k-SpecNET: Localization and classification of indoor superimposed sound for acoustic sensor networks

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

***Abstract*—Automatic localization and classification of environmental sound events can provide great aid to many human-centric applications. However as many papers have mentioned, environmental sound events in daily life are complicated and hard to classify especially when multiple sounds happen simultaneously. Being different from many other works, we use an acoustic-sensor-network to solve this problem and decompose overlapping sound events using a sound localization model. The core of our contribution is to first find and locate the keypoints from each microphone's spectrogram and then aggregate them. With these aggregated keypoints as input, we then use 2 different classification models to further classify the type of sound sources. Compared with other classification models that only use single microphone, our experiments show that our solution is both accurate and low-cost in terms of calculation effort.**

***Index Terms*—Environmental sound localization, TDOA, acoustic sensors, superimposed sound, Keypoints localization, Deep Neural Network, Convolutional Neural Networks**

I. Introduction

Human presence and activity recognition has become a popular research topic and refers to identifying the location, movement and action of a person based on information from the surrounding sensors. Indoor human activity information is important because it can be applied to many real-life human-centric problems from health-care of the inhabitants to energy saving in smart buildings [1], [2]. Numerous sensors can be used for human activity recognition. One of the commonly used is the audio sensors mostly because human activities are always accompanied by some sort of sounds. Even though both audio localization and classification are well investigated, most of the existing works treat them as isolated problems. Additionally, because of the difficulties in classifying overlapping sound, the solutions are either focused on single events or overlapping events in very limited scenarios and assumptions. State-of-the-art acoustic localization requires specifically designed hardware, special emitters on the moving targets generate distinctive signals, and microphone arrays capture these sound signals by using the direction-of-sound to calculate the location. These methods achieve a high localization accuracy and are often used in robotic applications [3], [4]. Being obtrusive and expensive, these methods are ill-suited for human-centric applications. Instead acoustic sensor networks are used to discreetly track human activities by leveraging the time-of-arrival (ToA) of the sound from the source. The implementation of these networks vary from simple localization using energy-based threshold [5] to estimate the ToA and applying an audio event classification algorithm [6] for sound classification. Some times the quality of the network does not provide a reliable ToA estimation [7] thus, new algorithms are required to compensate for these uncertainties. Various systems are developed to classify non-speech sound events, some of which are both robust and efficient for single events [8]. Sound event classification is implemented by first extracting feature from the sound signal then performing a feature-based classification with supervised learning algorithms. Because the audio stream is by nature stochastic and large in size, the features are extracted on overlapped frame-basis to compress the data and to preserve the dynamic character of the sound.

Support Vector Machine (SVM) learning algorithms are used to classify non-overlapping sound events [8] where mel frequency cepstral coefficients (MFCC), subband power and several other frequency features are extracted on frame-basis.

When the sound is more complex with overlapping events, three different methods can be used to deal with:
I. The first method decomposes the signal based on matrix factorization by reducing a matrix into its constituent parts. Non-negative matrix factorization (NMF) [9] is used to transform the input audio into four components, where sound events are separated into different components for recognition. Applying additional constraints, such as sparsity on the NMF improves the decomposition [10]. By treating the components as a coupled matrix factorization problem [11], the problem is transformed into a supervised learning that maps the audio class label to each component. NMF based methods have some drawbacks such as the complexity is hard to control when large number of samples also the method predicts the class label for short frames, losing the global characters of the event.
II. The second method extends the frame-based models to overlapping events through hierarchical models. The SVM algorithm [12] is used to classify the extracted speech sound from overlapping events with improved accuracy [13] by transforming the frequency subbands to a new domain making the features easier to be classified. However, the scalability of these methods, for more than two overlapping events, showed to be limited due to the underperformace of the SVM. When SVM is replaced with deep recurrent neural network (RNN) architecture [14] RNN was able to reach a better performances.
III. The third category is based on image-like features ex-

