Audio Engineering Society

Convention Paper

Presented at the 152nd Convention
2022 May, Online

Classifying Sounds in Polyphonic Urban Sound Scenes

Jakob Abeßer[1]

[1] *Fraunhofer IDMT, Ilmenau, Germany*

Correspondence should be addressed to Jakob Abeßer (jakob.abesser@idmt.fraunhofer.de)

ABSTRACT

The deployment of machine listening algorithms in real-world application scenarios is challenging. In this paper, we investigate how the superposition of multiple sound events within complex sound scenes affects their recognition. As a basis for our research, we introduce the Urban Sound Monitoring (USM) dataset, which is a novel public benchmark dataset for urban sound monitoring tasks. It includes 24,000 sound scenes that are mixed from isolated sounds using different loudness levels, sound polyphony levels, and stereo panorama placements. In a benchmark experiment, we evaluate three deep neural network architectures for sound event tagging (SET) on the USM dataset. In addition to counting the overall number of sounds in a sound scene, we introduce a local sound polyphony measure as well as a temporal and frequency coverage measure of sounds which allow to characterize complex sound scenes. The analysis of these measures confirms that SET performance decreases for higher sound polyphony levels and larger temporal coverage of sounds.

1 Introduction

The recognition of individual sounds is a crucial task in the analysis of complex sound scenes, which surround us every day [1]. The human auditory system can easily identify and focus on different sound sources in our surrounding while computational methods often struggle with this task. As a first reason for this, environmental sounds have diverse characteristics and often include short transients, wide-band noise, and harmonic signal components. Second, the sound duration ranges from very short events such as gun shots and door knocks to very long and almost stationary sounds such as running machine. Third, if multiple sounds appear simultaneously, they overlap and blend into novel sound mixtures, which complicates their recognition.

As a first contribution of this paper, we introduce the Urban Sound Monitoring (USM) dataset[1], which comprises of five second audio recordings of various polyphonic sound scenes. These scenes were created by mixing isolated sounds taken from the public FSD50K dataset [2] with a focus on sound classes relevant for urban sound monitoring. The USM dataset is intended as public benchmark for various machine listening tasks such as sound event tagging (SET) and localization, source separation, and sound polyphony estimation. As a second contribution, we evaluate three different convolutional neural network (CNN) architectures of different complexity for SET on the USM dataset. In comparison to sound event detection (SED), SET only deals with a classification of active sounds without their precise temporal localization. In our experiments, we investigate the influence of the

[1] A detailed dataset description including a download link can be found at https://github.com/jakobabesser/usm.

Table 1: Comparison between the USM dataset and three related sound event detection datasets.

	FSD50K	**FUSS**	**URBAN-SED**	**USM**
Sound duration	0.3 s - 30 s	10 s	10 s	5 s
Sound classes	200	357	10	26
Audio files	51,197	22,000	10,000	24,000
Polyphony level	1	1-4	1-9	2-6
Sounds (foreground/background)	1	1-3/1	1-9/1	1-3/1-3

sound polyphony on the SET performance. We introduce novel measures based on the average local sound polyphony as well as on the temporal and frequency coverage in the time-frequency domain, which allow to estimate the difficulty of the SET task for different polyphonic sounds scenes.

2 Urban Sound Monitoring (USM) Dataset

In this section, we introduce the USM dataset, which includes 24,000 polyphonic sound scenes created by mixing isolated sound samples from the FSD50K dataset. The USM dataset provides a test-bed for various sound monitoring applications such as SET, sound polyphony estimation, loudness estimation, sound localization, and source separation, which will be detailed in Section 2.4.

Table 1 compares the USM dataset with the existing FSD50K [2], FUSS [3], and URBAN-SED [4] datasets with respect to the duration of the included sound mixtures, the number of sound classes, the total number of audio files, the sound polyphony level (i. e., the number of simultaneous sounds), as well as the number of sounds positioned in the foreground (predominant) or background (less prominent). The FSD50K dataset contains only isolated sounds, whereas the other three datasets include polyphonic sound mixtures. While the FSD50K and FUSS datasets cover hundreds of sound classes, the USM and URBAN-SED datasets focus on smaller subsets of 26 and 10 sound classes, respectively. At the same time, the latter two datasets include mixtures with higher sound polyphony levels of up to six and nine sounds within mixtures of five and 10 seconds duration, respectively. In contrast to the URBAN-SOUND dataset, the USM dataset omits the addition of artificial background noises, which can lead to potential confusion with texture-like sound classes, especially for the task of sound polyphony estimation.

Table 2: Mapping between FSD50K sound classes (second column) to 26 sound classes included in the USM dataset (first column), which are grouped to six sound categories.

