Audio-Based Activities of Daily Living (ADL) Recognition with Large-Scale Acoustic Embeddings from Online Videos

DAWEI LIANG, University of Texas at Austin, USA
EDISON THOMAZ, University of Texas at Austin, USA

Over the years, activity sensing and recognition has been shown to play a key enabling role in a wide range of applications, from sustainability and human-computer interaction to health care. While many recognition tasks have traditionally employed inertial sensors, acoustic-based methods offer the benefit of capturing rich contextual information, which can be useful when discriminating complex activities. Given the emergence of deep learning techniques and leveraging new, large-scale multimedia datasets, this paper revisits the opportunity of training audio-based classifiers without the onerous and time-consuming task of annotating audio data. We propose a framework for audio-based activity recognition that can make use of millions of embedding features from public online video sound clips. Based on the combination of oversampling and deep learning approaches, our framework does not require further feature processing or outliers filtering as in prior work. We evaluated our approach in the context of Activities of Daily Living (ADL) by recognizing 15 everyday activities with 14 participants in their own homes, achieving 64.2% and 83.6% averaged within-subject accuracy in terms of top-1 and top-3 classification respectively. Individual class performance was also examined in the paper to further study the co-occurrence characteristics of the activities and the robustness of the framework.

CCS Concepts: • **Human-centered computing → Empirical studies in ubiquitous and mobile computing**;

Additional Key Words and Phrases: Activity Recognition, Deep Learning, Multi-Class Classification, Audio Processing

ACM Reference Format:
Dawei Liang and Edison Thomaz. 2019. Audio-Based Activities of Daily Living (ADL) Recognition with Large-Scale Acoustic Embeddings from Online Videos. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 17 (March 2019), 18 pages. https://doi.org/10.1145/3314404

1 INTRODUCTION

Sensing and recognizing human daily activities has been demonstrated to be useful in many areas, from sustainability to health care. For example, older adults in their own homes can benefit from proactive assistance and monitoring as a way to "live-in-place" and not be forced to move to an assisted-living or nursing facility. While on-body inertial sensors such as accelerometers and gyroscopes are popular in many human activity recognition applications, prior work suggests that they are not effective at recognizing complex and multidimensional activities on their own [2, 18, 29]. Audio, on the other hand, offers much promise in this respect; many daily activities generate characteristic sounds that can be captured with any off-the-shelf device with a microphone. Hence, researchers have proposed several different types of audio event recognition frameworks over the years, from applications on wearable and mobile devices [30, 37] to home-based sensor systems [5, 21]. With the development of deep neural networks in recent years, several efforts have been made by researchers to model large-scale

Authors' addresses: Dawei Liang, dawei.liang@utexas.edu, University of Texas at Austin, Austin, USA; Edison Thomaz, ethomaz@utexas.edu, University of Texas at Austin, Austin, USA.

2474-9567/2019/3-ART17 $15.00
https://doi.org/10.1145/3314404

acoustic events. These include the usage of deep learning for sound classification on existing datasets [33] and the recognition of acoustic categories in the wild [19]. However, most such frameworks suffer from the laborious collection of ground truth training data. Some researchers have explored the use of crowd-sourced data to alleviate the problem, such as Nguyen et al. and Rossi et al. [27, 31]. Despite encouraging results, these methods have proven difficult to scale as they partially rely on human input or interaction.

Large-scale, open-source audio collections now offer a rich source of audio data reflecting a large number of everyday activities. In this work, we present a novel scheme to recognize activities of daily living in the home. Instead of directly collecting ground truth data and labels from users as in most prior research, we explored the feasibility of using large scale of audio embeddings from general-sourced YouTube videos as the only training set. Due to the considerable size and highly unbalanced characteristics of the on-line data, our method combines both oversampling and deep learning approaches. The contributions of this work can be summarized as:

- A novel ambient audio-based framework for recognizing activities of daily living that relies exclusively on 519,270 audio embedding features of online Youtube videos from a large-scale audio dataset.
- An evaluation of the framework with 14 subjects in their homes and 15 activities of daily living, including a performance analysis of the impact of balanced vs. unbalanced audio classes and embedding segmentation. The proposed method showed promising results using off-the-shelf smart phones and was robust to environmental variability.