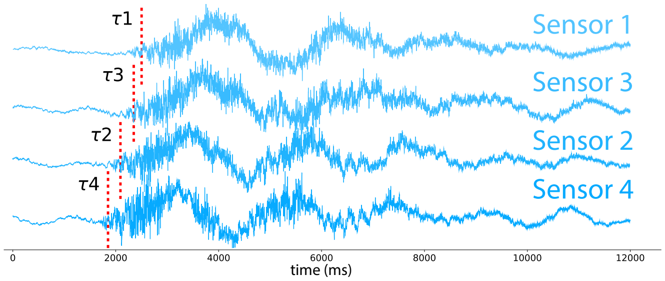

Fig. 1. Single sound event synchronization is easy

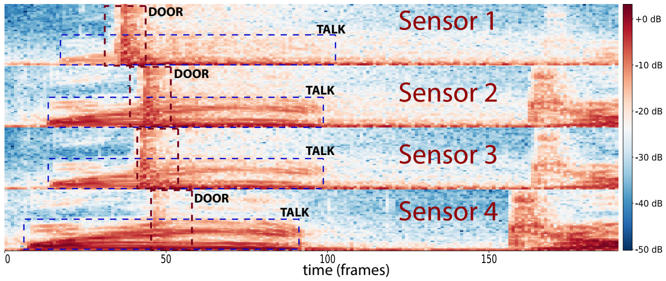

Fig. 2. Spectrograms of the sound received by different sensors

tracted from the sound STFT spectrogram, given the spectrogram can be interpreted as an image [15], [16]. This allows us to classify overlapping events with local spectrogram features where features are extracted from so called 'keypoints' in the spectrogram and Hough transform voting is used to detect all possible sound combinations. Adding subband power distribution (SPD) as an image feature [15] in a KNN algorithm, yields better results for larger number of sound classes. Comparative empirical studies [16] show that spectrogram features and deep learning algorithms, outperform SVM in noisy environments.

In this paper we describe, k-SpecNET, a light-weight and scalable system for localization and classification of overlapping indoor sound events. The major contribution of this system is the use of spectrogram keypoints [17]. Keypoint is a concept used in image processing to detect objects. While it is difficult to split an overlapping sound into its sub-components, we only split and locate the small keypoints which have sufficient information to represent an event when aggregated.

The results show that k-SpecNET works well with randomly mixed 2-events and is scalable for more overlapping events.

The remainder of the paper is organized as follows: Section II explains the Time Difference Of Arrival (TDOA) based sound localization algorithms. Section III-IV describe the methodology used for event localization and classification. Section V describes the performance evaluation and comparison with baseline models. We conclude this paper with our open discussions in Section VI.

II. A BASIC TDOA ALGORITHM

Our sound event localization method is based on TDOA, able to locate the acoustic source through the time differences among all acoustic sensors using trilateration [18].

In a 2D space topology, the TDOA algorithm is defined as: Let (x, y) be the unknown coordinate of a sound source and let $[(x_m, y_m)]_{m=1}^{M}$ be the known coordinates of acoustic sensors, where M is the number of acoustic sensors. Let τ_m be the relative time of arrival to different sensors and let τ_1 equals to 0 for simplicity. Let r_m be the distance of the sound source to sensor m, and v be a constant speed of sound, we can write the distance as a function of speed and time for $m = 2, 3...M$.:

$$r_m - r_1 = v\tau_m - v\tau_1 = v\tau_m \tag{1}$$

The above Equation 1 can be transformed into:

$$v\tau_m - v\tau_2 + \frac{r_1^2 - r_m^2}{v\tau_m} - \frac{r_2^2 - r_m^2}{v\tau_m} = 0 \tag{2}$$

Replacing r with x, y coordinates, for any $m = 3, 4..., M$.:

$$\begin{bmatrix} A_3 & B_3 \\ A_4 & B_4 \\ ... & ... \\ A_M & B_M \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} C_3 \\ C_4 \\ ... \\ C_M \end{bmatrix} \tag{3}$$

$$A_m = \frac{1}{v\tau_m}(-2x_1 + 2x_m) - \frac{1}{v\tau_2}(2x_2 - 2x_1) \tag{4}$$

$$B_m = \frac{1}{v\tau_m}(-2y_1 + 2y_m) - \frac{1}{v\tau_2}(2y_2 - 2y_1) \tag{5}$$

$$C_m = v\tau_m - v\tau_2 + \frac{1}{v\tau_m}((x_1^2 + y_1^2 - x_m^2 - y_m^2) - \frac{1}{v\tau_2}(x_1^2 + y_1^2 - x_2^2 - y_2^2) \tag{6}$$

