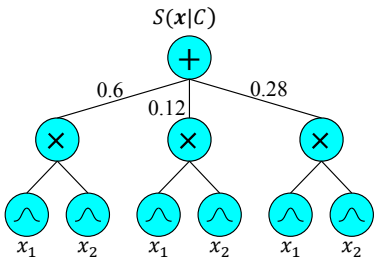

Figure 1: *SPN speaker model with univariate Gaussian leaves.*

2.2. SPN speaker models with Gaussian leaves

An SPN [22] specifies an unnormalised joint distribution over a set of random variables, $\mathbf{X} = (X_1, X_2, ..., X_B)^\top$, where in this case, $\mathbf{X}$ is the LSSEs for a time-frame of clean speech. An observation of $\mathbf{X}$ is denoted by $\mathbf{x} = (x_1, x_2, ..., x_B)^\top$. Hence, the SPN, S, for speaker class C is a function of the observed feature vector, $S(\mathbf{x}|C)$, where the value of the SPN is given by its root. An SPN consists of multiple layers of sum and product nodes, with distributions as leaves. The multivariate distribution of the i^{th} leaf is over a subset of the variables: $\mathbf{X}_i \subseteq \mathbf{X}$, and is assumed to be normally distributed: $\mathbf{X}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|i, C)$, with mean $\boldsymbol{\mu}$, and diagonal covariance $\boldsymbol{\Sigma}$. The PDF for the i^{th} leaf is given by

$$\mathcal{N}(\mathbf{x}_i|i, C) = \prod_{d \in \mathbf{D}} \frac{1}{\sqrt{2\pi\Sigma_{i,C}(d,d)}} e^{-\frac{(\mathbf{x}_i(d) - \boldsymbol{\mu}_{i,C}(d))^2}{2\Sigma_{i,C}(d,d)}}, \tag{2}$$

where $\mathbf{D} \subseteq (1, 2, ..., B)^\top$ indicates the random variable indices for $\mathbf{X}_i$. An SPN over two variables with univariate Gaussian leaves is shown in Figure 1.

If node i is a product node, its value is given by the product of the values of its children, $Ch(.)$: $S_i = \prod_{j \in Ch(S_i)} S_j$, where S_j is the j^{th} child of node S_i. If node i is a sum node, its value is given by the sum of the values of its children: $S_i = \sum_{j \in Ch(S_i)} w_{ij} S_j$, where weight w_{ij} is the non-negative weighted edge between S_i and S_j. To be a *valid* joint distribution, an SPN must be both *decomposable*, and *complete*, as described in [22]. The scope of a node, $Sc(.)$, is defined as the set of variables that are descendants of it. An SPN is said to be decomposable when the scopes of the children of its product nodes are disjoint: $\forall S_j, S_k \in Ch(S_i), Sc(S_j) \cap Sc(S_k) = \emptyset$, where $\emptyset$ indicates an empty set. An SPN is said to be complete when the scopes of the children of its sum nodes are identical: $\forall S_j, S_k \in Ch(S_i),\ Sc(S_j) = Sc(S_k)$.

2.3. Marginalisation for SPNs

For marginalisation, each component of an observed noisy speech feature vector is classified as either a reliable or an unreliable representation of the corresponding unobserved clean speech component. The noisy speech feature vector, $\mathbf{y}$, can thus be described as the union of the reliable and unreliable components: $\mathbf{y} = \mathbf{y}^r \cup \mathbf{y}^u$. Here, we not only apply marginalisation to SPNs, but also *bounded marginalisation*, as proposed by Cook *et al.* [22]. For bounded marginalisation, the value of an unreliable component is utilised as the upper bound of the unobserved clean speech component value. For LSSEs, the bounds are taken from $[-\infty, \mathbf{y}_n^u]$. Thus, the PDF for the i^{th} leaf becomes:

$$\mathcal{N}(\mathbf{y}_i^r, \mathbf{x}_i^u \leq \mathbf{y}_i^u|i, C) = \mathcal{N}(\mathbf{y}_i^r|i, C) \int_{-\infty}^{\mathbf{y}_i^u} \mathcal{N}(\mathbf{x}_i^u|i, C) d\mathbf{x}_i^u. \tag{3}$$

For marginalisation, the unreliable components are treated as missing and the bounds are taken from $[-\infty, \infty]$. The integral in Equation (3) thus reduces to unity, giving $\mathcal{N}(\mathbf{y}_i^r|i, C)$. When all of the components of $\mathbf{y}_i$ are unreliable, it is treated as a vector with no instantiated components: $\mathcal{N}(\mathbf{y}_i^r = \emptyset|i, C) = 1$.