2 RELATED WORK

Activity recognition is centered on sensor data collection and processing. For the most part, recognition problems have been framed around specific activities and the utilization of single sensing methods. While multiple sensing modalities can often improve recognition, the utilization of multiple sensors introduce new challenges across the entire pipeline, such as lack of sufficient training data, constraints around power and computing, and difficulty synchronizing sensor data for processing and annotation. Subject and location sensitivity can also make it harder for the generalization of activity models [39].

2.1 Inertial Sensing

A large number of activity recognition approaches rely on inertial sensors such as accelerometers and gyroscopes embedded in smartphones [2, 18], smart watches [35, 36] and wearables [29]. For example, Kwapisz et al. [18] managed to recognize walking, jogging, going upstairs, going downstairs, sitting and standing by sensing with a smartphone in subjects' pockets. Thomaz et al.[36] proposed the usage of 3-axis accelerometers embedded in an off-the-shelf smart watch for detecting eating moments. Similarly, Ravi et al. [29] showed the feasibility of attaching a sensor board to the human body for simple movement classification.

2.2 Audio Sensing

Researchers have also proposed activity recognition methods leveraging audio and video resources. Microphones have the benefits of simplicity and flexibility for implementation. Eronen et al. [8] proposed a pilot study to recognize activity context based on sounds by using statistical learning methods. Yatani and Truong [40] explored the recognition of 12 activities related to throat movement such as eating, drinking, speaking and coughing by acoustic data collected around the neck and throat region. This was achieved by using a wearable headset consisting of a tiny microphone and a Bluetooth module. Another study showed that human eating activity can also be effectively inferred by using wrist-mounted acoustic sensing [37]. This implies the practicality of simple audio-based activity recognition with off-the-shelf devices such as smartwatches. With rapidly improving smartphones in recent years, phone-based acoustic sensing also shows great promise for activity recognition tasks. The *AmbientSense* application [30] is an example. It is an Android app that can process ambient sound data

in real-time either on the device or on an server. It was tested on mainstream smartphones (i.e., Samsung Galaxy SII, Google Nexus One) and yielded satisfactory results on the classification of 23 context of daily life. In 2009, Lu et al. [24] developed *SoundSense* to detect speech, music and ambient sound categories based on a mobile platform. Acoustic sensing can also be used for indoor scenarios, especially when video-based methods may bring privacy concerns. Laput et al.[21] described the concept of general-purpose sensing where multiple sensor units including a microphone were embedded in a single home-oriented sensor tag. Chen et al. [5] provided an audio solution for detection of 6 common activities in the bathroom based on MFCC features. This work targets elder care and is aimed at replacing more direct behavioral observations, which individuals might not be comfortable sharing with clinicians. More recently, acoustic sensing and recognition have been significantly improved based on the usage of deep learning techniques. Salamon and Bello [33] proposed an architecture combining feature augmentation and a CNN to evaluate on-line audio data. Lane et al. [19] developed *DeepEar* to classify multiple categories for different sensing tasks based on a well-tuned fully connected network.

2.3 Audio-Based Classification Using Online Data

Most of the prior work requires manual collection of ground truth audio data from individual users. This can be quite laborious especially when targeting multiple classes of activities. Also, it is unreasonable to have to rely on end-users to train the model on their own before using it. Hence, Hwang and Lee [16] introduced a crowd-sourcing framework for the problem. They developed a mobile platform to collect audio data from multiple users. The platform could then generate a global K-nearest neighbors (KNN) classifier based on Gaussian histogram of MFCC features to recognize basic audio scenes. However, this still requires collection of user data and the performance of the system highly depends on the size and quality of the training set. General-purposed acoustic database, on the other hand, can potentially serve as an ideal data source to existing systems.