Based on Equation 3, the sound source coordinates (x, y) are obtained by calculating the A, B, C vectors and then solving the equation (only yields an answer when $M >= 4$, i.e. requires four sensors at least). The unknown variable in Equation 3 is τ_m, thus the localization problem boils down to resolve τ_m. Figure 1 shows the sound wave of a *door slam* event received by 4 sparsely spaced acoustic sensors, the red dashed lines show the calculated τ for each sensor. Notice how time of arrival is shifted in ms for each sensor making the synchronization a crucial task. One method to synchronize these signals is by using time-smoothing together with cross-correlation [6]. Each two signals are synchronized when their cross-correlation value reaches the maximum.

However cross-correlation only gives the similarity between signals and not hints of signal sub-components, this method does not work when there are overlapping sound events, e.g. someone happens to be talking while the door slams.

III. OVERLAPPING SOUND SOURCE LOCALIZATION

As discussed in Section II, TDOA algorithm locates a sound event well but struggles when estimating mixed sound signals. However, mixed sound signals can be distinguishable when applying a Short Time Fourier Transformation (STFT) as illustrated in Figure 2 where the spectrogram shows two overlapped events received by 4 microphones. One can easily find the start of door event in each graph, a tall (wide frequency) yet narrow (short time) column. The other signal concentrated in lower frequencies corresponding to voice bands is people talking. We aim to look into every small yet significant regions in the spectrogram, from which the time-of-arrival difference (i.e. τ) is easier to be synchronized. Following are the three steps of our sound source localization algorithm:

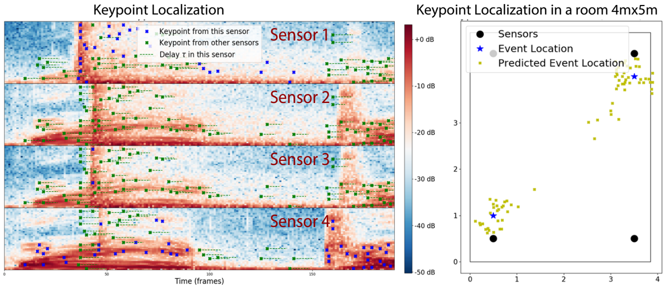

Fig. 3. Keypoints synchronization (left) and localization (right)

A. Keypoint detection

The idea of keypoints is to find important transients in both spectral and temporal domain, and use them together to locate the sound events. There are two reasons behind our approach:

I. Because of noise, sound attenuation and multi-path effect, the same event captured by different sensors differ both in time of arrival and frequency bands. Therefore, it is easier to synchronize τ of small keypoints than the whole spectrogram. Since keypoints are power peaks, they are better noise resistant and more likely to be present in all nearby sensors. After successfully synchronizing the keypoints, we can estimate the number and location of sound sources through aggregation.

II. Keypoints contain the most important information of the signals, thus can be used in sound classification. The keypoints distribution indicates the major frequency bands of an event and their variation over time. The geometric shape surrounding a keypoint describes the Q-factor information around the major frequency bands. By clustering the keypoints, we extract important features to classify the sound sources.

Mathematically, a keypoint is expressed as: $K_i = [s_i, f_i, t_i]$ where $s = 1, 2, ..M$ and M is the sensors number, f and t are the frequency and time. This three-element tuple means one keypoint is detected at sensor s, time t and frequency f. We define G as the spectrograms of all sensors and K_i is a coordinate of G, so that $G(K_i)$ is the amplitude of a keypoint.

Keypoints are detected at locations that are local maximum across both frequency and time, subject to a constant threshold, this can be expressed as:

$$\begin{gathered} G(K_i) = G(s_i, f_i, t_i) = max(G(s, f, t), thr), \\ \forall s, f_i - h < f < f_i + h,\ t_i - w < t < t_i + w \end{gathered} \tag{7}$$

thr is a constant filter for background and microphone noise so that no keypoints would be detected when no events happen. $\boldsymbol{h}$ and $\boldsymbol{w}$ attributes define the local area, height and the width, in a spectrogram. To clarify, in a STFT spectrogram of a 20 kbps sound with window size 1024 and 3/4 overlap, $h = 10$ and $w = 6$ represents a region of 400 HZ and 15 ms. This region must be large enough to capture important shape information and small enough to have sufficient elected keypoints.