3. Experiment setup

3.1. Signal processing

The feature vectors for the GMM and SPN speaker models are computed using a Hamming window function, with a time-frame duration of 32 ms (512 discrete-time samples) and a time-frame shift of 16 ms (256 discrete-time samples). The 257-point single-sided PSD estimate for a time-frame is used and includes both the DC and Nyquist frequency component. The LSSEs are computed from the PSD estimate using 26 triangular-shaped critical band filters spaced uniformly on the mel-scale.

3.2. Classification of reliable spectral components

Here, the reliability of a spectral component is determined by its *a priori* SNR, as in [31]. A component with an *a priori* SNR

[1] For convenience, the time-frame index is omitted from the notation.

Table 1: *ASI accuracy (%) for the real-world non-stationary noise sources. The average improvement over the model in the preceding row is shown in the last column. The highest accuracy for each condition is shown in boldface.*

Model	Marg.	Bounds	SNR level (dB)										Average impr.
			Voice babble					Street music					
			-5	0	5	10	15	-5	0	5	10	15	
GMM [29]	✗	✗	**0.00**	**0.00**	0.63	13.02	**50.48**	0.00	**0.00**	0.95	5.40	25.40	-
SPN	✗	✗	**0.00**	**0.00**	**1.59**	**15.56**	50.16	**0.00**	**0.32**	**1.27**	**6.03**	**25.71**	+0.48
GMM [29]	✓	✗	**2.22**	6.35	18.10	46.98	79.37	**4.76**	**10.48**	20.32	37.46	66.35	-
SPN	✓	✗	**2.22**	**7.30**	**19.05**	**50.79**	**83.49**	4.13	**10.48**	**24.13**	**40.95**	**71.43**	+2.16
GMM [29]	✓	✓	**15.24**	29.21	48.57	72.70	89.21	20.63	32.06	54.60	71.11	85.40	-
SPN	✓	✓	14.60	**32.70**	**55.87**	**77.78**	91.43	**22.54**	**34.92**	**59.37**	**74.29**	90.16	+3.49
SincNet [15] + IRM [30]	-	-	0.63	4.44	25.40	71.75	92.70	1.27	5.40	23.81	64.44	**92.38**	-17.14
SincNet [15]	-	-	0.32	1.59	18.10	56.83	**93.02**	0.63	2.86	11.11	46.98	85.40	-6.54
Xie2019 [16] + IRM [30]	-	-	0.63	1.27	10.48	28.89	53.33	0.32	1.27	4.44	20.63	40.95	-15.46
Xie2019 [16]	-	-	0.32	0.95	4.13	14.92	41.27	0.00	0.32	2.54	13.65	35.56	-4.85

greater than 0 dB is classified as reliable [32]. Deep Xi-ResNet from [33] is used here as the *a priori* SNR estimator. Deep Xi is a deep learning approach to *a priori* SNR estimation [34], and is available at: `https://github.com/anicolson/DeepXi`. It estimates the *a priori* SNR for each of the 257 frequency-domain components of a noisy speech time-frame. The *a priori* SNR estimate for each subband is subsequently found by applying the filterbank used to compute the LSSEs.

3.3. Training and test sets

The TIMIT corpus [35] (16 kHz, single-channel), which consists of 630 speakers with 10 utterances each, is used as the clean speech. The si^* and sx^* subsets are used for training (5 040 utterances) and the sa^* subset is used for testing (1 260 utterances). Each clean speech recording from the sa^* subset is mixed additively with one of four real-world noise source recordings to create the noisy speech for testing (315 clean speech recordings for each noise source). Each noisy speech recording is replicated at five SNR levels: $\{-5, 0, 5, 10, 15\}$ dB, forming a test set of 6 300 noisy speech recordings. The real-world noise sources include two non-stationary and two coloured. The two real-world non-stationary noise sources include *voice babble* from the RSG-10 noise dataset [36] and *street music* (recording no. 26 270) from the Urban Sound dataset [37]. The two real-world coloured noise sources include *F16* and *factory* (welding) from the RSG-10 noise dataset [36].

