Speaker Counting Model based on Transfer Learning from SincNet Bottleneck Layer

Wei Wang , Fatjon Seraj, Nirvana Meratnia, Paul J.M. Havinga
Pervasive Systems Group
University of Twente, Enschede, The Netherlands
Email: {w.wang1,f.seraj, n.meratnia, p.j.m.havinga}@utwente.nl

***Abstract*—People counting techniques have been widely researched recently and many different types of sensors can be used in this context. In this paper, we propose a system based on a deep-learning model able to identify the number of people in the crowded scenarios through the speech sound. In a nutshell the system relies on two components: counting concurrent speakers in overlapping talking sound directly and clustering single-speaker sound by speaker-identity over time. Compared to previously proposed speaker-counting systems models that only cluster single-speaker sound, this system is more accurate and less vulnerable to the overlapping sound in the crowded environment. In addition, counting speakers in overlapping sound also gives the minimal number of speakers so that it also improves the counting accuracy in a quiet environment.**

Our methodology is inspired by the newly proposed *SincNet* deep neural network framework which proves to be outstanding and highly efficient in sound processing with raw signals. By transferring the bottleneck layer of *SincNet* model as features fed to our speaker clustering model we reached a noticeably better performance than previous models who rely on the use MFCC and other engineered features.

I. Introduction

A. Motivation

The people counting technique has become very popular and widely researched topic in substantial human-centric IoT applications [1]. Knowing the number and mobility pattern of customers can help retail businesses adjust their marketing plan [2]. In shopping malls renters can decide the rent charges based on the areas of concentration and number of people [3]. In smart buildings, knowing the number of residents at specific time-frames can help save utility cost from a smart HVAC(heating, ventilation and air conditioning) controlling system [1]. People counting problem can be addressed using a single or a multitude of sensors and each approach has its pros and cons. For example, in small environments such as a room, people can be counted using cheap PIR binary sensors which are favourable for their price but falls short in accuracy [4]. While, computer vision techniques can count people in real time, have the line of sight drawback and raise privacy concern [5]. WiFi or BLE receivers can also count people passively through monitoring broadcast beacons from smartphones, they fell short if not everyone carries a smartphone and the distances between broadcasters and receivers is difficult to estimate [6].

In this paper, we propose a people counting system using text-independent human speech. We count the number of people in a specific location, according to individual voices regardless of the speech content. Speech sound has a pervasive nature and posses large amount of information. Whats more, microphone is relatively cheap among all the sensors used for human counting. On the other hand, sound has disadvantages, especially in quiet environments, like auditoriums where the number of speakers is less than the total number of persons. To clarify, this work addresses the scenarios with a equal distribution of speaking persons.

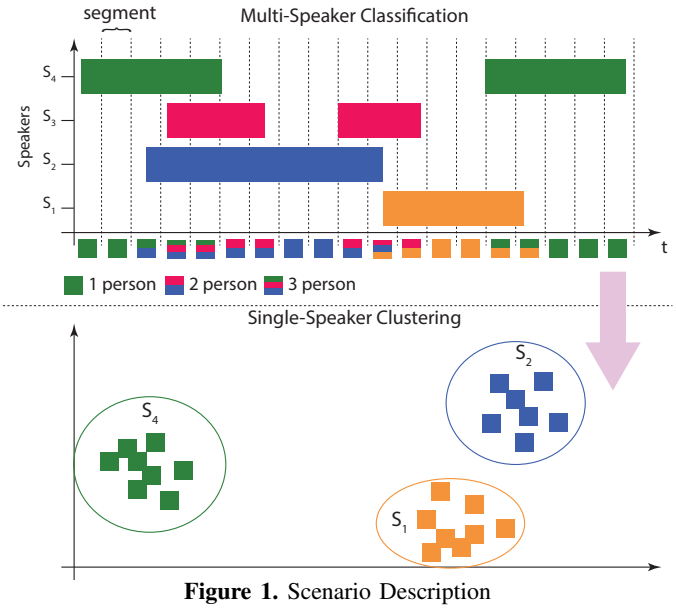

Figure 1. Scenario Description

To achieve this we address two related challenges: First, we estimate how many people are speaking at each time segment (*e.g. 2sec*). For this we propose a classification model that yields the number of speakers even when voices overlap. Second, all the classified single-speaker segments from the previous step are clustered into identity clusters. For this, an unsupervised single-speaker clustering model is proposed, based on the identity-correlated feature so that the number of clusters corresponds to the number of speakers. This approach is more robust than previously proposed cluster based counting models [7][8], neglecting the uncertainties inherited by crowded environments. Nevertheless, counting speakers when voices are overlapped still provides us with a useful information regarding the minimum number of speakers, that can further increase the counting accuracy. Figure 1 shows an

978-1-7281-4657-7/20/$31.00 ©2020 IEEE

overview of how this two combined approaches provide an estimate of people in that precise environment.