USM Class	**FSD50K Classes**
(1) Miscellaneous sounds	
- siren	ambulance (siren), emergency vehicle, fire truck, siren
- gunshot	gunfire, machine gun
- glass break	glass, shatter
- church bell	church bell
- alarm	alarm, car alarm
- lawn mower	lawn mower
(2) Climate sounds	
- wind	wind
- rain	rain
- thunderstorm	thunder, thunderstorm
(3) Animal sounds	
- birds	bird
- dogs	bark
(4) Human-made sounds	
- music	music
- singing, cheering, applause	applause, booing, cheering, crowd
- speech	kid speaking, conversation, woman speaking, male speech, man speaking, speech
- scream	screaming, shout
(5) Construction site sounds	
- sawing	chainsaw, sawing
- hammer	hammer
- jackhammer	jackhammer
- drilling	drill, power tool
(6) Vehicle sounds	
- car	car
- truck	truck
- bus	bus
- motorcycle	motorcycle
- train/tram	underground, train
- airplane	aircraft engine, airplane
- helicopter	helicopter

2.1 Class Taxononmy

Table 2 summarizes the 26 sound classes covered in the USM dataset. For each of these sound classes, the first column lists the USM class label and the second column lists semantically corresponding sound classes in the FSD50K dataset. For instance, both samples from the "glass" and "shatter" class were taken from the FSD50K dataset and mapped to the "glass break" class in the USM dataset. These 26 sound classes can be grouped into six categories: miscellaneous sounds, climate sounds, animal sounds, human-made sounds, construction site sounds, and vehicle sounds.

2.2 Sampling Procedure & Dataset Split

The USM dataset is divided into three subsets: a training set (20,000 sound scenes), a validation set (2,000 sound scenes), and an evaluation set (2,000 sound scenes). The training and validation sets are derived from samples of the FSD50K development set and the evaluation set only includes samples from the FSD50K evaluation set. Sound samples used to create the training and validation set are strictly separated to avoid a sample bleeding between both sets. A total of 24,424 unique sound samples are selected from the FSD50K dataset. These samples are published under one of the three licenses CC0, CC BY, and CC Sampling+, and allow for content remixing as well as for commercial application.

Figure 1 illustrates the sound duration distribution over the samples from each of the 26 sound classes. This distribution reproduces mostly natural sound characteristics. For instance, while short events such as alarms, breaking glass, or gunshots show lower duration values, longer lasting sounds like church bells, helicopter, rain, or thunderstorm exhibit higher values. Notably, sounds from the music class consistently have short duration values below 15 seconds which indicates that the FSD50K dataset mostly includes one-shot instrument samples instead of longer music recordings. The initial audio sample selection shows a strong class imbalance. However, as will be explained in Section 2.3, the random sampling procedure leads almost to an equal sound class distribution in the USM dataset.

2.3 Sound Scene Rendering

The following iterative procedure is used to render the *i*-th sound scene in the USM dataset. First, we randomly select the number of sounds mixed in the foreground $N_i^{\mathrm{F}} \in [1:3]$ and sounds mixed in the background $N_i^{\mathrm{B}} \in [1:3]$. The resulting number of sounds, i. e., the sound polyphony level, is $L_i = N_i^{\mathrm{F}} + N_i^{\mathrm{B}}$. Then, we randomly select L_i source samples from L_i different sound classes and assign them either to the foreground or background sounds. For each selected source sample, we randomly crop a five second long segment. Since samples in the FSD50K only have weak sound class labels, this cropping procedure might introduce label noise in the USM dataset if a five second long segment from a longer sample was selected without the annotated sound actually being audible. If the original sample duration is smaller than five seconds, the sample is placed at a random position within the five seconds, which may cause silence segments in the beginning. If the original sample duration is larger than five seconds, we randomly crop a segment of five seconds from it. Our intuition is that humans easily recognize such

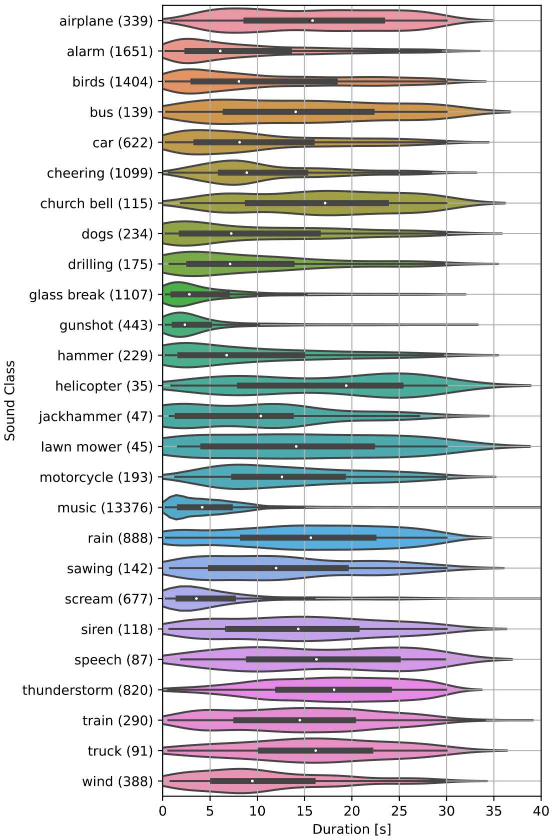

Fig. 1: Distribution of sound durations in the source samples taken from the FSD50K samples. The number of unique source samples per sound class is given in brackets. Duration range limited to 40 s for better readability.