Figure 3 left shows keypoint detection results for sound events illustrated in Figure 2. All the keypoints are detected in sensors **1** and **4** while few were detected in sensors **2** and **3**. Because the nearest sensor captures the event first with the highest amplitude and the talking occurs near sensor **1** while the door is close to sensor **4**.

B. Keypoint localization

Sound localization concerns finding the relative time-of-arrival for each sensor. Taking any keypoint as the start, we find when it arrives at the other sensors. Since sound frequency does not change through propagation, this keypoint shows up at the same frequency on all sensors with a short time delay T. We define the square-shaped region centered at keypoint K_i as $Region(K_i)$, from the same region the keypoint is elected:

$$\begin{gathered} Region(K_i) = \{[s_i, f, t]\}, \\ f_i - h < f < f_i + h,\ t_i - w < t < t_i + w \end{gathered} \tag{8}$$

and we define its spectrogram as:

$$GReg(K_i) = \{G(p)\}, \forall p \in Region(K_i) \tag{9}$$

Our synchronization algorithm is to search the τ in each sensor that makes $GReg([s, f_i, t_i + \tau])$ 'matches' $GReg(K_i)$ the best, $\forall s \neq s_i$. A typical similarity function of 2 vectors is cross-correlation, with which we can find τ by:

$$\begin{gathered} \tau_{s,i} =_t (\rho(GReg(K_i), GReg([s, f_i, t_i + t]]))), \\ \forall s \neq s_i, t \in \{0, 1, ...T_{max}\} \end{gathered} \tag{10}$$

$$\begin{gathered} \tau_{s,i} =_t (\rho(GReg_{pr}(K_i, t), GReg([s, f_i, t_i + t]))), \\ \forall s \neq s_i, t \in \{0, 1, ...T_{max}\} \end{gathered} \tag{11}$$

where $\tau_{s,i}$ is the sound arrival time at sensor s for keypoint K_i, T_{max} is the maximum of possible time arrival difference, ρ is the Pearson correlation function [19]. T_{max} is estimated through speed of sound and the distance between sensors.

Cross-correlation method works well with array sensors close to each other, but fall short when sensors are sparsely distributed and factors such as reflection and attenuation can highly influence the signal received by each sensor. By 'matching' the signals at different sensors, we should consider these factors where sound attenuation is the most important. The sound amplitude attenuation in air due to atmospheric absorption can be expressed as: $A_a = ar \quad db$, where r is sound travelling distance in meter, and a is the attenuation coefficient in dB per meter [20]. The coefficient a is dependant on relative humidity, temperature and sound frequency. According to [20] for an optimal indoor environment, the humidity and temperature are fixed at 40% and 20°C, respectively. Therefore sound attenuation in the simplified model has 2 parameters: distance and frequency. Higher the frequency, further the signal propagation, and faster the amplitude decay.

The sound magnitude attenuation in the same frequency band is in proportion to the distance. A keypoint K_i detected by sensor s_i with amplitude $G(K_i)$, would have the amplitude $G_{pr}(K_i, t_i, s_j)$ when it is detected by s_j after time T:

$$G_{pr}(K_i, T, s_j) = G(K_i) \times 10log_{10}a(t_i - T) + e \tag{12}$$

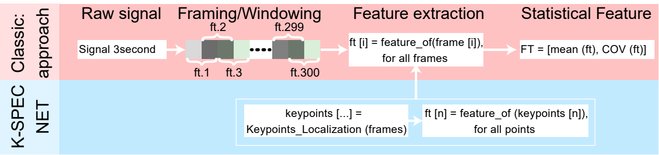
Classic: approach
K-SPEC NET
Raw signal
Framing/Windowing
Feature extraction
Statistical Feature
Signal 3second
ft.2
ft.299
ft.1
ft.3
ft.300
ft [i] = feature_of(frame [i]), for all frames
FT = [mean (ft), COV (ft)]
keypoints [...] = Keypoints_Localization (frames)
ft [n] = feature_of (keypoints [n]), for all points