Over the past several years, the *Freesound* database[1] [10] has been one of the most commonly used databases for audio research. Started in 2005 and currently maintained by the Freesound team, it is a crowd-sourced dataset consisting of over 120,000 annotated audio recordings. Variants of Freesounds have also been created; Salamon et al. [34] released the *UrbanSound* database containing 18.5 hours of urban sound clips selected from Freesound. Säger et al. [32] improved the Freesound recordings by adding adjective-noun and verb-noun pairs to the audio tags and constructed a new *AudioPairBank* dataset. Rossi et al. [31] first attempted context recognition based on MFCC features extracted from the on-line Freesound database by using a Gaussian Mixture Model (GMM). However, due to the limited size of the training set (4678 audio samples for 23 target context), the top-1 classification accuracy based on dedicated sound recordings was just 38%. The performance was improved to 57% by manually filtering over one third of the samples as outliers. Nguyen et al. [26, 27] leveraged semi-supervised learning methods to combine the on-line Freesound data with users' own recordings. After manually filtering outliers for quality, they trained a semi-supervised GMM on MFCC features extracted from 163 Freesound audio clips for 9 context classes. The model was then applied to unlabeled user-centric data recorded by smart phones with a headset microphone. The performance was evaluated based on the second half of the user data with an average accuracy of 54% for 7 users. To further improve the performance, Nguyen et al. [26] also presented two active learning mechanisms, where a supervised GMM was first trained on the same Freesound data or well-labeled user data and then interactively queried users for labeling the unlabeled user-centric data. Clearly, from the prior work we can see that the existing crowd-sourced datasets do not generalize well-enough across users, and previous research still needs to rely on user data and manual filtering of outliers for better performance.

With the introduction of large audio datasets such as the *AudioSet* database [12], the idea of domain adaptation from the web has been developed in several activity recognition research. Hu et al. [15] proposed to use web search text as a bridge for similarity measures between sensor readings. Fast et al. [9] developed *Augur*, a system

[1] https://freesound.org/

leveraging contexts from on-line fictions to predict human activities in the real world. In terms of audio-based classification, Aytar et al. [3] described the *SoundNet* framework for knowledge transfer between large-scaled videos and target sounds based on a deep CNN. To the best of our knowledge, however, very few attempts have been made to adapt such tremendous scale of on-line audio samples for real-world activity recognition, and this can be even challenging when leveraging YouTube sound features due to the ambiguous source of the raw videos from movies, cartoons to crowd-sourced data. The most relevant up-to-date achievement was proposed by Laput et al. [20], where the researchers developed a mixed process of audio augmentation for a deep network and combined online sound effect libraries with the Audio Set data for audio context classification. Their work shows promising results when applying the augmentation process with the online sound effect data. However, the performance of the framework dropped significantly when using only the video sounds (i.e. the Audio Set [12] data) without augmentation. Moreover, their work mainly focused on the classification of environmental, and not individual, context. In our research, we aimed to study the feasibility and performance reported from the perspective of individual activity recognition by leveraging only online video sound clips for training. Our in-lab and multi-subject studies showed that the proposed framework was able to yield promising performance even without any feature augmentation or semi-supervised learning techniques.

3 IMPLEMENTATION

3.1 *AudioSet*

In 2017, Google's Sound Understanding team released a large-scale acoustic dataset, i.e., AudioSet [12], endeavoring to bridge the gap in data availability between image and audio research. The AudioSet contains over 2 million audio soundtracks drawn from general YouTube videos. The dataset is structured as a hierarchical ontology consisting of 527 class labels. All audio clips are equally chunked as 10 seconds long and labeled by human experts.

The dataset does not provide original waveforms of the audio clips. Instead, the samples are presented in the form of both source indexes and bottleneck embedding features. The audio index contains information of the audio ID, URL, class labels, and start and end time of the sample within the corresponding source video. The embedding features are generated from a VGG-like deep neural network (DNN) architecture [14] trained on the YouTube-100M dataset. The generation frequency is roughly 1Hz (96 10ms audio frames, i.e. 0.96 seconds of audio per embedding vector). In other words, one embedding vector can describe one second of audio clip, and therefore there are 10 embedding vectors for each audio clip within the dataset. Before released, the embedding vectors have also been post-processed by principle component analysis (PCA) and whitening as well as quantization to 8 bits per embedding element. Only the first 128 PCA coefficients are kept and released.