B. Related works

One can use multi-channel microphones for both counting and identifying people, using the DoA(direction of arrival) principle to decompose and classify mixed signals as individual person. Mirzaei et al. [9] counted people by evaluating the differences of phase and amplitude between received spectrums by microphones. Walter et al. [10] counted multiple sound sources with a Bayesian infinite Gaussian mixture model. Signal decomposition requires specific signal processing methods specific to individual task, making these models efficient only in quiet environment with few people.

Most recently, Stoter et al. [11] proposed a complicated RNN(recurrent neural network) model on the same topic and achieved better performance based on huge training dataset. The concurrent speaker counting model however could not well proximate the real number of people, since not everybody speaks simultaneously. Moreover, the number of speakers becomes hard to estimate when it reaches a certain level (it is difficult to distinguish between 11 and 12 speakers).

Friedland [8] and Xu et al. [7] estimate the number of speakers through clustering single-speaker sounds using MFCC and other engineered features. These models however did not filter out multi-speaker sounds. Thus, are likely to falsely cluster multi-speaker sound segments as new speakers. Furthermore, the features used can not perfectly represent the speaker's identity so that the feature-clusters would not equal to the speaker-clusters.

To address and fill this gap, we propose a hybrid system that combines the principles researched for multi-speaker detection and clustering [11][7][8], for a more realistic counting scheme. This system relies on multi-speaker counting model to split the single-speaker sound and count the persons in multi-speaker sound segments. While all the single-speakers are then clustered to give an estimation of unique number of people.

In order to cluster audio segments according to the speaker-identity, selecting the right features that highly correlate with the speaker's voice is of paramount importance. However, engineered features used in [8][7] are not specifically designed for speaker identification, and one can imagine the MFCC features of the same speaker might alter dramatically along with different contents or contexts. A more popular engineered feature that correlates with speaker identity is I-vector, applied in smart voice applications to verify the host's voice from a random speech pool [12]. The I-vector feature however needs the prior knowledge of speakers and is mainly used for verify a certain speaker's voice which is not suitable in our case.

To get the good input features for our classification and clustering, we use the Transfer Learning paradigm to use the weights from the bottleneck layer *SincNetBN*, calculated by *SincNet* speaker recognition model. *SincNet* is a novel CNN-based architecture which accepts raw audio input and encourages the learning of meaningful audio-specific filters [13]. SincNet is designed on the idea of implementing multiple parametric sinc filters with deep neural network. In contrast to a standard CNN, that learns all elements of each filter, SincNet only learns low and high cutoff frequencies from the data. This offers a very compact and efficient way to derive a customized filter bank specifically tuned for the desired application. In experiments conducted on the speaker recognition task with very few training data($<$ 15 seconds per each person), SincNet outperformed other models in both accuracy and efficiency [13].

We evaluated our system using TIMIT [14] and Libricount [15] with a total more than 800 minutes of 500 different speakers and compared with the baseline method [7]. In our experiments, our single-speaker clustering model which uses the *SincNetBN* feature outperforms the baseline model which uses MFCC and other engineered features.

The rest of this paper is organized as follows: Section II gives an overview of the Transfer Learning paradigm and explain what knowledge we transfer from *SincNet* into our model. Section III describes the methodology used to tackle the problem of distinguishing individual sounds in different contexts, as well as sub-models in each subsection. Section IV describes the dataset used and the experimental results of the sub-models respectively. We conclude this paper with our open discussions in Section V.

II. Transfer learning and *SincNetBN* feature extraction

Representative learning paradigm allows us to reuse the functions and vectors learned by trained deep learning neural networks on other related problems. This methodology is called *Transfer learning*[16]. The concept of *Transfer Learning* is successfully proved by many computer vision works when using the bottleneck layer, normally fully-connected as input features for new tasks alike [17] [18].

A. SincNet Speaker Identification Model

For this purpose we identified *SincNet* network [13] and use the weights extracted from output of the bottleneck layer.

Although clustering speakers without prior knowledge of the speakers is an unsupervised learning problem, we rely on *SincNet* a supervised learning speaker recognition model, especially because it was trained on enormous and representative data and demonstrated a high accuracy. The detailed composition of the *SincNet* model for speaker recognition is shown in Figure 2 [13].

Figure 2 describes the speaker recognition model with associated *SincNet* layers, followed by conventional layers [13]. A *SincNet* layer is a modified CNN layer with parameter constraints on the learned filters. Compared to standard CNNs, where every parameter of the filterbank is directly learned, the *SincNet* only learns the low and high cutoff frequencies of band-pass filters.

In audio processing, a standard CNN layer performs as the impulse response filters (FIR) on the time domain waveform.