3.4. ASI systems

GMM: For each speaker, a GMM consisting of 48 diagonal-covariance clusters is trained on the training set using the expectation-maximisation (EM) algorithm [38], and the k-means++ algorithm for parameter initialisation [39].

SincNet: [15] is available at: `https://github.com/mravanelli/SincNet` and is trained using the training set with default hyperparameters.

Xie2019: [16] is available at: `https://github.com/WeidiXie/VGG-Speaker-Recognition` and is trained using the training set with default hyperparameters and a 1 second input spectrogram size.

SincNet + IRM & Xie2019 + IRM: The LSTM-IRM estimator from [30] is used as the front-end for SincNet and Xie2019. The training data and configuration from [40] is used specifically.

SPN: Each speaker is modelled using an SPN with univariate Gaussian leaves. The SPFlow library is used to implement the SPN speaker models [41]. A variant of the LearnSPN algorithm [28] that partitions and clusters variables using the Hirschfeld-Gebelein-Rényi maximum correlation coefficient [42] is used as the structure learning algorithm. The minimum number of instances to split is set to 50 and the threshold of significance is set to 0.3 for the structure learning algorithm.

4. Results and discussion

4.1. Real-world non-stationary noise sources

Table 1 shows the ASI accuracy for two real-world non-stationary noise sources: *voice babble* and *street music*. Over all of the tested conditions in Table 1, SPN speaker models demonstrated an average improvement of 0.48% over GMM speaker models (no marginalisation). This indicates that SPN speaker models are better able to model the joint distribution of each speaker's features. It can be seen that the robustness of SPN speaker models increases significantly when either marginalisation or bounded marginalisation is used. SPN speaker models attained an average improvement of 2.16% and 3.49% over GMM speaker models when marginalisation and bounded marginalisation are used, respectively. The performance improvement that SPN speaker models posses over GMM speaker models is thus extended when either marginalisation or bounded marginalisation is used.

SPN speaker models employing bounded marginalisation are able to outperform SincNet + IRM, with an average improvement of 17.14%. While SincNet + IRM achieved the best accuracy at 15 dB for both non-stationary noise sources, it is outperformed at lower SNR levels by SPN speaker models employing bounded marginalisation. The results presented in Table 1 show that SPN speaker models are robust to real-world non-stationary noise sources when marginalisation and bounded marginalisation is used, especially at lower SNR levels.

4.2. Real-world coloured noise sources

Table 2 shows the ASI accuracy for two real-world coloured noise sources: *F16* and *factory*. Over all of the tested conditions, SPN speaker models demonstrated an average improvement of 1.33% and 2.66% over GMM speaker models when marginalisation and bounded marginalisation are used,

Table 2: *ASI accuracy (%) for the real-world coloured noise sources. The average improvement over the model in the preceding row is shown in the last column. The highest accuracy for each condition is shown in boldface.*

Model	Marg.	Bounds	SNR level (dB)										Average impr.
			F16					Factory					
			-5	0	5	10	15	-5	0	5	10	15	
GMM [29]	✗	✗	**0.32**	**0.32**	**0.95**	0.95	10.16	**0.63**	**1.27**	**0.63**	1.90	12.06	-
SPN	✗	✗	**0.32**	**0.32**	0.32	**2.54**	**14.92**	**0.63**	0.63	**0.63**	**2.54**	**13.65**	+0.73
GMM [29]	✓	✗	**1.90**	7.30	21.27	34.29	58.73	**3.17**	5.71	10.79	25.40	53.65	-
SPN	✓	✗	**1.90**	**10.16**	**21.59**	**34.60**	**59.37**	2.54	**6.35**	**14.29**	**28.89**	**55.87**	+1.33
GMM [29]	✓	✓	19.37	35.24	46.98	62.54	80.32	**11.75**	18.41	36.83	54.60	81.90	-
SPN	✓	✓	**22.54**	**36.83**	**49.84**	**66.35**	**81.90**	10.48	**21.59**	**39.68**	**56.83**	82.54	+2.06
SincNet [15] + IRM [30]	-	-	0.63	1.27	5.71	26.67	72.70	0.95	1.59	13.02	44.13	**86.67**	-21.52
SincNet [15]	-	-	0.32	0.63	4.13	16.19	57.78	0.00	0.95	5.71	35.56	78.41	-5.37
Xie2019 [16] + IRM [30]	-	-	0.32	0.32	2.86	6.98	20.00	0.00	0.32	0.63	2.86	21.27	-14.41
Xie2019 [16]	-	-	0.32	0.63	3.17	7.62	21.90	0.95	0.63	1.27	5.71	26.67	+1.33