Fig. 4. Feature extraction flow of statistic-SVM

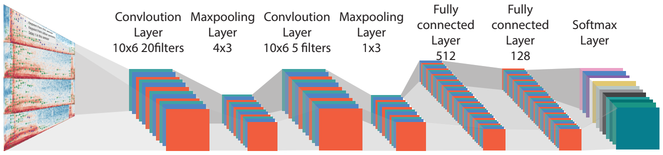
Convloution Layer 10x6 20filters
Maxpooling Layer 4x3
Convloution Layer 10x6 5 filters
Maxpooling Layer 1x3
Fully connected Layer 512
Fully connected Layer 128
Softmax Layer

Fig. 5. CNN model for event classification

where e stands for the unpredictable residuals such noise and reflection. Because the prediction depends on the amplitude and time delay, $G_{pr}(K_i, T, s_j)$ can be written as $G_{pr}(K_i, T)$.

According to Equation (12), we can predict the attenuated spectrogram at the keypoint region $GReg(K_i)$ after time delay t by point-wise operation, denoted as $GReg_{pr}$:

$$GReg_{pr}(K_i, t) = \{G_{pr}(p, t)\}, \forall p \in Region(K_i) \tag{13}$$

Replacing the attenuated spectrogram $GReg_{pr}(K_i, t)$ of $GReg(K_i)$ in (10), we obtain the attenuation-adjusted model as: An example of the keypoint localization result is shown in Figure 3, where the left chart shows keypoints synchronization result, i.e. finding τ, and the right chart shows the keypoint locations calculated with τ. A blue 'x' denotes a detected keypoint. For each blue 'x' of any sensor, we put one green 'x' at the same coordinate of the other 3 sensors as a mark. The green dash line after the green 'x' represents the time arrival delay τ. In the right chart of keypoints localization, the two blue stars indicate the sound events locations, the yellow dots are the predicted locations of all keypoints. In this chart, most of the predicted keypoints are closely distributed around the real locations, only a few are far away. The large distance between locations indicates an error in keypoint synchronization, due to some spectrograms regions where multiple events are heavily overlapped.

C. Keypoints aggregation

Following keypoint identification, we aggregate the keypoints according to their locations to make up the original events, assuming that each event happens at a different location. K-means clustering algorithm partitions the samples into k clusters, with each sample belonging to the cluster with the nearest distance from center. [21]. In our case, the event number is the cluster number k with center the event location.

Our keypoints aggregation algorithm consists of 3 steps:

I. Find the best k with silhouette analysis from the k-means.
II. Filter out outlying keypoints and small clusters.
III. Re-calculating the cluster-centers as the final locations of events with the filtered k and keypoints.

Silhouette analysis in step 1 is a method of interpretation and validation of consistency within clusters of data based on a so called silhouette value [22]. Silhouette value measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation), similarity is based on the distances to cluster centers. The k with the highest silhouette is chosen as the optimal cluster number, meaning the samples are closer to their own center while far away from other cluster centers. However, this is a general method for stochastic samples and has some deficiencies, for example it only works with cluster numbers greater than one, which is not true in our case.

Step 2 improves further keypoint location accuracy because unlike normally distributed random samples, keypoints seem as bi-polar. Most of them are *correct* and close to the event location. Some keypoints that cannot be equally synchronized, we call them *wrong*, are sparsely distributed elsewhere making them harder to synchronize where multiple events overlap in the same major frequency bands and time. In the location graph we use the 'distance-to-center' criteria to filter them out. Let d_{ci} be the distance of point i to its corresponding cluster center c, we use a threshold D_{thr} to prune the outliers:

$$Discard\ K_i,\ if : d_{ci} < D_{thr}, \forall i \tag{14}$$

The threshold D_{thr} is a constant that denotes the maximum localization error to tolerate. This error can be resulted from noise, background sound or even the movement of sound source. Another approach to improve the result is to remove the clusters which contain very few members, as they are likely to be made up from falsely synchronized keypoints. The cluster centers are re-calculated based on the remaining keypoints indicating the final locations of the sound events.