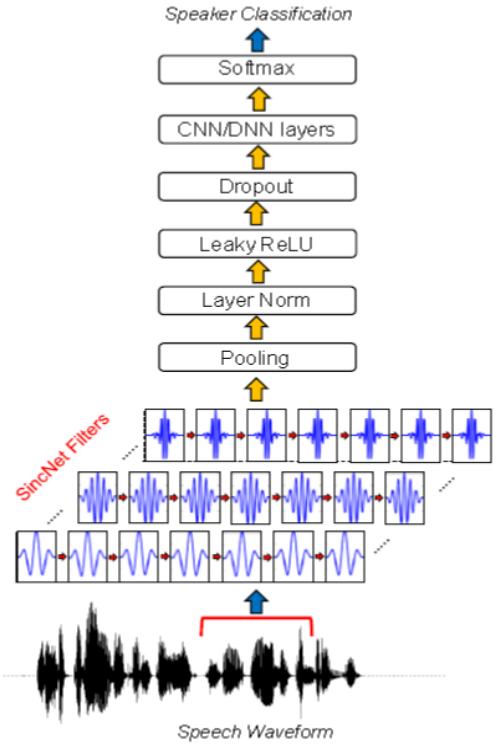

Figure 2. SincNet model of speaker classification

One CNN layer consists of many filters of the same size and for each filter in a CNN layer we have:

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \quad (1)$$

, where x, y, h are the input, output and filter respectively and L is the filter size. During training, all parameters of the filter h needs to be trained. Mathematically, the filters learned by CNN often take noisy and incongruous multi-band shapes which is not an issue for solving a general problem but do not appear to be an efficient representation of the audio signal.

A *SincNet* however is based on parametrized sinc functions and a trained *SincNet* layer stands for a set of band-pass filters of which only the low and high cutoff frequencies are to be learned. For a *SincNet* layer, the input x is convolved with a predefined function g which has two learnable parameters:

$$y[n] = x[n] * g[n, f_1, f_2]$$

, where g has the form:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n)$$

and when transformed to frequency domain(using Fourier transform) g has the shape of a rectangular band-pass filter:

$$G[f, f_1, f_2] = rect\left(\frac{f}{2f_2}\right) - rect\left(\frac{f}{2f_1}\right)$$

, where $sinc(x) = sin(x)/x$ and $rect$ is the rectangular function. The only parameters to be learned from a filter are f_1 and f_2, which are the low and high cut-off frequency of the band-pass filter g.

As the ideal band-pass filter requires an infinite number of elements L and g is with limited length, a windowing function is also applied to smooth out the abrupt discontinuities at the stop band. So eventually, g becomes:

$$g_w[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n]$$

, where w is the Hamming window function.

With this scheme of predefined filter g, this *SincNet* layer could automatically learn a set of band-pass filters which are much more meaningful and interpretable than standard CNN for audio processing problems.

B. SincNetBN feature extraction

SincNet splits long audio segments into $200ms$ frames and use majority voting over the whole segment results after soft-max layer, to decide the speaker identity. The weights of the bottleneck layer (previous to the soft-max layer) can be deemed as a speaker-identity associated feature.

Originally, its bottleneck layer which is full connected has 2048 units, which means the feature length would be 2048 if we use the bottleneck layer weights as our feature. To keep the feature length as short as possible, we retrained the *SincNet* model with different bottleneck layer units.

To use these weights as the feature for clustering an audio segment, we also need to adapt the length difference. The weights or feature extracted from $200ms$-frames can not be directly used for our clustering model since we need one feature per-segment only while a segment has multiple frames.

To both unify the length and compress the data, we propose the *SincNetBN* feature which uses the statistic (i.e. mean and variance) of all frame features as the global feature that represents each audio segment. Let ft_i be the bottleneck layer weights of frame i, the unified feature of the entire audio segment is:

$$ft = [Mean(ft_i), Cov(ft_i)]$$

With the 512-length weights extracted from each $200ms$ frame, any audio segment would have the feature length of 1024 regardless of the segment duration.

As the short-frame feature correlates with the speaker-identity, the statistic feature *SincNetBN* not only unifies the feature length of audio segments, but also inherits the information about the number of speakers in an overlapping sound. Thus, it can be also used in the multi-speaker classification task.

III. Methodology

This section describes the methodology and integration of aforementioned *SincNetBN* into our model. We first give a brief overview of the speaker-counting model and explain in detail all sub-components.

A. System overview

The flow-chart of the entire system is shown in Figure 3. Our system takes the audio stream as the input, and outputs the speaker number over time. Altogether it consists of 4 sub-functions as depicted:

1) segmentation,
2) feature extraction,
3) multi-speaker sound classification,
4) people clustering and counting.

