A sound-based crowd activity recognition with neural network based regression models

Wei Wang
w.wang-1@utwente.nl
University of Twente
Enschede, The Netherlands

Fatjon Seraj
f.seraj@utwente.nl
University of Twente
Enschede, The Netherlands

Paul J.M. Havinga
p.j.m.havinga@utwente.nl
University of Twente
Enschede, The Netherlands

ABSTRACT

Activities performed by humans can be recognized by the sound they emit while being performed, hence, researchers have proposed methods that use sound to recognize human activities, by detecting the presence of sound events in short time frames. However, in crowded environments, many sound events overlap making it impossible to distinguish the individual events and methods of detection to fail.

To address this issue and make the sound-based model suitable for crowd activities, this paper proposes to predict the proportion of activities happening in a specific place, by designing two neural network-based regression models: a CNN-model and a concatenate model. The CNN-model takes the Mel-bands as the input and is very popular in single activity recognition problems. Based on the CNN-model, we also designed a concatenate model which additionally inputting the global FFT feature to further improve the performance.

The evaluation of this approach is performed over 3 generated groups of audio samples, where each group has a different crowded-level. Both RMSE and coefficient of determination (R^2 score), are used as evaluation metrics. The experiments show that the concatenate model works statistically better throughout the dataset, with a R^2 score of 0.7377. Results show that using the concatenate model with both short-frame and holistic features provides a better result than any single-feature based model.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → **Sound and music computing**; • **Human-centered computing** → *Ubiquitous and mobile computing theory, concepts and paradigms.*

KEYWORDS

Ambient Intelligence, Crowd Activity Monitoring, Automatic Sound Event Recognition, Machine Learning, Convolutional Neural Network, Concatenate Neural Network, Mel-bands Spectrogram, R-squared score

PETRA '20, June 30-July 3, 2020, Corfu, Greece

ACM ISBN 978-1-4503-7773-7/20/06…$15.00
https://doi.org/10.1145/3389189.3389196

ACM Reference Format:
Wei Wang, Fatjon Seraj, and Paul J.M. Havinga. 2020. A sound-based crowd activity recognition with neural network based regression models. In *The 13th PErvasive Technologies Related to Assistive Environments Conference (PETRA '20), June 30-July 3, 2020, Corfu, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3389189.3389196

1 INTRODUCTION

1.1 Background

Human presence detection and activity recognition constitute a research topic referred to identify the location, movement and actions of an individual based on information collected from on-body or the surrounding sensors. Indoor human activity information is of particular interest in many real-life human-centric applications, ranging from health-care, energy-saving to hazardous and anomaly detection in smart buildings [22].

Apart from the individual-level activity recognition which focuses on a specific person, crowd activity recognition has also drawn a lot of attention recently especially due to the needs of privacy [14]. In a crowd activity recognition task, the interactions and the holistic status of a group of people are of more interest than the individuals' activities. In particular, the recognition and analysis of crowd behaviours can be applied in substantial applications [29], such as crowd management, surveillance system, public space design, etc. For example, detection of the anomalous events in the crowd can help surveillance systems to react and alarm faster. Modelling of the motion and actions of a crowd can help understand the interests of customers and provide valuable guidelines to the space design in retails.

While numerous sensors can be used for activity recognition, the audio sensors are one of the most pervasive and useful ones, given the fact that human activities generate distinctive sound events. Hence, by detecting and recognizing sound events such as talking, walking, hands clapping, one can identify the context of human activities in the vicinity [8]. Other methods like computer vision-based recognition suffer from the line-of-sight problem and raise more privacy concern than audio sensors[5].

On the other hand, using sound also has disadvantages. One of the greatest challenges in sound processing are the overlapping sound events, e.g. when multiple people are activities happen simultaneously in a confined space or in the scenarios of broadcast audio sources from a radio or a TV set. Because of the complexity associated with decomposing the sound information, the performance of sound-event recognition models drops dramatically in the presence of multiple simultaneous sound sources [19][7]. Therefore, unlike computer-vision which is widely researched in crowd behavioural analysis, most sound-based techniques can only identify human

activities in a less densely environment where only a few sound events overlap[26][24]. More specifically, very few researches are conducted using sounds statistical or holistic information to recognize crowd activities.

A common definition of 'crowd' used in many works refers to the scenario when the population density is sufficiently large to disable individual tracking and activity recognition. In this case, the techniques of crowd-activity recognition are not the simple extension of the ones used in individual-activity recognition [13]. In many computer-vision based researches on crowd behavioural analysis, the goal is to extract some kind of information from crowded video footage, which include the directions, velocities and unusual events such as falling, fighting [3][25][16], etc.

1.2 Related works

As mentioned above, while sound can also be used for human activity recognition, it is rarely used in very crowded environments. Sound-based human activity recognition falls under the category of *'Environmental Sound Recognition'* (ESR). ESR models aim to automatically detect and identify audio events from the captured audio signal, which mainly work in a quiet environment with non-overlapping events [10][28].