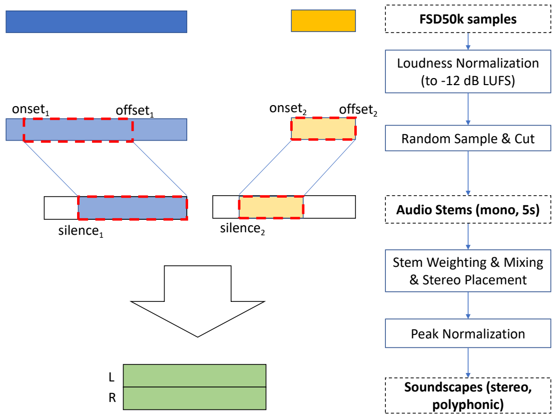

Fig. 2: Audio synthesis steps to render polyphonic sound scenes as described in Section 2.3. The sample cropping procedure is illustrated for an audio sample longer than 5 s (blue) and an audio sample shorter than 5 s (yellow).

sounds event if their duration is truncated to 5 seconds.

The new sample arrays of five seconds duration are denoted as $x_{i,j} \in \mathbb{R}^{5 \cdot f_s \times L_i}$ with j indexing the source sample to be mixed. Both cases are illustrated in Figure 2. We use the *pyloudnorm* Python package [5] to normalize all stems $x_{i,j}$ to the same perceived loudness of -12 dB LUFS based on ITU-R BS.1770-4 specification[2]. For each soundscape, we randomly sample mixing coefficients $\alpha_{i,j} \in [-20,-8]$ dB for the background sounds and $\alpha_{i,j} \in [-6,0]$ dB for the foreground sounds. Similarly, we randomly sample stereo panning coefficients $\beta_{i,j} \in [0,1]$ for each sound with 0 indicating a left panning and 1 indicating a right panning.

Finally, we render polyphonic stereo signals from the selected stems using the following steps. We use the same sample rate of $f_s = 44.1$ kHz as in the FSD50K dataset. First, we compute the mixing coefficients as $\hat{\alpha}_{i,j} = 10^{\frac{\alpha^{i,j}}{20}}$. Secnod, we normalize the mixing coefficients as $\hat{\alpha}_{i,j} \leftarrow \frac{\hat{\alpha}_{i,j}}{\sum_j \hat{\alpha}_{i,j}}$. Finally, we mix mono samples to two channels $s \in \mathbb{R}^{5 \cdot f_s \times 2}$ as $s_{i,0} = \sum_{j=1}^{L_i} (1-\beta_{i,j})\ \hat{\alpha}_{i,j} x_{i,j}$ (left channel) and $s_{i,1} = \sum_{j=1}^{L_i} \beta_{i,j}\ \hat{\alpha}_{i,j} x_{i,j}$ (right channel), which are stored as stereo audio file. This procedure is used generate around 160,000 unique stems from which we mix 24,000 sound scenes.

2.4 Application Scenarios

In this section, we will briefly discuss possible application scenarios, which the USM dataset can be used for as a benchmark dataset.

Sound Event Tagging and Detection In several application scenarios in urban environments, recognizing different sound classes is a key functionality. In acoustic traffic monitoring, vehicle types such as cars, trucks, and busses need to be distinguished [6]. During public events such as concerts, detecting rare sound events such as gunshots or bomb explosions allows to anticipate panic situations and to alert security authorities immediately. Audio-based construction site monitoring systems allow to recognize typical working steps such as drilling, hammering, or sawing and oversee the construction progress. Security monitoring applications require to detect rare sound events such as breaking glass, which can indicate burglary into buildings or apartments. Over the last years, automatic noise monitoring systems were developed to identify the most disturbing sound sources in urban environments [7, 8]. Furthermore, the recognition of specific animal vocalizations is useful for bioacoustic monitoring tasks [9].

While SED combines the temporal detection and classification of sound events in audio recordings, the USM dataset only provides weak sound annotations. However, we believe that SET on relatively short segments (five seconds) is a meaningful proxy task for SED as it also provides a rough and often sufficient temporal resolution in practical application scenarios.