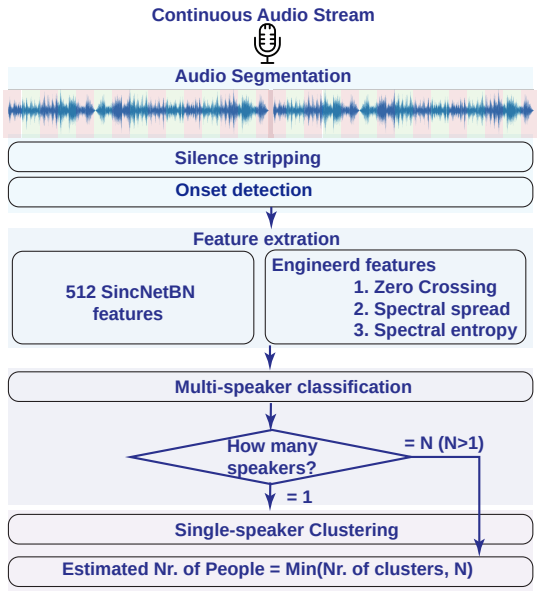

Figure 3. People counting system flow

Audio segmentation refers to recognizing and extracting the duration of speech sound from background noise (or silence) in a continuous audio stream and cut it into small segments. We use a 'flexible-length' segmentation scheme which first detects the onset and the end of a speech sound which works as: (1) The audio stream is smoothed in time domain, and cut into fixed short frames (20ms) while the power is calculated for each frame. (2) The frames with power higher than a threshold are labeled as 'active'. The threshold can be preset static value or dynamic adjusting. (3) Adjacent 'active' frames are combined to form a segment. (4) Segments shorter than a given duration are dropped, while segments longer than a given threshold are split by needs. An example of this algorithm is shown in Figure 4.

Speech containing segments are then passed through *SincNet* to get the bottleneck layer weights as features both for multi-speaker classification and single-speaker clustering. The multi-speaker classification model labels each audio segment with the estimated number of speakers, from $1 \sim 10$. The audio segments with only a single speaker are then clustered based on the *SincNetBN* features where each cluster stands for a different person. The final estimated number of people is the minimal number of speakers from the multi-speaker classification model and the cluster number. The details of each function are depicted in following subsections.

B. Multiple-speaker sound classification

This model inputs the features extracted from section II and outputs the number of speakers. The novelty consists on using and combining *SincNetBN* features with other engineered audio features to increase the performance. These additional

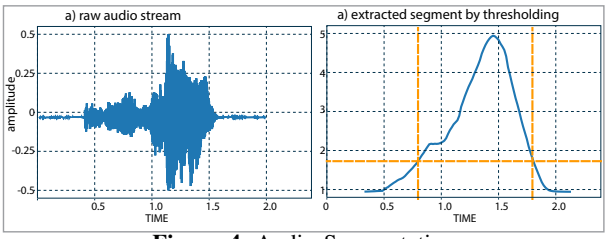

Figure 4. Audio Segmentation

audio features are engineered from frequency domain and listed as follows:

(i) **Zero crossing rate** (ZCR) is the rate at which the signal changes from positive to negative or vice versa. This feature is one of the simplest and widely used in speech recognition, which characterizes the dominant frequency of signal[19].

(ii) **Spectral spread** is the magnitude-weighted average of the differences between the spectral components and the spectral centroid, together they describe how disperse and wide the frequency bands are [20].

(iii) **Spectral entropy** is calculated as the entropy of spectrum it reflects the flatness but spectrum [21].

As described in section II, *SincNetBN* already inherits information about the number of speakers in an overlapping sound which makes the classification model simpler and more robust. As a result, we used two different classification models in this task: SVM and a 2-hidden layer full-connected neural network, both of these two classifiers are light weighted compared to the RNN model used in [11]. We use the RBF(radial basis function) kernel for the SVM model as it is most commonly for non-linear problems. Our DNN model is shown in Figure 5, the output layer is a softmax layer which contains 10 nodes corresponding to $1 \sim 10$ persons.

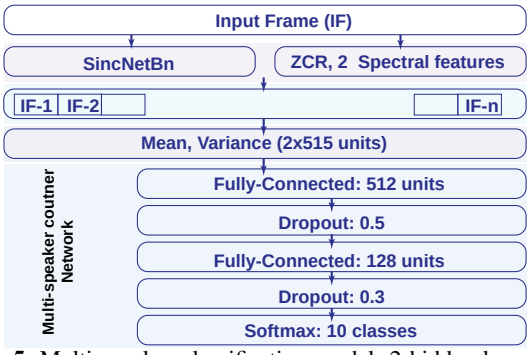

Figure 5. Multi-speaker classification model: 2-hidden-layer DNN

C. Single-speaker sound clustering

As has been discussed previously, concurrent speaker-number in the multi-speaker sound is far from a perfect approximation of the total occupants number. The sheer multi-speaker counting has also over simplified the problem that it keeps no memory of each individual's existence, models like this would make little help to most applications. A better and more appropriate method is therefore to identify and remember 'who is speaking'. With this speaker-identity 'remembered'

model, the number of speakers can be approximated more accurately and viewed dynamically over time.