Table 3: *Average number of parameters used by each ASI system for each of the 630 speakers.*

	SPN	GMM	Xie2019	SincNet
Params. per speaker	2 502	2 544	13 545	36 718

respectively. This indicates that marginalisation and bounded marginalisation are more suited to SPN speaker models than GMM speaker models. SPN speaker models utilising bounded marginalisation were also able to outperform SincNet + IRM, with an average performance increase of 21.52%.

The results presented in Tables 1 and 2 show that SPN speaker models are robust to both real-world non-stationary and coloured noise sources when marginalisation or bounded marginalisation is used. The number of parameters that each ASI system expends on a speaker is specified in Table 3. SPN speaker models are more robust than SincNet, whilst employing 14.7 times fewer parameters on average per speaker. This exhibits the parameter efficiency of SPN speaker models.

4.3. Future direction

In this work, standard SPNs are used as speaker models. The structure learning algorithm used to find each SPN speaker model is LearnSPN (introduced in 2013) [28], which was the second-ever proposed. An increase in performance can likely be realised by utilising more advanced SPN architectures, such as random and tensorised SPNs (RAT-SPNs) [43], DGC-SPNs [27], or dynamic SPNs [26]. Such SPN architectures use predefined structures (i.e. no structure learning algorithm is required) and can be trained discriminatively using modern stochastic gradient descent optimisation algorithms [44]. Alternatively, they can be trained generatively using hard EM [22]. These capabilities are to be made available to researchers through toolkits such as LibSPN [24]. In this work we investigate SPNs and marginalisation on a simple robust ASI task. ASI was chosen as the task to demonstrate the robustness capabilities of SPNs, as a system could be quickly developed. The results presented in this work lead to more complicated robust speech processing tasks being investigated in future work (complicated in the sense that more system development is required). Such tasks include robust ASR and ASV.

5. Conclusion

SPNs utilising marginalisation are proposed for robust ASI. They are evaluated using real-world non-stationary and coloured noise sources at multiple SNR levels. It was found that SPN speaker models and marginalisation are more robust than two recent CNN-based ASI systems that employ significantly more parameters. With the development of new toolkits and architectures, SPNs and marginalisation are predicted to have a bright future in robust ASI, as well as robust ASR and ASV.

6. References

[1] Z. Zhang, "Mechanics of human voice production and control," *The Journal of the Acoustical Society of America*, vol. 140, no. 4, pp. 2614–2635, 2016.

[2] Y. Tu, J. Du, L. Dai, and C. Lee, "A speaker-dependent deep learning approach to joint speech separation and acoustic modeling for multi-talker automatic speech recognition," in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2016, pp. 1–5.

[3] T. J. Park and P. Georgiou, "Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks," in *Proc. Interspeech 2018*, 2018, pp. 1373–1377.

[4] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1, pp. 91 – 108, 1995.

[5] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 2, 2003, pp. 53–56.

[6] R. J. Mammone, Xiaoyu Zhang, and R. P. Ramachandran, "Robust speaker recognition: A feature-based approach," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 58–71, 1996.

[7] B. Raj and R. M. Stern, "Missing-feature approaches in speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 101–116, 2005.

[8] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, no. 3, pp. 267 – 285, 2001.

[9] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23–61, 2011.

[10] A. Nicolson, J. Hanson, J. Lyons, and K. Paliwal, "Spectral subband centroids for robust speaker identification using marginalization-based missing feature theory," *International Journal of Signal Processing Systems*, vol. 6, no. 1, pp. 12–16, 2018.

[11] A. Nicolson and K. K. Paliwal, "Bidirectional long-short term memory network-based estimation of reliable spectral component locations," in *Proc. Interspeech 2018*, 2018, pp. 1606–1610.