Sound Polyphony Estimation The sound polyphony, i. e., the number of simultaneously audible sounds, influences the timbral complexity of a sound scene. In other research fields, the term “polyphony” is used in a similar fashion to measure the number of simultaneously sounding musical notes (music information retrieval) or the number of speakers in a recording (automatic speech recognition). When using the USM dataset, we relax the requirement of “simultaneously audible” sounds in such way that we measure sound polyphony as the number of audible sound classes within a 5 s long sound scene recording.

[2] `https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-4-201510-I!!PDF-E.pdf`

Sound Event Localization Sound event localization aims for a spatial localization of different sound sources by estimating their azimuth and elevation relative to the audio recording device. Datasets such as the one used in the DCASE 2019 challenge task “Sound Event Localization and Detection” combine room impulse responses (RIR) measured in different recording locations, ambient (non-directional) noise components, and a synthetic mixing of polyphonic sound scenes. As discussed before in Section 2.3, we used a simpler approach to synthesize sound scene synthesis in the USM dataset by positioning sounds in the stereo panorama using the level difference approach.

Source Separation Only recently, researchers began to put a stronger focus on the application of source separation algorithms on environmental sound mixtures [10, 11, 3]. The USM dataset includes both the audio mix (sound scene) and the corresponding single tracks (stems) and hence provides a suitable test-bed for source separation algorithms.

2.5 Critical Discussion & Dataset Limitations

The dataset generation procedure explained in Section 2.3 goes along with certain disadvantages and limitations, which will be discussed in this section.

Fixed Sound Scene Duration Since sound events have a wide range of durations, the choice of a fixed sample duration of five seconds (such as in the ESC-50 dataset [12]) might truncate longer sounds and make them harder to recognize. The choice of five seconds is a trade-off between common sound durations (compare Fig. 5, [2]) and the requirement for near real-time sound recognition scenarios as discussed in Section 2.4.

Stereo Sound Source Placement In contrast to real-life sound scenes, where sound sources such as vehicles are moving, sounds in the USM dataset are located as static sources at random positions in the stereo panorama. Also, restricting the dataset to a stereo setup with two audio channels is a simplification compared to similar datasets for sound event localization, which include spatial audio recordings with multiple audio channels. Similarly to the fixed sound scene duration of 5 seconds, the choice of a stereo audio setup is motivated by practical considerations of low-cost acoustic sensors in urban sound monitoring application scenarios.

Noise & Label Noise & Microphone Characteristics As a consequence of the mixing process, the USM sound scenes directly derive characteristics from the audio samples in the FSD50K dataset. The loudness normalization of these samples prior to the sound scene mixing can potentially boost the underlying noise levels. Existing label noise based on incomplete or erroneous annotations directly propagates to the USM dataset (see Section IV.C [2]). We will show in Section 3.4 that the label noise by randomly selecting five second long segments from longer recordings is negligible. Audio samples in the FSD50K dataset come from different uploaders in the FreeSound database and hence are recorded with different microphone setups [2]. The mixing procedure explained in Section 2.3 consequently can lead to unrealistic blendings of different microphones characteristics. Nevertheless, a positive side-effect might be that this allows to train sound recognition models, which are more robust to changes in recording conditions.

Sound Scene Realism Real soundscapes are often characterized by typical background noises as well as occasional unknown sound events, which are not within the discussed sound taxonomy. However, we decided not to include such components in the synthesis process of the USM dataset in order to have a complete annotation about all audible sounds, which is of importance for the sound polyphony estimation as well as for the sound event tagging task. In its current form, the USM dataset does not include acoustic effects due to sound reflections on buildings, microphone directionality, as well as reverberations, which could be a potential extension of this work. The random selection of source samples in the mixing procedure often leads to unrealistic sound scenes. As a result, sound recognition algorithms trained with the USM dataset will not be biased towards common sound co-occurrences in acoustic scenes.

Scaper Salamon et al. published in [4] the Scaper library for sound scene synthesis. It covers most requirements, which arise from the dataset creation process described in Section 2.3. However, there are two main differences in both synthesis approaches, why we decided not to use the library. First, in Scaper, a continuous texture-like sound is required as background sound, which should not contain any prominent sound events and is therefore not included in the sound annotation of the resulting sound scene. Since one goal of the USM dataset is to allow for sound polyphony estimation, we aim for a complete annotation of all audible sound events allowing the USM dataset. Secondly, the Scaper library was not designed to randomly position individual sound sources in the stereo panorama.

3 Classifying Sounds in Polyphonic Sound Scenes

In order to create a baseline for the USM dataset, we investigate the performance of three deep neural network architectures for sound event tagging (SET) using the USM dataset.