When then environment becomes more crowded and therefore more *noisy*, several models can be used to tackle the issue of overlapping events, from different perspectives:

I. The first method decomposes the mixed signals into the sub-components based on matrix factorization by reducing a matrix into its constituent parts [11, 17]. Non-negative matrix factorization (NMF) is one of the most popular signal decomposition techniques and has been proved to be very good in decomposing music signals. However, NMF is an unsupervised model which only decompose signals according to the similarity of features in time and frequency domain. In a word, NMF only separates a signal into several components but does not know which component belongs to which sound event.

II. The second method does not decompose the entire signal but only focus on the local features in the image-like spectrogram from Short Time Fourier Transformation (STFT) [7, 15]. These local features are normally called 'keypoints', which are the small areas in a spectrogram image with high power density. A keypoint is like a glimpse of the local spectral information, it needs to be big enough to characterize the features of a sound event. In the meantime, a keypoint also needs to be small enough so that it can be easily separated from a mixed signal. Comparative studies in [15] showed that spectrogram features and deep learning algorithms could work well where only two events overlap.

III. The third method extends the neural-network based non-overlapping models for overlapping events through a multi-labelling output layer with the sigmoid activation function. [20] first use convolutional neural network with a softmax output layer to classify non-overlapping sound events. In order to classify overlapping events, [4] replaced the output layer activation function with a sigmoid function which bounds every output-node value to (0,1). With a post-processing step which binarize the outputs by thresholding, this model can identify multiple types of sound events happening at each time frame.

Although the aforementioned models can tackle the overlapping events problem to a certain extent, they are designed for recognizing individual activities but not the crowd as a whole. These models have not been proved applicable in the crowded environment since the audio dataset used by these researches have at most two or three sound events overlapping [23]. Moreover, these models are all classification models which only tell whether something is happening but without the magnitude information. In a very crowded environment, these yes or no results can barely improve our awareness of the environment as some events may occur all the time. In this scenario, we would rather know more about how many are walking than if someone is walking.

To fill in this gap, this paper proposes to estimate the proportion of different activities from overlapping sound events. The rationale consists of telling what activities happened at a specific time, by building a regression model that outputs the ratio of each perceived activity in a relatively long duration. The model is more feasible than precisely detecting the activities of the individuals, still providing insights of what is most probably happening in that particular place and time.

The remainder of this paper is organized as follows: Section 2 explains the methodology used for our crowd activity recognition and the intuition behind our model. Section 3 describes our dataset and the performance evaluation of our models and the baseline. We conclude this paper with our open discussions in Section 4.

2 METHODOLOGY

As aforementioned, our regression model should output the proportion of different sound events from the input audio stream of a short time. This regression model can infer what are the major activities of the crowd and whether there are anomalies. Our dataset consists of 6 different sound events: walking, clapping, talking, door slam, phones, trolley wheels. All of these sound events are very common indoor sound and can be used to reveal the human activities.

Mathematically, the output event-proportion is defined as the summed duration of this event over the summed duration of all events. Figure 1 shows an example of how we calculate this proportion over two different sound events. In this figure, there is only one person talking at $t1$ so that the output is : $\{P_{talking} = 1, P_{walking} = 0\}$. At $t3$, two are talking and one is walking so that : $\{P_{talking} = 0.66, P_{walking} = 0.33\}$.

In this work, two different neural network models are proposed and compared in order to find the promising method. The first model is a CNN-network with frame-based input features, which can be deemed as the state-of-the-art method for single-event sound classification. In general, a CNN-based model is good at detecting the local characters of a sound spectrogram but falls short for the global or statistic characters.

However, a CNN-based model can learn and find repetitions of some short and sharp sound fragments all concentrated at 4KHZ 5KHZ as the 'footsteps', but can hardly learn the knowledge of 'more low-frequency sound (300HZ 3KHZ) means more people are speaking'. In order to learn statistic characters better, we have designed the second model which additionally takes the global Fast Fourier transform (FFT) bands input based on our first model. The details of these two models are described below.

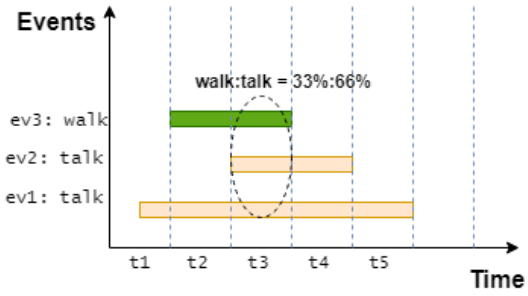

Figure 1. Scenario Description: Our model outputs the proportion of sound events

2.1 CNN-based model and Mel-spectrogram features

This model is a CNN-based model which takes the frame-based Mel-spectrogram features as the input. This model and feature combination has been used in many sound-processing works, especially in the non-crowded environment.