The single-speaker clustering model inputs the *SincNetBN* features and dynamically cluster them over time. Only single-speaker sound is clustered since it is a very difficult task to separate the multi-speaker signal into its sub-components which until now exhibits no perfect solution. A good feature and the corresponding distance(likelihood) function are the two most decisive factor that affects the clustering algorithm. With the *SincNetBN* feature extracted from section II, we mainly compared 2 different likelihood function: Euclidean distance and cosine likelihood.

In order to count people in real time, we run the clustering algorithm whenever new sample comes. Therefore our 'grouping' algorithm should be light-weighted to run on the fly. The intuition of our algorithm of counting people is inspired from [7], which is basically heuristic and straight-forward:

1) Let G_m be the set of samples(audio segments) that belong to speaker-group m, where $m = 1, 2...M$ where M being the group number. Let C_m be the center of the group G_m. The initial speaker-group number M is 0.
2) When a new sample s comes in, calculate the likelihood cost of s to all existing sample-group centers C_m, where we denote as d_m.
3) According to the value of d_m and two constants (θ_0, θ_1) from experimental results, there are three different cases:
   a) If $min(d_m) > \theta_0, \forall m$, this sample is too far from every group, we should create a new group for this sample.
   b) If $min(d_m) < \theta_1, \forall m$, this sample then belongs to the group with the minimal distance. We next put this sample to the corresponding group and update G_m, C_m.
   c) If $\theta_1 < min(d_m) < \theta_0$, we are not confident about which group this sample belongs to, we should discard this sample.

With the calculated N from multi-speaker sound classification and the speaker-groups M, we estimate the speaker number as $Max(M, N)$. In realtime applications that require information about the number of persons in a certain time frame, e.g. the people in the last 5 minutes, this requirement can be simply fulfilled by aging the samples kept in the speaker-clusters G_n.

IV. Experimental results

In order to evaluate the model we use a precise evaluation methodology not only statistical methods but also heuristic ones. The performance is evaluated using independent dataset, while visualization tools are used to visualize the clusters.

This section presents the experimental results which justify our methodology for the entire system and each sub-model.

A. Dataset

We mainly use two public speech dataset in our experiments: TIMIT [14] and Libricount [15]. The TIMIT dataset is mainly used for model training and Libricount is mainly used for the final evaluation [15].

TABLE I: dataset used for evaluation

	SincNetBN training	**Multi-speaker model**	**Single-speaker model**	**Final evaluation**
Nr. Speakers	300	100	40	40
Nr. Minutes	200	1000	100	480
Source	TIMIT	TIMIT	TIMIT	Libricount

The TIMIT corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. The dataset contains a total of 3696 sentences, 8 sentences spoken by each of 462 speakers of which the gender is nearly equal. We deliberately separate the 462 speakers into 2 parts, each is used for the training of SincNetBN features and the multi-speaker sound classification model.

The Libricount dataset was released as a synthetic dataset for speaker count estimation. This dataset contains a simulated cocktail party environment of $0 \sim 10$ speakers, mixed with 0dB SNR from random utterances of different speakers from the LibriSpeech 'CleanTest' dataset [22].

The concrete data partition is shown in in Table I. 100 speakers data are randomly picked from TIMIT and shuffled for training the multi-speaker classification model. Doing this we ensure that the model training and final evaluation use completely different datasets and prevent data pollution.

B. SincNetBN feature extraction

This section gives the comparative results of using different bottleneck layer units(neurons) in a SincNet speaker recognition model. The original bottleneck layer in [13] has 2048 units and achieved very good performance. We gradually reduce the units and retrain the model since less feature length is better for the later clustering model. Figure 6 gives the speaker recognition accuracy with different bottleneck units (keeping other layers exactly the same). For comparison, we also put in the results of MFCC feature together with a GMM model as the benchmark, which is a classic speaker recognition model [23]. As one can see, units of 512 is the best choice as it performs equally good as of 1024 while units of 256 slides a big step in performance.

We also presented the distribution of the bottleneck layer weights in Figure 7. As the output-layer predictor can be roughly seen as the linear combination of bottleneck layer weights, this means a tiny altering of the bottleneck layer (feature) would not change the speaker-identify (output). With this trained SincNet model we define *SincNetBN* feature as the the $[mean, variance]$ of the bottleneck layer weights on all frames, which has the length of 1024.