The original vectors are all stored within TensorFlow [1] Record files. Given the significant size of the embeddings and the lack of convenience for data processing, Kong et al. [17] provided a converted Python Numpy version of the raw embeddings which are adopted in our research. Their converted dataset has been released publicly online[2].

3.2 Class Selection and Labeling

Before implementation, we needed to consider the range of target activities and how to associate class labels in the AudioSet with them. We narrowed our scope to common activities that frequently take place in a home, and activities of daily living (ADL) in particular. Also, the range of our target classes was limited to target activities that are suitable for audio-based recognition. Here 'suitable' means that the sound of the activity could be featured and easily captured in practice. Classes such as 'silence' were also not chosen because the corresponding attributes can be ambiguous from sleeping, standing, to maybe just absence of a person in the

[2] https://github.com/qiuqiangkong/ICASSP2018_audioset

Table 1. Target activities and association with AudioSet [12] labels

Category	Activity Class	Associated AudioSet Labels
Bathroom	Bathing/Showering	Bathtub (filling or washing)
	Washing hands and face	Sink (filling or washing); Water tap, faucet
	Flushing toilet	Toilet flush
	Brushing teeth	Toothbrush
	Shaving	Electric shaver, electric razor
Kitchen	Chopping food	Chopping (food)
	Frying food	Frying (food)
	Boiling water	Boiling
	Squeezing juice	Blender
	Using microwave oven	Microwave oven
Living/Bed room	Watching TV	Television
	Listening to music	Piano
	Floor cleaning	Vacuum cleaner
	Chatting	Conversation; Narration, monologue
Outdoor	Strolling	Walk, footsteps; Wind noise (microphone)

room. Body movement with very weak sound features is not suitable for audio-based recognition as well. Further more, it is not always possible to find an exact matching between the AudioSet labels and the actual activities. In such cases, we adopted an indirect matching process. That is, we first determined the most relevant objects and environmental context associated with the target activities. We then chose AudioSet classes of such objects and contexts as representation of the activities. For example, we used class 'water tap' and 'sink' as representation of 'washing hands and faces' as all three classes involve usage of water and the features are similar. This is actually a subjective process as there is no quantized measurement to determine the similarity between such relevant classes and the actual target classes. For the class 'listening to music', we focused on studying only piano-related musics as examples.

It is noted that the dataset provides a quality rating of audio labels based on manual assessment. Most of the labels have been assessed by experts based on a random check of 10 audio segments within the label. The samples of each label are actually divided into three subsets (*evaluation*, *balanced training*, and *unbalanced training*) for training and evaluation purposes. The evaluation and balanced training sets are of much smaller size than the rest unbalanced training set, and due to the considerable size of samples and factors such as misinterpretation or confusibility, many class labels of the unbalanced training sets are actually of poor rating results. In our framework, we did not consider the sample ratings and we incorporate all three evaluation, balanced training and unbalanced data for our training set.

We therefore determined 15 common home-related activities for the framework. They are associated with 18 AudioSet labels. Table 1 shows the association between our target activities and the AudioSet class labels, and all audio embeddings of the listed AudioSet classes are used as the only training data in our proposed scheme.

3.3 Oversampling

A typical characteristic of the AudioSet data is the unbalanced distribution in terms of the class size. In our implementation, we also removed samples with label co-occurrence among the target classes to ensure mutual exclusiveness, and table 2 shows the number of embedding vectors per class in our raw training set without any sampling process. The totals include embeddings from all three subsets (evaluation set, balanced training set and

unbalanced training set). The actual size for some classes is slightly smaller than they appear in the released AudioSet since we adopted the converted Python Numpy version of features as mentioned. Classes 'chatting' and 'listening to music' have the most embeddings (174,220 and 115,200 respectively) while class 'brushing teeth' has the least (1230), which accounts for 0.7% of the largest class. In other words, the two majority classes account for over half of the whole training set. The unbalanced distribution of the class size leads to highly unbalanced training in our study. As we will see in the dedicated test section, the distribution of training class can heavily affect the recognition performance; therefore, we implemented two oversampling processes to address this issue.