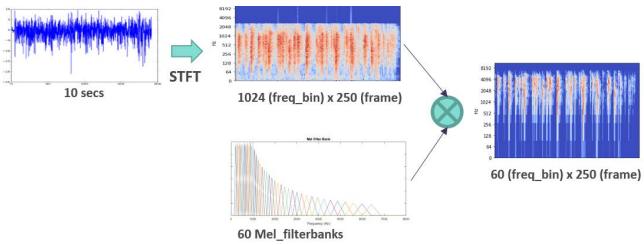

Figure 2. The Mel-spectrogram feature (with 60 bands)

The *Mel-spectrogram* feature is the product of the Mel-filterbanks and the Short-time Fourier transform (STFT) spectrogram. The STFT algorithm first divides the audio stream into frames of equal length and then computes the Fourier transform separately on each short frame. The Mel-spectrogram is the non-linear transform on the STFT so that each Mel-band sounds equal in distances to human listeners. Both STFT and Mel-spectrogram are commonly used features in audio processing where Mel-spectrogram performs generally better in many machine learning models. The Mel-spectrogram of an audio stream is a two-dimensional image-like feature in frame(or time) and frequency axes, as is shown in figure 2. If we let the frame length to be 100*ms* and half-overlapped, and let the Mel-bands number be 60, the Mel-spectrogram feature of a 10 seconds audio stream would be of size 60×200.

After the feature extraction, a CNN-based regression model is used to learn and predict the proportion of each sound event from this image-like feature set. This model is depicted in figure 3 of which the input features from the top are processed to get the regression results by the neural network. By convention, *CNN_n* stands for the *n*th CNN-layer and *Dense_n* stands for the *n*th full-connected layer. More details about the neural network layers and activation functions are specifically explained as follows:

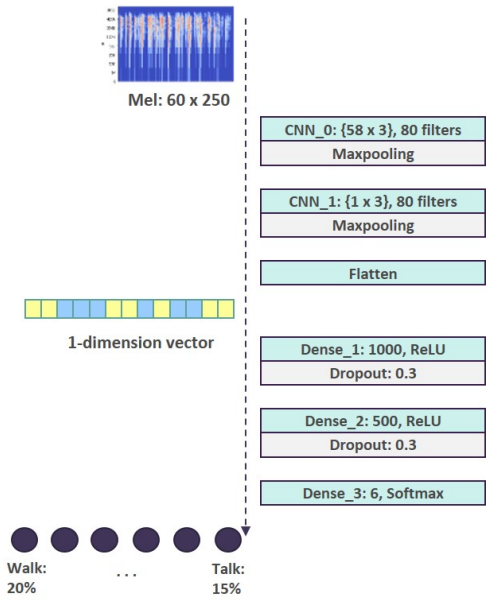

Figure 3. The flow of the CNN-based regression model

(1) *Conventional neural network (CNN)*: Our model begins with two stacked CNN-layers. A CNN-layer consists of many 2-dimensional filters of the same size(i.e. 5×5), which is also called the *filter-size* of a layer. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the filter and the input to produce a 2-dimensional output of that filter. As a result, the network learns filters that activate when it detects some specific shapes (or feature) at any position in the input. In the output of a CNN-layer, each neuron only receives a subarea of the input (of the same size as the filter-size), which is also called the *receptive field* of the neuron. Therefore, the receptive area of a CNN-layer is much smaller than the entire input layer, which greatly reduces the calculation density than a full-connected layer.
At the beginning, we stacked two CNN-layers together so that the second layer can detect a larger shape or feature from the spectrogram. The layer's parameters mainly consist of: filter-number, filter-size and the strides of the convolution. For the first CNN-layer, a good setting is 80 filters and a filter-size of 58×5 with stride 1×1. The height of the filter-size (58) should be only slightly smaller than the entire frequency-bands (60) so that the learned feature is not much frequency-invariant. The intuition of using the 'elongated' filter in the first CNN-layer is shown in Figure 4. As a comparison, using the small CNN filters (e.g. 5×5) can only detect the common shapes at different frequency-bands, which however sounds very different and makes little sense in classifying sound. The second CNN-layer could have 80 filters and the filter-size of 1×3 so as to detect a larger size of feature in time domain than the first CNN layer.