C. Multi-speaker sound classification

In this section, we classify the number of speakers in the multi-speaker sound. In our data, the signals of multi-speaker sound are all normalized otherwise the sheer amplitude could easily correlates speaker number which however would not be helpful in real applications. We experimented with two

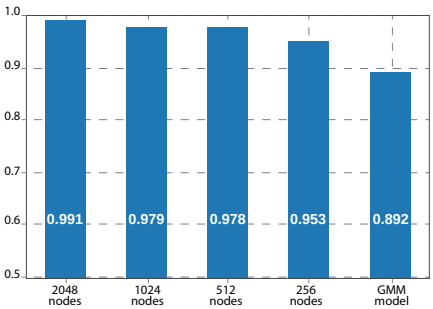

Figure 6. Speaker-recognition accuracy of SincNet with different bottleneck layer units, compared with classic GMM model

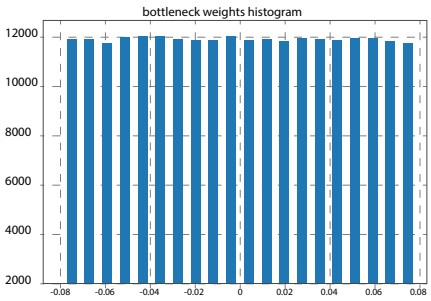

Figure 7. Bottleneck layer weights distribution for 512 bottleneck nodes

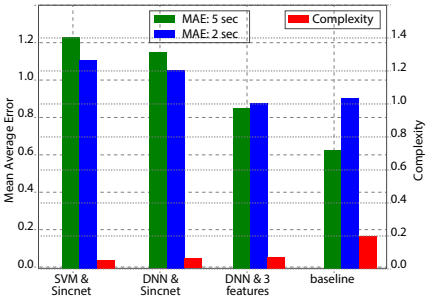

Figure 8. Multi-speaker sound classification

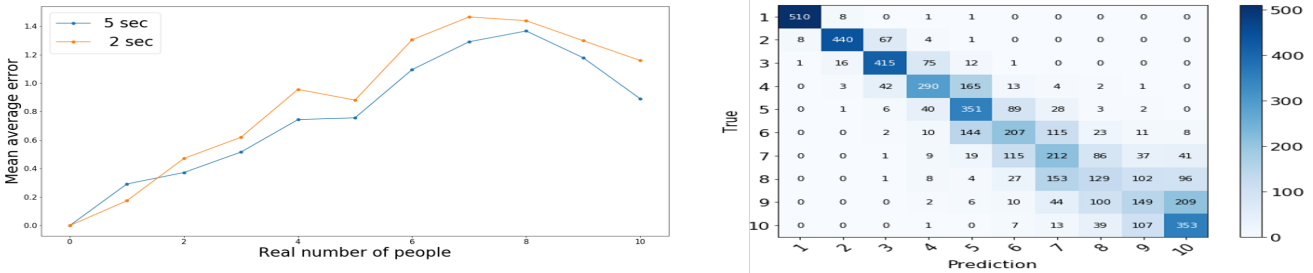

Figure 9. Detailed multi-speaker classification results: (left) MAE of different person number (right) Confusion matrix on 5-seconds audio

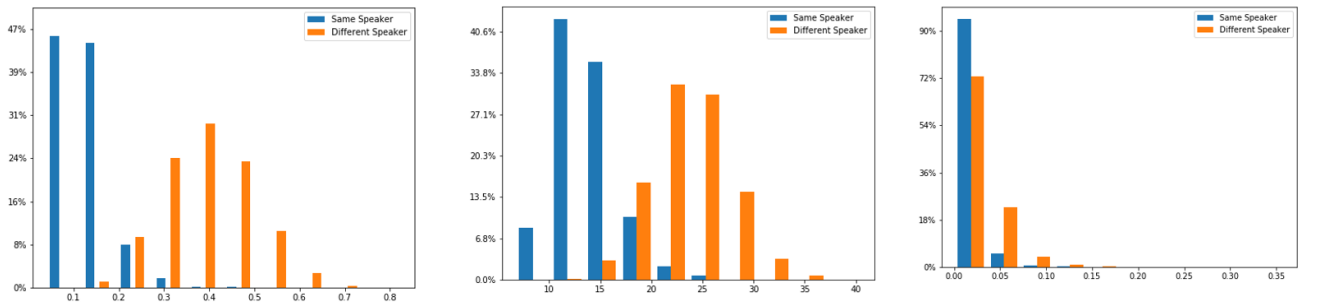

Figure 10. Density distribution of feature-distances from same and different speakers: left=SincNet+Cosine, middle=SincNet+Euclidean, right=MFCC+Cosine

different classification models: SVM and a normal DNN model with two full-connected hidden layers. We used the RNN model used in [11] as the baseline model, which takes the STFT spectrogram as the input feature. In order to test the model flexibility, we also prepared test audios with two groups of different audio-length: 5 seconds and 2 seconds. Apart from the SincNetBN feature, we also added three more simple features to achieve a better performance.

For each model and feature combination, we tested both the performance and the complexity. We used mean average error as the performance metric as it is often used in object counting tasks. Complexity equals to the processing time over the audio duration, which is also important in realtime counting applications.