The unbalanced distribution of labels can be affected by two factors. Firstly, the distribution actually reflects the diversity and frequency of the class labels within the source YouTube videos. For example, elements of chatting or music can be captured in a large amount of video topics, from advertisement and news to cartoons. Brushing teeth, on the contrary, appears much less, and typically just in some movie scenes or daily life recordings. Chatting can also involve several modalities according to the speaker's gender, age and the context of the speech, while brushing activities seem to be much more similar among each. Secondly, we are using only samples without label co-occurrence among the target classes. The size of the remaining disjoint data can also affect the actual distribution in our training set.

The effects of unbalanced training on classification have been discussed in past work [4, 13, 23]. Without prior knowledge of the unbalanced priors, a classifier tends to predict the majority classes, and so there should be a higher cost for misclassifying the minority classes [23]. In our scheme, we implemented random oversampling with replacement and synthetic minority oversampling technique (SMOTE) [4] to handle the problem. The process of random oversampling can be divided into two steps. The first is to calculate the sampling size for each minority class, i.e. to calculate the difference of size between the target class and the majority class. Then each minority class is re-sampled with replacement until the sampling size is filled. This method replicates existing data without introducing any extra information into the dataset. The SMOTE, on the contrary, works by adding new elements to the minority classes. It leverages the K-nearest-neighbors (KNN) approach to first generate new data points around the existing data points. Then one of the neighbors is randomly selected as

Table 2. Number of embedding vectors per activity class

Activity Category	**# of Embedding Vectors**
Chatting	174,220
Listening to Music	115,200
Strolling in Courtyard	81,450
Watching TV	22,250
Flushing Toilet	22,190
Floor Cleaning	19,710
Washing Hands and Face	17,080
Frying Food	15,820
Bathing/Showering	14,270
Squeezing Juice	12,600
Shaving	8,570
Using Microwave Oven	8,180
Boiling Water	4,440
Chopping Food	2,060
Brushing Teeth	1,230
Total	519,270

the synthetic new elements and is introduced to the minority class. In our implementation, the oversampling process was developed based on the Python imbalanced-learn package [4, 22]. All parameters were set as default in the imbalanced-learn package version 0.3.3 except that the random state was kept as 0. By the oversampling processes, we actually obtained 2,613,300 embedding vectors in total for the 15 classes. The total was the same for both random oversampling and SMOTE.

3.4 Architecture

Deep learning has been proven to be effective for classification when large amounts of training data is available. Due to the considerable size of audio samples involved in our study, and also to keep the same feature format as released in the AudioSet, we adopted neural networks in our proposed framework for both embedding feature extraction and classification. Figure 1 shows the architecture of the framework. Overall, there are two networks in our design, a pre-trained feature extraction network and a classification network. More specifically, we adopted the pre-trained *VGGish* model [14] as the extraction network and all parameters of the network were fixed during our training process. The classification network consisted of a 1-dimensional convolutional layer and dense layers. The parameters and weights of the classification network were trained and fine-tuned on the AudioSet data. Additionally, we added an embedding segmentation process between the two networks to improve recognition performance.

In AudioSet, the frame-level features of the audio clips were generated by a VGG-like acoustic model pre-trained on the YouTube-100M dataset. To enable researchers to extract the same format of features, Hershey et al. [14] provided a TensorFlow version of the model called *VGGish*. It has been trained on the same YouTube-100M dataset and can produce the same format of 128-dimensional embeddings for every second of audio sample. The VGGish model takes as input non-overlapping frames of 64-bin log mel spectrogram lasting 0.96 seconds each from the