3.1 Audio Features

Since sound event location was not tackled in this paper, we averaged both stereo channels to one mono channel. As audio features, we compute log-magnitude scaled mel-spectrograms with 128 bins using a hopsize of 441 samples (10 ms) and an FFT size of 1024 samples (23.2 ms) at a sample rate of 44.1 kHz. Each five second audio file is represented by 501 time frames. For normalization purpose, we scale each audio signal to a maximum absolute value of 1 and apply no additional feature normalization.

3.2 Neural Network Architectures

We compare three neural network architectures. The first model is a replication of the `VGG-like` model used in [2], which outperformed three larger models based on convolutional recurrent neural networks (CRNN), ResNets, and DenseNets for SED on the FSD50K dataset. The model consists of three groups of convolutional layers—three layers with 32 filters, two layers with 64, and one layer with 128 filters. All layers use 3x3 kernels followed by batch normalization and a ReLU activation function. Between each layer group, a 2x2 max pooling operation is applied. After the convolutional front-end, both global max pooling and global average pooling are applied and the results are concatenated before two final dense layers with 256 units and 26 units are passed. The model has around 259k parameters. The second model `MN-S` and the third model `MN-M` are MobileNetV2 model variants [13] using width multiplier values of $\alpha = 0.35$ leading to 444k parameters and $\alpha = 1$ leading to 2.2M parameters, respectively. The MobileNetV2 combines several improvements such as depthwise separable convolutions, which approximate traditional convolutional layers using fewer parameters, and residual connections, which improve the gradient flow through the network. We selected the MobileNetV2 models as this architecture was intended to be used in mobile and resource constrained application scenarios.

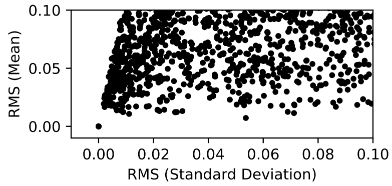

Fig. 3: Clip-wise mean and standard deviation over frame-level RMS values in the sound stems of the USM evaluation set. The figure only shows the lower range of the distribution.

3.3 Model Training

We train all models for the task of polyphonic sound event tagging (SET), i. e., a multi-label sound classification over a five second audio segment. In particular, we want to investigate the influence of the sound polyphony level in the training data on the models' SET performance on polyphonic sound scenes. We compare two scenarios and train each model by using either only the isolated sounds (stems) of the USM training and validation sets or using only the mixed sound scenes (mixtures). The corresponding models are denoted with the pre-fix "`-s`" and "`-m`", respectively. We use the binary crossentropy as loss function and the Adam optimizer with an initial learning rate of 0.005 and a batch size of 32. We train all models for 200 epochs and use early stopping on the validation set with a patience of 20 epochs. As data augmentation, we apply grid distortion using the Albumentations Python library [14] as well as SpecAugment [15], which combines time stretching and time/frequency masking, with an individual probability of $p = 0.5$.

3.4 Label Noise Induced by Segment Selection

As discussed in Section 2.3, cutting five second long segments from longer sound recordings could potentially create label noise if the segment does not contain the annotated sound anymore. In order to estimate

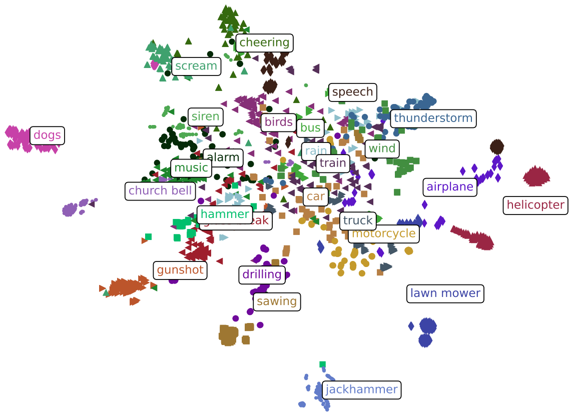

Fig. 4: Distribution of different sound classes in the latent space of the `VGG-Like-s` model (zoom in for better view).

the influence of this type of label noise in the dataset, we investigate 8,007 stems in the USM evaluation set and compute the mean and standard deviation over the frame-level RMS values in each stem. Figure 3 shows this distribution for the lower range of both features. By manually checking the stems with the lowest mean and standard deviation values, we found that only 5 stems (around 0.06 %) with both measures being almost zero actually only contain silence. Therefore, we consider this type label noise to be negligible.

3.5 Latent Space Exploration

In this first experiment, we want to explore how different sound classes distribute in a latent space at an intermediate layer of the `VGG-Like-s` model, which was trained solely on isolated sounds. In particular, we extract 128-dimensional embedding vectors after the convolution layer based network front-end, which should learn to recognize sound-specific patterns in the mel-spectrograms. For the purpose of visualization, these embedding vectors are mapped to a two-dimensional latent space using t-Distributed Stochastic Neighbor Embedding (t-SNE) [16] with a perplexity of 20. This method aims at preserving proximity relationships when mapping data from high-dimensional to low-dimensional feature spaces. Figure 4 illustrates how the 26 sound classes included in the USM dataset distribute over this latent space. In particular, we visualize a random selection of 2,500 sound stems taken from the USM test dataset.