The results are shown in Figure 8. The sheer SincNet feature, DNN performs a little better than SVM model, but the overall performance is noticeably worse than the baseline method. After adding two more features, DNN could perform equally good as the baseline in 2 seconds audios (0.947 vs 0.972). Our model outperforms the baseline in complexity a lot as RNN is significant slower than a normal DNN model. Another advantage of our model is the flexibility with input audio length as it is trained with statistic features unlike the baseline which always inputs the same length audio (5s).

The detail performance of DNN model and 3 features are shown in in Figure 9. The left shows the MAE on different speaker number and the right chart shows the confusion matrix of 5 seconds audios. We could observe from the charts that the error increases dramatically when people number exceeds 5 persons. It is worth mentioning that the MAE decrease at 8 people simply because the model has an up limit on the output.

D. Single-Speaker clustering and people counting

In this section, we first conducted experiments to justify the ability of SincNetBN feature to differentiate speaker-identity from text independently utterances. As aforementioned, a good

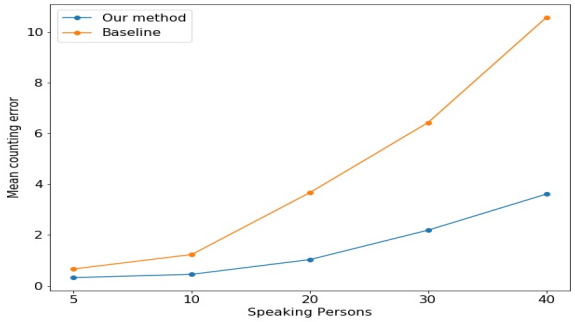

Figure 11. Final evaluation of counting people with long audio stream

speaker clustering model depends on both the used feature and the distance calculation function between features from different segments. We used SincNetBN feature Euclidean and Cosine likelihood as the distance function in the comparison experiments. We also chose Cosine likelihood function with MFCC feature as the baseline [7]. The distribution of all feature-distances are shown in Figure 10. From this figure we could see the MFCC is much less capable than SincNetBN feature in separating speakers as the distance distribution of two groups are mostly overlapping. With SincNetBN feature, the Cosine likelihood function works slightly better than Euclidean since it only calculates the angle difference between two vectors. This statistics shows that 2 audios must belong to the same speaker if their Cosine distance is less than 0.15 and must be from different speakers if the distance is more than 0.4.

We finally tested the MAE of the entire people counting system with the libricount dataset. Figure 11 shows the final results in different experiments from both our model and the baseline in [7]. With the minimal of 5 and maximum of 40 different persons, our method outperforms the baseline in each test, and especially when number of people are bigger.

E. Visualization

We also use t-distributed stochastic neighbor(t-SNE) to visually view the results. t-SNE is a machine learning based dimensionality reduction algorithm which is commonly used for embedding high-dimensional data for visualization in a low-dimensional space[24]. Specifically, it models each high-dimensional object by a low-dimension point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

Figure 12 shows this 2-dimensions t-SNE plot of our clustering model,with different distance functions, compared with the baseline function. From which we could see that 40 speakers can be separated quite well with SincNetBN feature while is hard with MFCC feature.

V. Open discussion and conclusion

Automatic audio signal processing is still a *hot* research topic in artificial intelligence. However, compared to speech recognition, the pervasive environmental sound was overlooked by researchers. To reuse speech recognition techniques in environmental sound recognition, one of the major barriers is the impact of high overlapping sound occurrence rate.

To count people from the speech sound in a crowded environment, we employed a two step system which consider both overlapping and single-speaker sound. The direct classification of multi-speaker sound works quite well when speaker number is small (i.e. < 5) however falls short when the number increases.

On the other hand, the clustering of single-speaker provides a more stable and practical estimation of speaker-number over time. However, this method is more complicated then the previous one and vulnerable to overlapping sound. To tackle this issue, we combined the multi-speaker model with the single-speaker clustering model to achieve better and stable performance in the crowded environment.

Our second contribution is that the SincNet bottleneck layer which highly correlates the speaker-identity can be used in both multi-speaker classification and single-speaker clustering. In the multi-speaker classification experiments, a much simpler full-connected DNN model can achieve comparable results to the RNN baseline model. Moreover this feature helps to dramatically increase the performance of singer-speaker sound clustering compared to the baseline which uses MFCC. In an example t-SNE plot we could see vividly that the sound from different speakers stay much further than from the same speaker.

References

[1] M. Drăgoicea, L. Bucur, and M. Pătraşcu, "A service oriented simulation architecture for intelligent building management," in *International Conference on Exploring Services Science*. Springer, 2013, pp. 14–28.

[2] J. York, P. Dixon, T. C. Opdycke, and W.-C. W. Wu, "Measuring effectiveness of marketing campaigns presented on media devices in public places using audience exposure data," Jan. 29 2009, uS Patent App. 12/233,872.