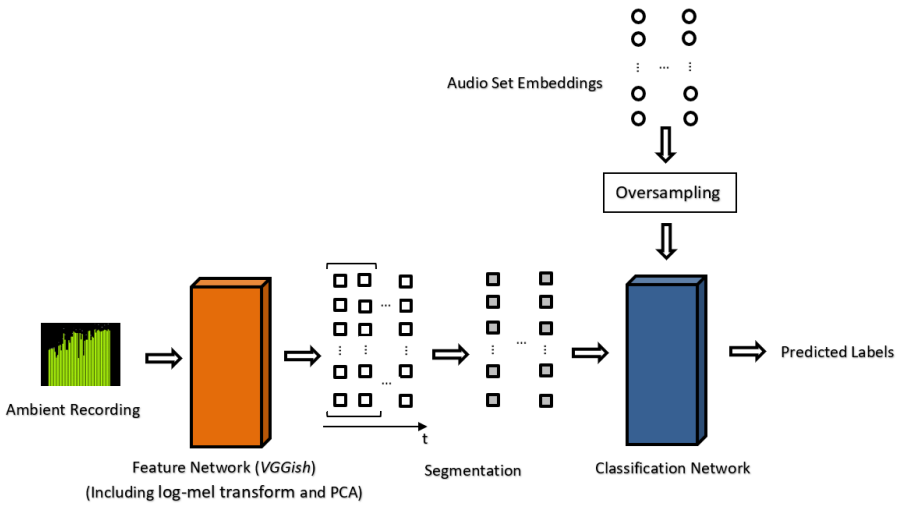

Fig. 1. Architecture of our proposed scheme. We applied the VGGish model [14] as the feature extraction network. The feature network was pre-trained on the YouTube-100M dataset and all parameters were fixed in our training process. The generated embeddings were then segmented and passed to the classification network. Our classification network consisted of a plain 1-dimensional convolutional layers and dense layers, and the model was trained and fine-tuned on the oversampled AudioSet [12] embeddings.

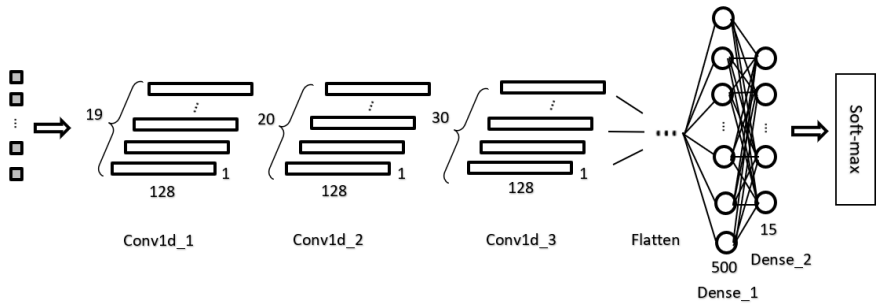

Fig. 2. Architecture of the classification network. The classification network was constructed as a 3 1-dimensional convolutional layers and 2 fully connect (dense) layers. The network takes as input segmented embedding vectors and outputs confidence distribution of the activity labels.

raw audio waveforms. The source code and weights of the pre-trained VGGish model are available in the public AudioSet model GitHub repository[3]. This code repository also includes pre-processing steps for extracting the log mel spectrogram features to feed the model, post-processing steps for PCA transform, and element-wise quantization, which have also been adopted on the released AudioSet data. In our implementation, the audio pre-processing step takes as input audio waveforms with 16 bit resolution, so we manually convert other formats of audio samples (such as raw recordings from smart phones) to the wave format using a free on-line converter[4] before passing the raw audio for processing. The parameters of the VGGish network kept constant during the whole training and validation process. The network outputs a vector of 128 syntactic embeddings for every input instance.

Our classification network consisted of 3 convolutional layers and 2 dense (fully connected) layers. The structure is shown in Figure 2. The convolutional layers were all 1-dimensional tensor with linear activation and consistent paddings to ensure the same feature size. The number of channels are 19, 20 and 30 respectively for the 3 layers. The kernel size was all set as 5 with a stride of 1. We applied 500 neurons for the first dense layer. The second dense layer is the output layer, thus there were 15 neurons and the output activation was set as softmax. A flatten layer was used to connect the convolutional layers and the dense layers. We chose categorical cross entropy as the loss. In terms of the optimizer, we applied stochastic gradient descent with Nesterov momentum. The learning rate was set as 0.001 with 1e-6 decay and 0.9 momentum. The network took as input 128-dimensional segmented and normalized embeddings from the segmentation step of our architecture and output predicted confidence distribution of the labels. Under the top-1 classification scenario, the label with the highest confidence was selected as the final prediction. Our classification network was built and compiled on Python Keras API [6] with the Tensorflow [1] backend. The weights were trained and fine-tuned on normalized AudioSet embeddings.