Table 3: Mean average precision (mAP) scores for SET over the USM evaluation set for the compared neural network models.

Model	**Trained on**	**mAP**	**# Parameters**
`VGG-Like-m`	mixes	**0.37**	222k
`VGG-Like-s`	stems	0.29	222k
`MN-S-m`	mixes	0.35	444k
`MN-S-s`	stems	0.26	444k
`MN-M-m`	mixes	0.36	2.2M
`MN-M-s`	stems	0.26	2.2M

Several groups of sounds can be identified. Speech, cheering, and scream are created by humans and cluster in the latent space. Another group of sounds include alarm, music, church bell, and siren, which all have strong harmonic signal components. Sawing and drilling are typical noises from construction sites that have characteristic sound repetition patterns. It can be further observed that environmental sounds such as rain, wind, and thunderstorm are close but overlap with the vehicle sound classes bus and train. Another nearby group of sounds are the vehicle classes car, truck, and motorcycle, which include running engine sounds. Notably, the classes dogs, jackhammer, lawn mower, and helicopter form the most unique cluster in this latent space. A similar latent space structure was observed for the other two model architectures trained on stems.

3.6 Model Comparison for Sound Event Tagging

In this section, we evaluate the models for SET on the USM dataset. Similar to [2], we use the un-weighted mean average precision (mAP) as an evaluation measure to compare different models. The mAP approximates the area under the precision-recall (PR) curve and is not dependent on the applied decision threshold to binarize SET prediction. Here, the average precision (AP) values are computed per class and are averaged over all classes without taking possible class imbalance into account.

Table 3 summarizes the mAP values for different architectures as well as their complexity measured in number of parameters. As a first observation, SET models trained only on isolated stems perform significantly worse on mixed sound scenes than models trained on mixes, particularly for higher sound polyphonies. This is understandable since these models were never trained to recognize overlapping sounds.

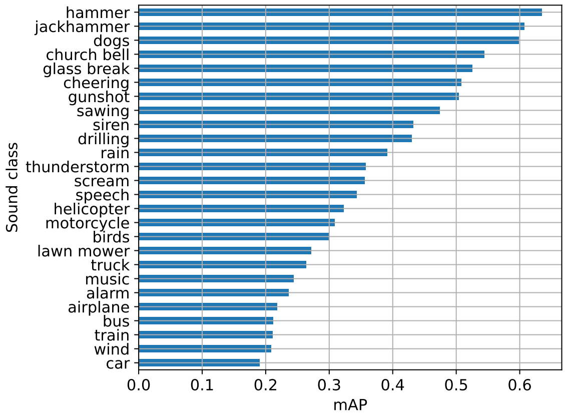

Fig. 5: Class-wise mAP scores for the `VGG-Like-m` SET model sorted in decreasing order.

A complementary and more diverse perspective is provided in Figure 5, where the mAP scores are shown for the best-performing model `VGG-Like-m` separated per sound class and sorted in decreasing order. It can be observed that especially time-localized sounds such as hammer, jackhammer, dog barking, church bells, and glass break (compare also Figure 1) are easier to recognize. Longer sounds with noise-like characteristics such as wind or passing/running vehicles such as cars, trains, busses, or airplanes are harder to distinguished for the model. Presumably, one reason for this is that these sounds are typically longer than the five second sound scene recordings we analyze here.

3.7 Influence of Sound Polyphony and Sound Coverage

In this section, we further investigate how the SET performance is affected by the way multiple stems overlap in the time-frequency domain. In addition to the global sound polyphony P_G, which measures the number of unique sound events in a given sound mixture, we define a *local sound polyphony* measure $P_L \in \mathbb{R}$ as the average number of overlapping sounds per time-frequency bin. We compute P_L as follows. As shown in Figure 6, we first compute the log-magnitude scaled mel-spectrograms $X_i \in \mathbb{R}^{K \times N}$ for all underlying stems with the indices $i \in [1:P_G]$ for $K = 128$ mel bands and $N = 501$ time frames. We then apply a threshold operation

$$A_i(k,n) = \begin{cases} 1 & \text{if } \tilde{X}_i(k,n) \geq \tau \text{, and} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

to derive a binary activity map $A_i \in \{0,1\}^{K \times N}$ over the frequency bins $k \in [1:K]$ and time frames $n \in [1:N]$. The binarization threshold $\tau = 0.05$ is applied to the normalized mel-spectrogram

$$\tilde{X}_i \leftarrow \frac{X_i - \min\{X_i\}}{\max\{X_i\}}. \quad (2)$$

After accumulating the activity maps as $A_{\text{mix}} = \sum_{i=1}^{G_P} A_i$, we compute the *average local sound polyphony* as

$$P_L = \frac{1}{N \cdot K} \sum_{\substack{k \in [1:K] \\ n \in [1:N]}} A_{\text{mix}}(k,n). \quad (3)$$