[3] R. Kohavi, N. J. Rothleder, and E. Simoudis, "Emerging trends in business analytics," *Communications of the ACM*, vol. 45, no. 8, pp. 45–48, 2002.

[4] T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse, "Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 309–322.

[5] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–7.

[6] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using wifi," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 361–369.

[7] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: Unsupervised speaker count with smartphones," in *Proceedings of the 2013*

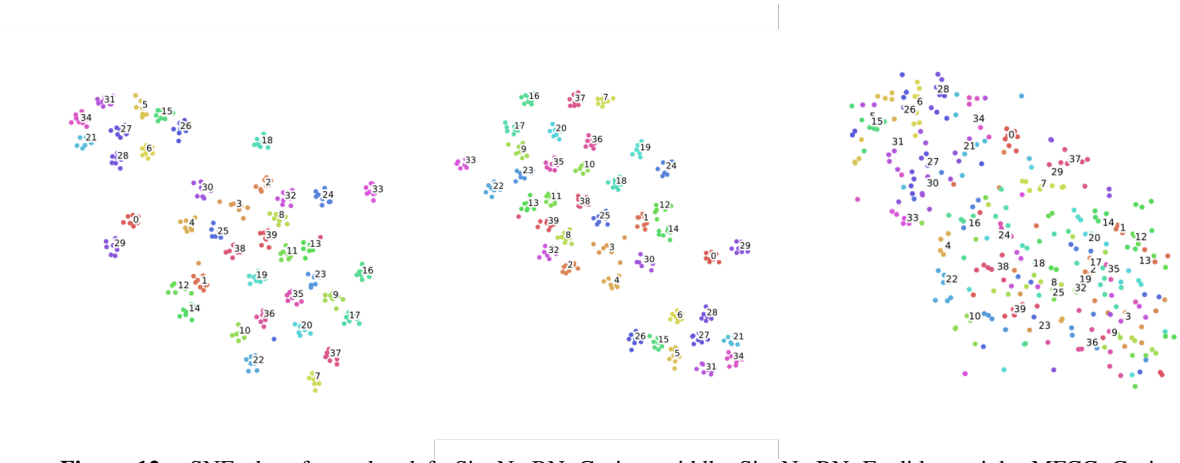

Figure 12. t-SNE plot of samples: left=SincNetBN+Cosine, middle=SincNetBN+Euclidean, right=MFCC+Cosine

ACM International Joint Conference on Pervasive and Ubiquitous Computing, ser. UbiComp '13. ACM, 2013, pp. 43–52, event-place: Zurich, Switzerland. [Online]. Available: http://doi.acm.org/10.1145/2493432.2493435

[8] G. Friedland, H. Hung, and C. Yeo, "Multi-modal speaker diarization of real-world meetings using compressed-domain video features," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 4069–4072.

[9] S. Mirzaei, Y. Norouzi *et al.*, "Blind audio source counting and separation of anechoic mixtures using the multichannel complex nmf framework," *Signal Processing*, vol. 115, pp. 27–37, 2015.

[10] O. Walter, L. Drude, and R. Haeb-Umbach, "Source counting in speech mixtures by nonparametric bayesian estimation of an infinite gaussian mixture model," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 459–463.

[11] F. Stöter, S. Chakrabarty, B. Edler, and E. A. P. Habets, "Classification vs. regression in supervised learning for single channel speaker count estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 436–440.

[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011, 9 - ivector.

[13] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," *arXiv*, 2018, 9 - sincNet. [Online]. Available: http://arxiv.org/abs/1808.00158

[14] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech communication*, vol. 9, no. 4, pp. 351–356, 1990.

[15] F.-R. Stöter, S. Chakrabarty, E. Habets, and B. Edler, "Libricount, a dataset for speaker count estimation," Apr. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1216072

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2016. [Online]. Available: https://books.google.nl/books?id=Np9SDQAAQBAJ

[17] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[18] J. J. Lim, R. R. Salakhutdinov, and A. Torralba, "Transfer learning by borrowing examples for multiclass object detection," in *Advances in neural information processing systems*, 2011, pp. 118–126.

[19] B. Kedem, "Spectral analysis and discrimination by zero-crossings," *Proceedings of the IEEE*, vol. 74, no. 11, pp. 1477–1493, 1986.

[20] D. Mitrović, M. Zeppelzauer, and C. Breiteneder, "Features for content-based audio retrieval," in *Advances in Computers*. Elsevier, 2010, vol. 78, pp. 71–150, 8 - 6 features. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0065245810780037

[21] J. Sueur, S. Pavoine, O. Hamerlynck, and S. Duvail, "Rapid acoustic survey for biodiversity appraisal," *PloS one*, vol. 3, no. 12, p. e4065, 2008.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[23] 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, clear & phone sound - supervised - GMM -.

[24] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.