In addition to the neural network and oversampling steps, we also applied segmentation to change the time length of the audio instance for recognition. This was necessary because the time length of a single embedding vector (0.96 seconds) can be too short to some activities and may not be able to capture enough information for recognition. Also, increasing the instance length can help to alleviate the effects of outliers and noise within the real world recordings. Hence, we introduced a segmentation process on embeddings between the two networks. In our architecture, the segmentation is completed by grouping the embedding vectors continuously in time using a fix-sized window with no overlaps. The vectors are then averaged within each group to yield a new

[3] https://github.com/tensorflow/models/tree/master/research/audioset
[4] https://audio.online-convert.com/convert-to-wav

128-dimensional vector. In other words, each group of audio segment is described by one averaged embedding vector. Activity labels are then assigned to the averaged vectors and those vectors serve as the actual instances for classification. The embeddings were converted as float32 and normalized to [-1, 1] by subtracting and dividing by 128 before fitting to the classification network.

Both the oversampling and training processes were applied using the NVIDIA K40 GPU on the server to accelerate the training process. The training embeddings were split as 90% for training and 10% for validation using the Python Scikit-learn package [28]. The TensorFlow version provided was TensorFlow-GPU 1.0.0 [1]. Before training, we set all random seeds as 0 to ensure the same training status. Besides, a batch of 100 embedding vectors were input each time. The classification network was trained until the validation performance no longer improved (in our study, 15 to 20 epochs depending on the re-sampling set in use).

4 FEASIBILITY STUDY

We evaluated the feasibility of our framework with a pilot study conducted in the home of one participant. There were two reasons for conducting this experiment. Firstly, we hoped to verify if our proposed methodology could actually work on real-world ambient recordings. Although the architecture had been well trained on the AudioSet data, there could be significant differences in the characteristics of the YouTube video sounds and real-world ambient sounds. Secondly, we needed a real-world validation set to determine the best combination strategy for the sampling process and the classifier.

In the pilot study, we collected sounds of target activities by placing an off-the-shelf smart phone (i.e., Huawei P9) near the location where the activities took place; the context of all activities was well-controlled with low variability. Specifically, we excluded irrelevant environmental noise such as sounds of toilet fans or air conditioners during the collection. Also, when a target activity was performed, there were no other on-going activities. The collection was manually started when the sound of the activity could be clearly captured. Sound recordings for each activity lasted for 60 seconds, and it was stopped when the proposed time ended. This same process was repeated for each individual activity until the collection for all 15 activities was completed.

We chose a segmentation window of 10 embedding vectors (9.6 seconds) for the study. The recognition performance was evaluated based on 3 different sampling processes (raw embeddings input/no oversampling, random oversampling, and SMOTE). We also tuned and trained a random forest classifier on the same training sets as a baseline. The random forest was built using the Python Scikit-learn package [28]. We used the overall accuracy and overall F-score as the performance metrics. In binary classification, the F-score is calculated as 2 * (precision * recall) / (precision + recall) and it incorporates information for both precision and recall performance. In our study, the overall F-score across multiple classes can be calculated by finding the weighted average of F-scores of the individual labels. Table 3 shows the recognition performance based on different architectures. For convenience, the random forest is abbreviated as RF in the table. From the results we can see that the random forest without any sampling process yields the worst accuracy and F-score (34.4% and 24.5%). This is

Table 3. Recognition performance leveraging different architectures of implementation.