In addition to the two polyphony measures P_G and P_L, we define the *temporal coverage* C_T and the *frequency coverage* C_F, which describe the fraction of all time frames and frequency bins, respectively, where a sound activity can be observed within the five second long snippets. Both are defined as $C_T = \frac{1}{N} \sum_{n \in [1:N]} a_F(n)$ and $C_F = \frac{1}{K} \sum_{k \in [1:K]} a_T(k)$. with $a_T \in \mathbb{R}^N$ and $a_F \in \mathbb{R}^K$ and

$$a_F(k) = \begin{cases} 1 & \text{if } \sum_{n \in [1:N]} A_{\text{mix}}(k,n) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

$$a_T(n) = \begin{cases} 1 & \text{if } \sum_{k \in [1:K]} A_{\text{mix}}(k,n) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Based on the finding that SET models perform worse with increasing global polyphony P_G as shown in Section 3.6, we now investigate in detail the dependency between the macro-weighted F score (using a common 0.5 threshold) and the polyphony measures P_G and P_L and coverage measures C_T and C_F. As can be seen in Figure 7, the average local sound polyphony P_L shows a stronger negative Pearson correlation coefficient ($r = -0.16$) compare to the global sound polyphony P_G ($r = -0.11$). Both correlations are significant ($p < 0.05$). Our interpretation is that P_L better reflects how different sounds overlap in the time-frequency space and therefore is better suited to estimate the difficulty of the SET task for a given sound scene compared to P_G.

As a second result, the SET performance decreases with increasing temporal coverage C_T ($r = -0.11$). This indicates that sound mixtures, which are more clearly localized in time, are easier to recognize. At the same

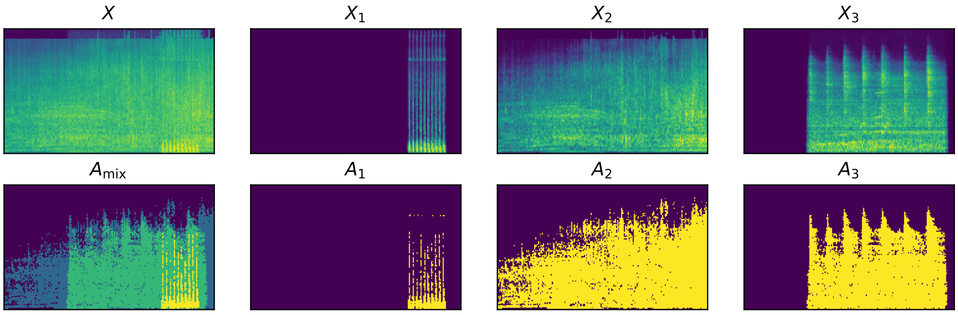

Fig. 6: Example of a polyphonic sound scene from the USM dataset as well as the underlying sounds. In the upper row, log-magnitude mel-spectrograms of the sound scene (X) and the individual stems (X_i) are shown and in the lower row, the local sound activity matrices A_i as well as the local sound polyphony matrix A_{mix} are shown. All figures display distributions of over mel-frequency (y-axis) and time (x-axis).

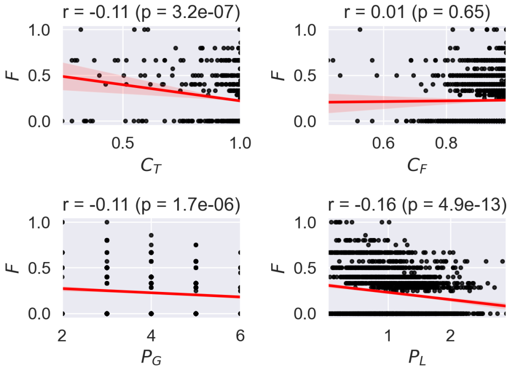

Fig. 7: Scatter plots (black) combined with linear regression lines (red) shown for the relationships between the per-sample F score and the temporal coverage C_T, frequency coverage C_F, global sound polyphony P_G, as well as the local sound polyphony P_L. Metrics are computed for the `VGG-Like-m` SET model over the USM evaluation set. Pearson correlation coefficient r and corresponding p-values are provided in the figure titles.

time, the frequency coverage C_F shows no significant correlation with the F score.