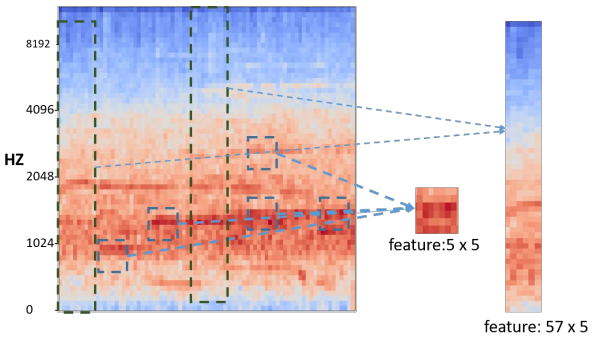

Figure 4. The initial CNN-layer filters are specifically chosen with an elongated rectangular shape in order to detect frequency non-invariant features

(2) *Maxpooling*: In neural network, a CNN-layer very commonly accompanies with a maxpooling-layer. A maxpooling layer applies a max filter to (usually) non-overlapping subareas of the input neurons. This layer can downscale the learned features from the CNN-layer and help with the overfitting by suppressing the influence of the smaller outputs.
In our model, the maxpooling-layer size in time-axis is exactly the same as that of its previous CNN-layer so that no important features are lost.

(3) *Flatten*: A flatten layer does not change the value but only reshapes the multi-dimensional matrix into a vector. This makes the data able to feed into a fully connected layer which is better suitable for the classification or regression task.

(4) *Rectified Linear Units (ReLU)*: We use two stacked full-connected layers with *ReLU* activation functions for the regression of the flattened data. ReLU is a non-linear function which is widely used as the activation function in neural network [6]. The basic ReLU function has the form of:

$$ReLU(x) = \begin{cases} x, & \text{if } x > 0, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Compared to the traditional sigmoid activation function, ReLU has several advantages such as faster computation, more efficient gradient propagation and sparse activation, etc. Many variants of the ReLU function have also been invented recently such as the Leaky ReLU, Noisy ReLU, or Parametric ReLU to solve potential problems [9, 21]. However as many papers have proposed, the actual differences in performance brought by these variants are very small and they do not grant better performance than the basic ReLU function [1]. For each of these two layers, the neuron number is always selected from between its input size and its output size.

(5) *Dropout*: It is by natural the neural network would likely to overfit and learn the statistical noise from the training data while performs poor with the new coming data. Dropout is an simple yet efficient technique for reducing overfitting in neural networks. In the training process, neurons of the dropout layer would be randomly set to zero at a given ratio (i.e. 30%) so that it looks like some neurons are dropped. It has the effect of making the training process noisy and encourages neuron values learned within a layer to be more sparse. We attached one dropout layer to each ReLU layer and tested the drop ratio from 20% to 50% in the experiments.

(6) *Softmax*: The output layer is a full-connected layer with the Softmax activation function, which is often used by classification models. Softmax function can normalize the input into a probability distribution like output where all values are positive and sums up to 1:

$$S(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}} \tag{2}$$

With its mathematical properties, Softmax is also suitable for calculating the proportions.

2.2 The Concatenate Model of two subroutines

As many works have stated, multiple stacked CNN-layers are good at classifying images through gradually integrate small shapes into higher-level structures, where each CNN layer inputs the output of the previous layer. This character makes CNN also popular in the sound classification task since the spectrogram feature is an image-like two-dimensional feature. However, our problem is to calculate the proportion of each event, which is essentially not an object-classification problem that values local shapes and features identification. Therefore, using only the CNN-based framework with the frame-based features may value too much the details while neglect the statistic features, thus is not fully capable of predicting the event-proportions. In order to verify this hypothesis, our second model is a concatenate model using both frame-based STFT and global FFT features, aiming to solve the problem by looking into both the local and holistic aspects.

More specifically, based on the previous CNN-based model, we addtionally add a subroutine that consists of multiple full-connected layers and inputs the FFT bands of the entire signal. In the FFT bands, only the magnitude part is used so that the feature-length is half the length of the audio stream.

The model architecture is shown in Figure 5, where the Mel-bands and FFT features are firstly extracted from the same audio stream. Next, each input feature is processed by a separate subroutine in the lower-level layers until the two subroutines concatenate (or merge) at the last hidden layer. At the output layer, the regression result is obtained based on the high-level features calculated from both subroutines.

The architecture of the STFT subroutine before concatenation is almost the same as the previously described CNN-based model. For the FFT subroutine, we first connect the input with the maxpooling layer to reduce the dimension of data for less overfitting. Next we use two ReLU layers to extract the high-level representations for the final regression step.

There are two types of layers specifically needed for the network-concatenation:

(1) *Normalization*: Normalization is a family of methods that normalize the output value of the previous layer, i.e. applies a transformation that maintains the mean close to 0 and the standard deviation close to 1. This technique is normally used to for the purpose of improving the speed, performance,