IV. Sound Event Classification

After the event localization, we aggregate the keypoints at each event location and classify them into pre-defined activity classes. We decided to adapt 2 models that performed good in a single audio events classification [23], [24], Statistics-SVM and Spectrogram-CNN. To apply these two models for this scenario, adjustments need to be made since our inputs are the keypoints that are different from the regular audio inputs.

1) Statistics-SVM [23]: Uses the statistics (mean and variance) of short-frames as the input features and SVM as the classifier. Feature arrays are generated by extracting features for each short audio stream frames. The statistics i.e. mean and variance are calculated from this feature array as the final features. Statistics features are mainly used in audio processing due to the audio signals time-varying characteristic, where the short frames represent the details better while the statistics aggregate the details and stands for a global representation [23], this feature extraction is shown in Figure 4.

Similarly we extract features by replacing the short duration frames with keypoints. A featureset, denoted as F_i, portraying the characteristics of the event both in time and frequency domain, is extracted from each keypoint:

$$F_i = \{f_i,\ Rolloff_i,\ STE_i\} \tag{15}$$

i. f_i is the center frequency of the keypoint.

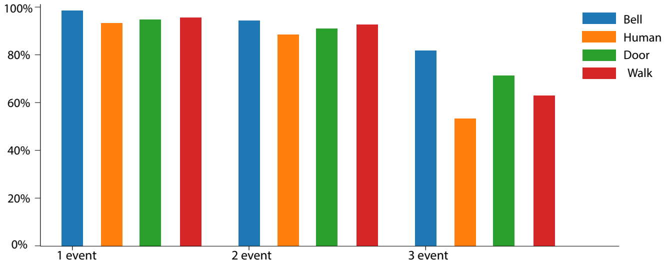

Fig. 6. Sound event localization accuracy for sound class

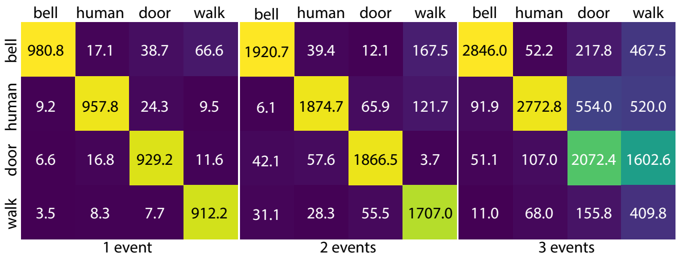

Fig. 7. Confusion matrix of event classification for K-Spec NET

ii. $Rolloff_i$ have two values from time and frequency domain respectively. Spectral-rolloff, used in music information retrieval [25], represents the point where N% power are concentrated below that frequency (normally N equals 90%). We define time-rolloff as the time duration where N% power are concentrated at the center.
iii. STE_i **(Short-time energy)** is a feature used in audio analysis that describes the energy of signal [25].

After F_i of each keypoint is extracted, the mean and variance of all F_i is calculated as the event feature. This event feature is then feed to SVM classifier to predict the event type.

2) Spectrogram-CNN: Convolutional Neural Network (CNN) is a class of deep neural networks [26], commonly used in computer vision domain. It requires minimal preprocessing compared to other image classification algorithms. Converting one-dimensional sound stream to a spectrogram allows the use of CNN for sound processing applications such as speech recognition [27] and environmental sound recognition [24]. Figure 5 shows the achitecture of uur CNN-based classification model, where the input is the incomplete spectrogram aggregated from the keypoints, while non-keypoints part are padded with zeros and the silent frames are removed from the incomplete spectrogram.

V. Experimental Results

This section, presents the experimental results for both localization and classification. The dataset is split 70% training and 30% test with five folds cross-validation for training set.

A. Dataset

Four classes are chosen as common office sound events: speech, footsteps, door slam, bell. 100 unique samples per each class are recorded with a mono-microphone 2m away from the sound source. Furthermore, an indoor sound simulation [28] is used to create synthetic complex sound events from the recordings. Three synthetic levels are created with level number meaning the number of overlapped sounds ($level^n = synthetic(n)$). For each level, we randomly pick single events to create 1000 overlapping events, resulting in different complexity levels in a noisy environment. During the shuffling, each overlapping event is labelled with its class and location.