To summarize, both the sound polyphony measures P_G and P_L as well as the coverage measure C_T allow to make predictions about the reliability of SET algorithms for polyphonic sound scenes. However, while C_T can be easily determined from the mel-spectrogram of a sound scene recording, the automatic estimation of P_G and P_L remains an open research question for future work.

4 Conclusion

In this paper, we investigate the challenging task of recognizing sounds in complex sound scenes. We focus on urban acoustic scenarios and introduce the novel USM dataset, which includes synthetically-mixed sound scenes. These sound scenes were created by mixing isolated sounds taken from the FSD50K library. The mixing process is controlled by defining the number of sounds (sound polyphony) as well as their individual levels and positions in the stereo panorama. In a benchmark experiment, we evaluate three different deep neural networks for SET on the USM dataset. These models cover different model sizes as well as a VGG-like and MobileNetV2 architecture. Each model is trained as two variants using either isolated sound stems or polyphonic sound scenes as training data. We confirm similar to [2] that the small VGG-based model is capable to outperform larger models (compare Table 3).

During an initial latent space exploration of the SET models, we find that semantically related groups of sounds such as vehicle sounds or construction site sounds indeed cluster together. Furthermore, our results verify the common assumption that the SET performance of CNN-based models decreases with increasing sound polyphony level. As another contribution, we propose both an average local sound polyphony measure as well as a temporal and frequency coverage measure, which can characterize the sound overlap in the time-frequency domain. These measures allow to better understand the SET performance for polyphonic sound scenes.

Acknowledgements

This work has been supported by the German Research Foundation (AB 675/2-2). We would like the reviewers for their valuable suggestions and comments.

References

[1] Virtanen, T., Plumbley, M. D., and Ellis, D., editors, *Computational Analysis of Sound Scenes and Events*, Springer International Publishing, Cham, Switzerland, 2018.

[2] Fonseca, E., Member, S., Favory, X., Pons, J., Font, F., and Serra, X., "FSD50K: an Open Dataset of Human-labeled Sound Events," *arXiv preprint arXiv:2010.00475*, 2020.

[3] Wisdom, S., Erdogan, H., Ellis, D. P. W., Serizel, R., Turpault, N., Fonseca, E., Salamon, J., Seetharaman, P., and Hershey, J. R., "What's all the FUSS about free universal Sound Separation," *arXiv preprint arXiv:2011.00803v1*, 2020.

[4] Salamon, J., MacConnell, D., Cartwright, M., Li, P., and Bello, J. P., "Scaper: A library for soundscape synthesis and augmentation," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 344–348, New Paltz, NY, USA, 2017.

[5] Steinmetz, C. J. and Reiss, J. D., "pyloudnorm: A simple yet flexible loudness meter in python," in *Proceedings of the 150th Audio Engineering Society (AES) Convention*, Virtual, 2021, ISBN 9781713830672.

[6] Abeßer, J., Gourishetti, S., Kátai, A., Clauß, T., Sharma, P., and Liebetrau, J., "IDMT-Traffic: An Open Benchmark Dataset for Acoustic Traffic Monitoring Research," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 551–555, Dublin, Ireland, 2021.

[7] Bello, J. P., Silva, C., Nov, O., DuBois, R. L., Arora, A., Salamon, J., Mydlarz, C., and Doraiswamy, H., "SONYC: A System for the Monitoring, Analysis and Mitigation of Urban Noise Pollution," *Communications of the ACM (CACM)*, 62(2), pp. 68–77, 2018.

[8] Abeßer, J., Götze, M., Clauß, T., Zapf, D., Kühn, C., Lukashevich, H., Kühnlenz, S., and Mimilakis, S., "Urban Noise Monitoring in the Stadtlärm Project - A Field Report," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE)*, New York, NY, USA, 2019.

[9] Stowell, D., "Computational bioacoustics with deep learning: a review and roadmap," *arXiv preprint arXiv:2112.06725*, 2021.

[10] Sudo, Y., Itoyama, K., Nishida, K., and Nakadai, K., "Environmental sound segmentation utilizing Mask U-Net," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5340–5345, Macao, Macau, 2019.

[11] Tzinis, E., Wisdom, S., Hershey, J. R., Jansen, A., and Ellis, D. P. W., "Improving Universal Sound Separation using Sound Classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 96–100, Barcelona, Spain, 2020.

[12] Piczak, K. J., "ESC: Dataset for environmental sound classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018, Brisbane, Australia, 2015.

[13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, Salt Lake City, UT, USA, 2018.

[14] Buslaev, A., Parinov, A., Khvedchenya, E., Iglovikov, V. I., and Kalinin, A. A., "Albumentations: fast and flexible image augmentations," *Information*, 11(2), 2020.

[15] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[16] van der Maaten, L. and Hinton, G., "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, 9, pp. 2579–2605, 2008.