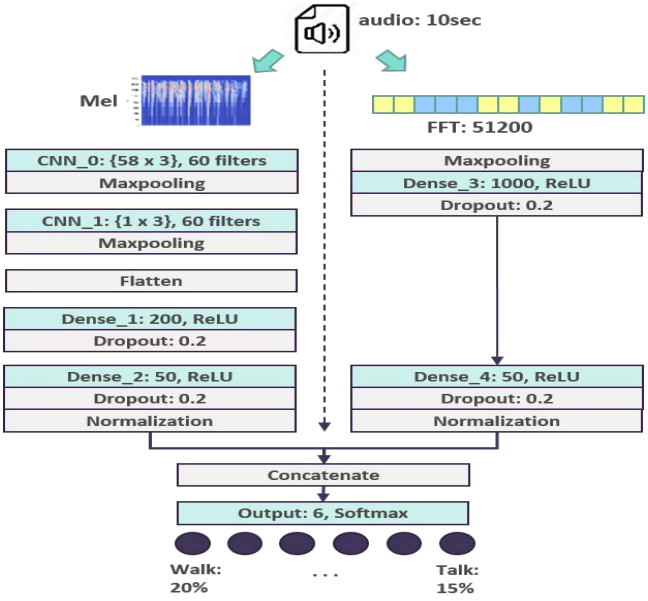

Figure 5. The flow of the concatenate model of two subroutines

and stability of artificial neural networks. In the implementation, we use the BatchNormalization layer proposed by [12]. Before the concatenation of the two subroutines, one *Batch-Normalization* layer is attached to each side. These normalization layers force the neuron-outputs of both subroutines subject to the same distribution thus are more balanced in their contribution to the subsequent concatenation layer.

(2) *Concatenate*: A concatenate layer can take inputs from multiple neural-network layers and merged them into a new vector without changing any neuron values. As an example, if each subroutine has 50 neurons, the concatenate layer would simply connect them and output 100 neurons to the following layer. In neural network, this method makes the framework more flexible and able to process more comprehensive information with a proper design. In one of the recent works, Ahmed et al. proposed a house price prediction model with both the photos and text-description inputs [2], which concatenate two subroutines from the bottleneck layer. Their model achieved much better and stabler results than the models with either image or text input.

The final softmax layer than receives the concatenated neurons and calculate the final regression results for each event type.

3 EXPERIMENTAL EVALUATION

In this section, we present the dataset and the experimental results of our two models from different aspects. In each experiment, we split the dataset into 80% training-set and 20% test-set and all results presented refer to the output of the test-set.

3.1 Dataset

Although there exists some environmental sound dataset for researches, for instance, the urban sound dataset from NYU [23] and TUT [18], none of them are proposed for the crowded environment. In a real crowded environment, it is very hard to label the ground-truth of all things happening around in the real time. Therefore, we simulated the crowded environment and built the dataset by ourselves.

Our dataset consists of 6 different sound events: walking, clapping, talking, door slam, phones, trolley wheels. These sound events are selected since all of them are very commonly heard indoor and are related to different human activities. To build the appropriate dataset, we first collected the clear sound clips from the famous online sound dataset: *www.freesound.org*. For each event we collected 100 sound clips and each is between 1 5 seconds long. Meanwhile, we also converted all the sound files into a unified form, i.e. 10KHZ sampling rate, 2 bytes per-sample and normalized amplitude.

With these single-event clips, we next created the crowded sound samples by the simulation tool 'pyroomacoustics', which was proposed in [27]. Pyroomacoustics is simulation tool for indoor acoustic environment, which can simulate the sounds people hear in complex environments. Users can customize the room size, place sound events in the room, and then obtain the mixed sound stream heard at any location. An example of our data simulation is shown in figure 6 where multiple walking, talking and bell sounds are heard by a microphone in the room centre.

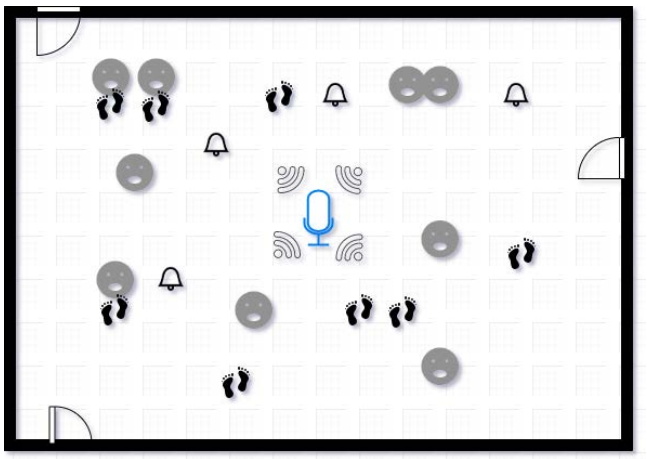

Figure 6. An example of the data simulation by tool 'pyroomacoustics'

Together we generated three groups of audio samples with different crowded-densities, noted as crowd-4, crowd-6, crowd-8. This group name refers to the average overlapping rate. Take group crowd-4 as an example, multiple short sound clips that add up to 40 seconds are mixed into a 10-seconds sample, so that an average of 4 events overlap in each transient.

We generated 20000 samples for each group and every sample is 10 seconds long. For the balance of data, the total length of each single event sound subjects to the same uniform distribution.