[12] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[14] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[15] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021–1028.

[16] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5791–5795.

[17] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[18] Y. Zhong, R. Arandjelović, and A. Zisserman, "GhostVLAD for set-based face recognition," in *Computer Vision – ACCV 2018*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 35–50.

[19] M. I. Mandasari, M. McLaren, and D. A. van Leeuwen, "The effect of noise on modern automatic speaker recognition systems," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4249–4252.

[20] I. Bisio, C. Garibotto, A. Grattarola, F. Lavagetto, and A. Sciarrone, "Smart and robust speaker recognition for context-aware in-vehicle applications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8808–8821, 2018.

[21] Z. Zhang, J. Geiger, J. Pohjalainen *et al.*, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 5, Apr. 2018.

[22] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 689–690.

[23] P. Jaini, A. Ghose, and P. Poupart, "Prometheus: Directly learning acyclic directed graph structures for sum-product networks," in *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, ser. Proceedings of Machine Learning Research, V. Kratochvíl and M. Studený, Eds., vol. 72. Prague, Czech Republic: PMLR, 11–14 Sep 2018, pp. 181–192.

[24] A. Pronobis, A. Ranganath, and R. P. N. Rao, "LibSPN: A library for learning and inference with sum-product networks and TensorFlow," in *Presented at the ICML 2017 Workshop on Principled Approaches to Deep Learning*, 2017, p. 807–814.

[25] M. Abadi, A. Agarwal, P. Barham *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[26] M. Melibari, P. Poupart, P. Doshi, and G. Trimponias, "Dynamic sum product networks for tractable inference on sequence data," in *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, ser. Proceedings of Machine Learning Research, A. Antonucci, G. Corani, and C. P. Campos, Eds., vol. 52. Lugano, Switzerland: PMLR, 06–09 Sep 2016, pp. 345–355.

[27] J. van de Wolfshaar and A. Pronobis, "Deep generalized convolutional sum-product networks for probabilistic image representations," *arXiv:1902.06155 [cs.LG]*, 2019.

[28] R. Gens and D. Pedro, "Learning the structure of sum-product networks," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 873–880.

[29] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[30] J. Chen and D. Wang, "Long short-term memory for speaker generalization in supervised speech separation," *The Journal of the Acoustical Society of America*, vol. 141, no. 6, pp. 4705–4714, 2017.

[31] J. Barker, L. Josifovski, M. Cooke, and P. Green, "Soft decisions in missing data techniques for robust automatic speech recognition," in *In ICSLP-2000*, 2000, pp. 373–376.

[32] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Boston, MA: Springer US, 2005, pp. 181–197.

[33] Q. Zhang, A. M. Nicolson, M. Wang, K. Paliwal, and C. Wang, "DeepMMSE: A deep learning approach to MMSE-based noise power spectral density estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[34] A. Nicolson and K. K. Paliwal, "Deep learning for minimum mean-square error approaches to speech enhancement," *Speech Communication*, vol. 111, pp. 44 – 55, 2019.

[35] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, 1993.

[36] H. J. Steeneken and F. W. Geurtsen, "Description of the RSG-10 noise database," *Report IZF 1988-3, TNO Institute for Perception, Soesterberg, The Netherlands*, 1988.

[37] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1041–1044.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[39] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford InfoLab, Technical Report 2006-13, June 2006.

[40] A. Nicolson and K. K. Paliwal, "Deep Xi as a front-end for robust automatic speech recognition," *arXiv:1906.07319 [eess.AS]*, 2019.

[41] A. Molina, A. Vergari, K. Stelzner *et al.*, "SPFlow: An easy and extensible library for deep probabilistic learning using sum-product networks," *arXiv:1901.03704 [cs.LG]*, 2019.

[42] A. Molina, A. Vergari, N. D. Mauro *et al.*, "Mixed sum-product networks: A deep architecture for hybrid domains," in *AAAI Conference on Artificial Intelligence*, 2018.

[43] R. Peharz, A. Vergari, K. Stelzner *et al.*, "Probabilistic deep learning using random sum-product networks," *arXiv:1806.01910 [cs.LG]*, 2018.

[44] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747 [cs.LG]*, 2016.