B. Sound event decomposition and localization

Two metrics are used to evaluate our method: all-correct and partially-correct. Partially-correct means at least one of the events is located correctly. Using partial correctness as a metric gives a better performance overview. An error of 0.5m is tolerated because the application scenario is not too strict, since the major objective is to split overlapping events into their sub-components through the localization algorithm. Moreover the baseline is only provided for signal decomposition (presented in the next subsection V-C together with event classification) while not for localization as it is hard to find one for our scenario. Table I shows the results for three different complexity level. Results show that when only two events are overlapped, the accuracy jumps above 90% and drops significantly in all experiments when three events are overlapped. Worth mentioning is that the attenuation-correlation model works better than the cross-correlation model. We also looked into the differences of performance between each sound class, as shown in Figure 6. These results are all from the attenuation-correlation model with the bell sound as the best located while human voice is the worst, this difference becomes especially obvious in 3-events experiments.

TABLE I
Localization and Classification Results for Agg. Keypoints

Results of sound localization using aggregated keypoints

	1-event		**2-events**		**3-events**	
	cro-corre	att-corre	cro-corre	att-corre	cro-corre	att-corre
All correct	97.10%	98.80%	87.10%	90.30%	61.70%	66.90%
Partly correct	-	-	93.50%	95.80%	74.10%	85.70%

cro-corre = cross-correlation model, att-corre = attenuation-correlation model

Results of sound classification using aggregated keypoints

	1-event			**2-events**			**3-events**		
	SVM	CNN	Baseline	SVM	CNN	Baseline	SVM	CNN	Baseline
Accuracy	92.60%	94.50%	97.80%	74.30%	86.10%	68.60%	45.90%	60.70%	-
F1 score	92.20%	93.90%	97.60%	72.70%	82.50%	65.30%	43.20%	52.50%	-
Complexity	0.41	7.3	1.6	0.57	8.1	2.2	0.62	8.5	-

SVM = statistic-SVM model, CNN = spectrogram-CNN model, Baseline = baseline model Overlap-SVM

C. Sound event classification

We choose the coupled-NMF algorithm in [11] and joint-keypoint in [29] as the baselines, given that both work in similar scenarios as ours while rely on one signal microphone input. The coupled-NMF turns the traditional unsupervised NMF model into a supervised learning problem which can automatically label the class of each sub-component. The joint-keypoint model detects events based on the joint probability of spectrogram-keypoints and event-class. While both the two baseline models decompose and classify the overlapping sound in one complex model, our model is lighter since we only classify the already decomposed signals.

Table I shows the results of SVM, CNN and the baseline models. The metrics consist of Accuracy, F1 score and Complexity. All experiments run on the RaspberryPi-3 on a single core in order to compare the complexity through running time. The complexity is calculated as the running time of classifying an audio sample divided by the audio length, meaning that a realtime model should have the complexity less than 1.0.

Of all models, statistic-SVM runs faster while coupled-NMF is the slowest, both outperform the baselines in classifying overlapped sound events, Spectrogram-CNN is the best performer. The major advantage for our models comes from the previous localization step which splits overlapped signals into single events, thus, simplifying the problem, with classification accuracy is quite close to the localization accuracy. Coupled-NMF was not used for 3-events because this model is designed for 2-events and the complexity increases exponentially with more events overlapping. In terms of complexity, our model is largely simplified as it only considers a single event and the synchronization algorithm contains straight forward calculations at large and does not bring too much overhead.

Confusion matrix in Figure 7 shows that the performance of each class is equally good for single event classification and significantly different for overlapping events. The bell event is the mos accurate, as its major frequency bands are much higher than the rest and can be easily identified. The good performance of classifying human voice mainly comes from the long duration of speech sound, so that many keypoints are able to be correctly located. Footsteps and door sound are most likely to be misclassified, because they have very similar frequency characteristics. However, considering the location information we can differentiate door and footsteps based on the sound location and room topology, since doors are static objects. To better classify the footstep sound we need to add harmonic features of a long duration into the model input.