Table 1: One of the best hyperparameters of each model found in experiments

CNN-model		Concatenate-model	
CNN_0 filter number	80	CNN_0 filter number	60
CNN_0 filter size	57 x 5	CNN_0 filter size	57 x 4
CNN_1 filter number	80	CNN_1 filter number	60
CNN_1 filter size	1 x 3	CNN_1 filter size	1 x 3
Dense_1 neuron number	1000	Dense_1 neuron number	200
Dense_1 dropout ratio	0.3	Dense_1 dropout ratio	0.2
Dense_2 neuron number	500	Dense_2 neuron number	50
Dense_2 dropout ratio	0.3	Dense_2 dropout ratio	0.2
		Dense_3 neuron number	1000
		Dense_3 dropout ratio	0.2
		Dense_4 neuron number	50
		Dense_4 dropout ratio	0.2

3.2 The hyperparameters

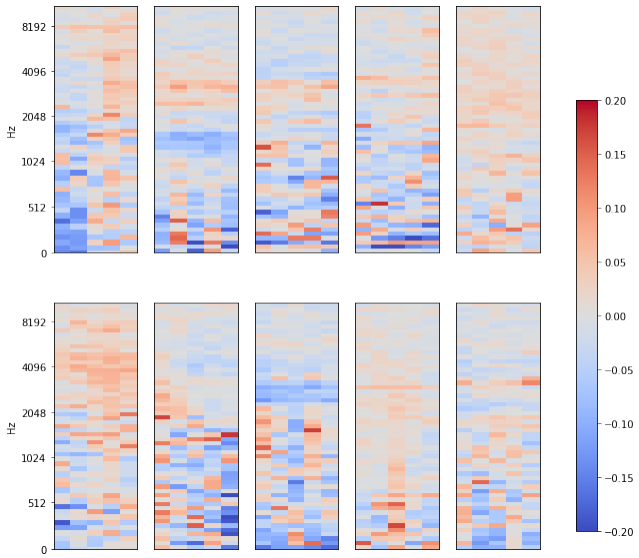

Figure 7. Some of the learned filters from the first CNN-layer

In addition to the model framework, the model hyperparameters also have a significant impact on the performance. In order to find suitable hyperparameters, we first conducted comparative experiments using different sets of values, of which the filter-numbers and filter-size are the most important ones. According to our experiments, one of the best hyperparameters of the two models is shown in table 1. In this table, all the notations such as CNN_1 and Dense_1 are consistent with the notations portrait in Figure 3 and 5. As we can see, compared to the CNN-model, the concatenate-model needs much less parameters in each CNN-layer and also less dropout ratio. This can be explained by the benefit of joint decision making, as each subroutine only needs to extract partial but not all information. Another important configuration is that Dense_2 and Dense_4 should have roughly equal neurons so that the concatenation does not bias towards either subroutine. For each experiment, we set the maximum training epochs to 2000 with the learning rate of 0.001.

For the first CNN-layer (CNN_1), there are 80 filters learned, which are of the size 57×6. Part of these learned CNN-features from one of our experiments are shown in the Figure 7, where the blue points are the features to be suppressed and the red ones are to be activated. As we can see, the learned features consist of many small sparse and non-consistent shapes, which makes sense since each input sample also looks disorganized as it consists of many different events. As a result, using CNN only would not give us as accurate results as in the single-sound classification task.

3.3 The results

Table 2 shows the comparison results for each model in different data groups. For comparison, we also built a control model using the global FFT-features and full-connected layers, i.e. the FFT sub-routine of the concatenate-model. In the experiments, this controlled method performs significantly worse than our proposed models.

We use the root mean square error (RMSE) and R2 score as the metrics of evaluating the regression results. The RMSE is the square root of the mean square error of all events. If the real proportion is t_i and the prediction is p_i for each event i ($i = 1, 2, .., N$, N equals to 6 in our case), we calculate the RMSE as:

$$RMSE = \sqrt{(\sum_{i=1}^{N} p_i - t_i)^2 / N}$$

R2 or R-squared score is known as explained-variation versus total-variation and is commonly used for measuring the differences between the real values and the predictions of a set of digits. R2 score can be explained as:

$$R2(t, p) = 1 - \frac{\sum_i (t_i - p_i)^2}{\sum_i (t_i - \bar{t})^2} \quad (3)$$

, where r and p refer to the real values and the predictions respectively. i is the index and $\bar{r}$ is the mean value of all r. The best possible R2 score is 1.0 when all the predictions and real values are equal and R2 can also be negative (because the model can be arbitrarily worse). A constant model that always predicts with the mean value of r, disregarding the input features, would get a R2 score of 0. In evaluating the regression results, R2 score provides a more intuitional sense than RMSE since its value range is independent of the dataset itself.

In all the three groups, the concatenate-model has shown significantly better performance than the CNN-model. Compared to MSE, R2 score can present the regression results more intuitively as its output range is regardless of the input range.

In order to reveal the impact of input audio length, we also tested the three models on shorter audio clips. In these experiments, we use the same audio content but cut each of the 10-seconds clips into 5-seconds and 3-seconds segments and tested with the same models and hyperparameters. Figure 8 shows that 10-seconds audio clips perform slightly better and more stable than 5-seconds and 3-seconds clips, proved that longer inputs can basically predict better.

Table 2: Overall results of crowd-activity prediction with different data-groups

(data-groups)	crowd-4		crowd-6		crowd-8	
	RMSE*	R2*	RMSE	R2	RMSE	R2
Concatenate-model	0.0306	0.7377	0.0378	0.6237	0.0451	0.4594
CNN-model	0.0334	0.7011	0.0418	0.4996	0.0512	0.303
Control-model*	0.0469	0.4191	0.0485	0.4095	0.0490	0.4029

*RMSE and R2 refer to the average value of all event class
*The control-model refers to the simple full-connected model with global-FFT feature inputs

Furthermore, the performance of all three models can drop very dramatically when the audio-clips become too short (i.e. 3 seconds).

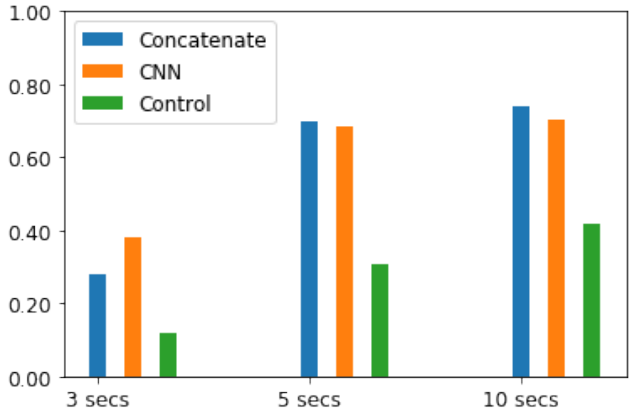

Figure 8. R2 score of each model using inputs with different audio-length

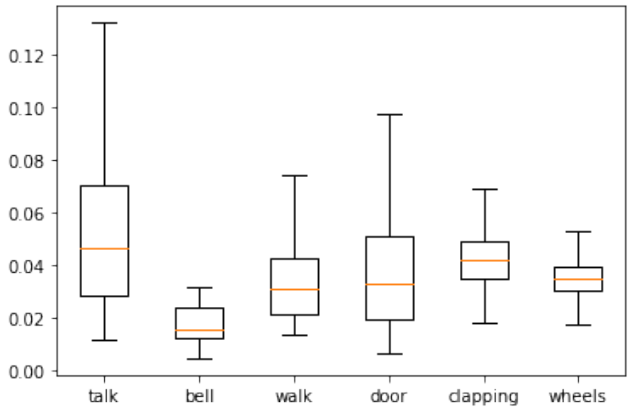

Figure 9. Per-event regression errors of crowd-4 with the concatenate-model

The per-event results of the concatenate-model is shown in Figure 9 using boxplot. In this figure, the per-event RMSE is calculated using the crowd-4 data. In our experimental results, the speech or speech sound has the largest error, while the bell or door sound is most accurately predicted.

Inaccurate predictions of speech sound in a crowd are reasonable because human speech is mainly concentrated in 300HZ-3KHZ and overlaps with the spectrograms of all other sound events. In addition, the noise in the environment is mainly concentrated in the low frequency band, which further makes it difficult to distinguish the speech.

The inaccurate prediction of speech sound in the crowd is reasonable because the human speech sound mainly concentrates in the low frequency bands (300HZ - 3KHZ), which can easily overlap with other sound events. Further more, the noise in the environment also mainly concentrate in the low frequency bands which makes the speech sound more difficult to be identified.

4 OPEN DISCUSSION

In this work, we have proposed a system to estimate the proportion of different sound events in the crowded environment. This work can be used to monitor the crowd activities in a new perspective, as most up-to-date proposed crowd-activity recognition systems are based on images. Our methodology is largely based on a CNN-based model with the spectrogram feature, which has been proved outstanding in classifying the single-event sound. However, although our problem is in many aspects similar to single-event sound classification, there is also a significant difference. Firstly, we used a regression model which outputs the proportion of each type of event instead of a classification model. This is because the sound of crowd which consists of many different overlapping events is difficult to be separated and classified. Secondly, we proposed a concatenate model using both the framely spectrogram feature and the global FFT features as the input. We believe the global FFT features could help improving the performance since it reflects the holistic characters and is a complement to the short framely features. This has also been proved by our results where the concatenate-model performs better than the CNN-model statistically. Furthermore, the concatenate-model actually needs less weights than the CNN-model and also allows less complexity and help with the overfitting.