VI. 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 tackle the sound overlapping issue, we deployed microphones in room corners and use a novel sound localization technique to decompose the overlapping sound events into their sub-components. Apart from the good performance and low complexity, this method is easily scaleable since it is based on the sound propagation model, while single sensor models are based on learning events from vast data of specific classes.

In our experiments, 90% of events are correctly located within a 50 cm error. Needless to say, this localization precision cannot be compared with microphone-arrays techniques that can reach millimeter level precision. However we chose not to further improve the precision since our major concern is the classification of overlapped sound events. In most cases, different sound events usually happen meters away apart.

The located keypoints are fed to a CNN network, making the algorithm more efficient. In the future, we aim to simplify the CNN classifier with less weights to balance the complexity and performance. We also intend to explore k-SpecNET in more complicated scenarios such the continuous monitor of crowd flow.

References

[1] F. Sadri, "Ambient intelligence: A survey," *ACM Comput. Surv.*, vol. 43, no. 4, pp. 36:1–36:66, 2011.

[2] R. Yang *et al.*, "Learning from a learning thermostat: lessons for intelligent systems for the home," in *ACM UbiComp*, 2013.

[3] J.-M. Valin *et al.*, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering," *Robotics and Autonomous Systems*, vol. 55, 2007.

[4] D. B. Ward *et al.*, "Particle filter beamforming for acoustic source localization in a reverberant environment," in *ICASSP*, vol. 2, 2002.

[5] J. Scott *et al.*, "Audio location: Accurate low-cost location sensing," in *PerCom*. Springer, 2005, pp. 1–18.

[6] Y. Guo *et al.*, "Localising speech, footsteps and other sounds using resource-constrained devices," in *IEEE IPSN*, 2011.

[7] A. Maddumabandara *et al.*, "Experimental evaluation of indoor localization using wireless sensor networks," *IEEE Sensors Journal*, 2015.

[8] Guodong Guo *et al.*, "Content-based audio classification and retrieval by support vector machines," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 209–215, 2003.

[9] T. Heittola *et al.*, "Sound event detection in multisource environments using source separation," in *CHiME*, 2011.

[10] A. Dessein *et al.*, "Real-time detection of overlapping sound events with non-negative matrix factorization," in *Matrix Information Geometry*. Springer, 2013, pp. 341–371.

[11] A. Mesaros *et al.*, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *ICASSP*, 2015, pp. 151–155.

[12] A. Temko *et al.*, "Acoustic event detection in meeting-room environments," *Pattern Recognition Letters*, vol. 30, 2009.

[13] H. D. Tran *et al.*, "Jump function kolmogorov for overlapping audio event classification," in *ICASSP*, 2011, pp. 3696–3699.

[14] G. Parascandolo *et al.*, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *ICASSP*, pp. 6440–6444, 2016.

[15] J. Dennis *et al.*, "Image feature representation of the subband power distribution for robust sound event classification," *IEEE TASLP*, 2013.

[16] I. McLoughlin *et al.*, "Robust sound event classification using deep neural networks," *IEEE TASLP*, vol. 23, no. 3, 2015.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, 2004.

[18] T. He *et al.*, "Range-free localization schemes for large scale sensor networks," in *MobiCom 2003*. ACM, 2003, pp. 81–95.

[19] I. Lawrence *et al.*, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, pp. 255–268, 1989.

[20] P. M. Morse and ASAAIP, *Vibration and sound*. McGraw-Hill, 1948.

[21] J. A. Hartigan *et al.*, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society*, 1979.

[22] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[23] M. Cowling *et al.*, "Comparison of techniques for environmental sound recognition," *Pattern Recognition Letters*, vol. 24, no. 15, 2003.

[24] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE MLSP*, 2015, pp. 1–6.

[25] D. Mitrović *et al.*, "Features for content-based audio retrieval," in *Advances in Computers*. Elsevier, 2010, vol. 78, pp. 71–150.

[26] K. Simonyan *et al.*, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[27] W. Xiong *et al.*, "The microsoft 2017 conversational speech recognition system," in *ICASSP*, 2018, pp. 5934–5938.

[28] R. Scheibler *et al.*, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *ICASSP*, 2018.

[29] J. Dennis *et al.*, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.