Our evaluation dataset is created from the tool 'pyroomacoustics', which can simulate indoor sound in the real acoustic environment. In our experiments, the per-event results show that the bell sound and the talking sound are the best and worst of all to predict. This result matches the common sense since the power of bell sound all concentrate in a short frequency bands and time. Regarding the input length, our experiments also shows that the input sound should not be less than 3 seconds, otherwise the accuracy would drop dramatically. This may partly be caused by the truncation of the sound events since many audio events longer than 3 seconds, so that the model could fail to catch the entire signal. In the dataset

simulation, we assume every sound event has the similar intensity so that each single-event sound used for generating the crowd sound is normalized in magnitude. This could also bring one issue to our research, i.e. the volume of each sound event in the real environment may differ a lot, e.g. some are talking very loudly while some are whispering. In this sense, our model would reflect the proportion of the 'event power' more than the 'event number' in the real life applications.

REFERENCES

[1] Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. 2014. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830* (2014).

[2] Eman Ahmed and Mohamed Moustafa. 2016. House price estimation from visual and textual features. *arXiv preprint arXiv:1609.08399* (2016).

[3] Saad Ali and Mubarak Shah. 2007. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–6.

[4] Emre Cakir, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. 2015. Polyphonic sound event detection using multi label deep neural networks. In *2015 international joint conference on neural networks (IJCNN)*. IEEE, 1–7.

[5] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. 2012. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 790–808.

[6] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015).

[7] Jonathan Dennis et al. 2013. Image feature representation of the subband power distribution for robust sound event classification. *IEEE TASLP* 21, 2 (2013).

[8] Antti J Eronen, Vesa T Peltonen, Juha T Tuomi, Anssi P Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho, and Jyri Huopaniemi. 2006. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 1 (2006), 321–329.

[9] Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy activation functions. In *International conference on machine learning*. 3059–3068.

[10] Guodong Guo et al. 2003. Content-based audio classification and retrieval by support vector machines. *IEEE Transactions on Neural Networks* 14, 1 (2003), 209–215. https://doi.org/10.1109/TNN.2002.806626

[11] Toni Heittola et al. 2011. Sound event detection in multisource environments using source separation. In *CHiME*.

[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[13] Anders Johansson, Dirk Helbing, Habib Z Al-Abideen, and Salim Al-Bosta. 2008. From crowd dynamics to crowd safety: a video-based analysis. *Advances in Complex Systems* 11, 04 (2008), 497–527.

[14] Shian-Ru Ke, Hoang Le Uyen Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. 2013. A review on video-based human activity recognition. *Computers* 2, 2 (2013), 88–131.

[15] Ian McLoughlin et al. 2015. Robust sound event classification using deep neural networks. *IEEE TASLP* 23, 3 (2015).

[16] Ramin Mehran, Alexis Oyama, and Mubarak Shah. 2009. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 935–942.

[17] Annamaria Mesaros et al. 2015. Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations. In *ICASSP*. 151–155.

[18] A. Mesaros, T. Heittola, and T. Virtanen. 2016. TUT database for acoustic scene classification and sound event detection. In *2016 24th European Signal Processing Conference (EUSIPCO)*. 1128–1132. https://doi.org/10.1109/EUSIPCO.2016.7760424

[19] Giambattista Parascandolo et al. 2016. Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings. *ICASSP* (2016), 6440–6444. https://doi.org/10.1109/ICASSP.2016.7472917

[20] Karol J Piczak. 2015. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[21] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).

[22] Fariba Sadri. [n. d.]. Ambient Intelligence: A Survey. 43, 4 ([n. d.]). https://doi.org/10.1145/1978802.1978815 3.

[23] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 1041–1044.

[24] Venkatesh Saligrama and Zhu Chen. 2012. Video anomaly detection based on local statistical aggregates. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2112–2119.

[25] Shobhit Saxena, François Brémond, Monnique Thonnat, and Ruihua Ma. 2008. Crowd behavior recognition for video surveillance. In *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 970–981.

[26] Shobhit Saxena, François Brémond, Monnique Thonnat, and Ruihua Ma. [n. d.]. Crowd Behavior Recognition for Video Surveillance. In *Advanced Concepts for Intelligent Vision Systems* (2008) *(Lecture Notes in Computer Science)*, Jacques Blanc-Talon, Salah Bourennane, Wilfried Philips, Dan Popescu, and Paul Scheunders (Eds.). Springer, 970–981. https://doi.org/10.1007/978-3-540-88458-3_88

[27] Robin Scheibler et al. 2018. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *ICASSP*. 351–355.

[28] Andrey Temko et al. 2009. Acoustic event detection in meeting-room environments. *Pattern Recognition Letters* 30, 14 (2009), 1281–1288. https://doi.org/10.1016/j.patrec.2009.06.009

[29] Beibei Zhan, Dorothy N Monekosso, Paolo Remagnino, Sergio A Velastin, and Li-Qun Xu. 2008. Crowd analysis: a survey. *Machine Vision and Applications* 19, 5-6 (2008), 345–357.