Architecture	**Accuracy**	**F-Score**
Baseline(RF) + Raw Embeddings	34.4%	24.5%
Baseline(RF) + Random Oversampling	36.7%	27.2%
Baseline(RF) + SMOTE	45.6%	37.1%
CNN + Raw Embeddings	52.2%	44.8%
CNN + Random Oversampling	**81.1%**	**80.0%**
CNN + SMOTE	73.3%	71.1%

comparable to the dedicated study by Rossi et al. [31], where the authors trained a GMM on 4678 raw samples from the crowd-scoured Freesound dataset and obtained 38% overall accuracy for 23 context categories. Clearly,

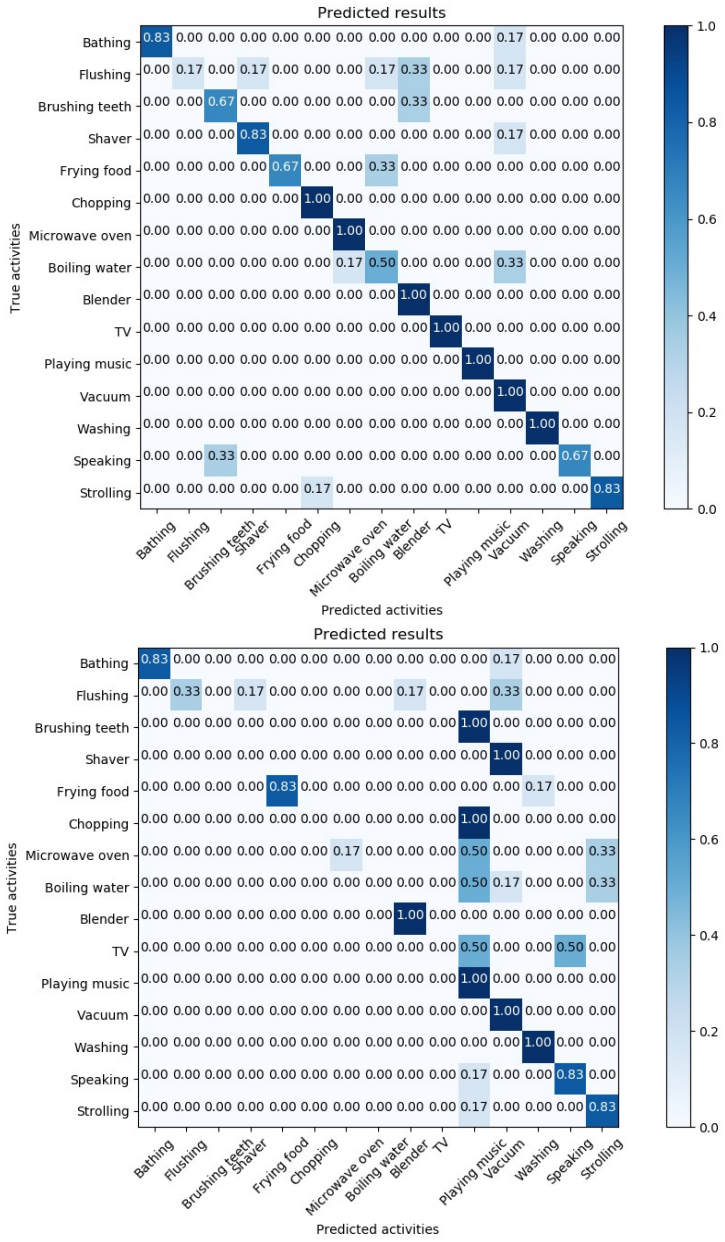

Fig. 3. Recognition results of the pilot study using random oversampling + CNN (top) versus raw embeddings input + CNN (bottom). The performance of the framework with the oversampling far exceeds the performance with only raw embeddings.

the introduction of the classification network significantly improves the recognition performance, especially if combining with the oversampling processes. The combination of random oversampling and our classification network yields the best performance (81.1% overall accuracy and 80.0% overall F-score). Generally, classifiers with oversampling outperform those without one. Figure 3 shows in details the performance of individual classes with and without oversampling, and the entries have been normalized for each class. As it can be seen, classification network input with raw embeddings overfits to some of the majority classes such as 'playing music' and 'strolling'. Network input with the random oversampled embeddings, on the contrary, yields equally promising results to most classes. The worst class for the top-1 architecture was 'flushing toilet' with only 17% class accuracy. This is probably because the segmentation length was too long to the flushing activity and too much irrelevant information was captured within the segments.

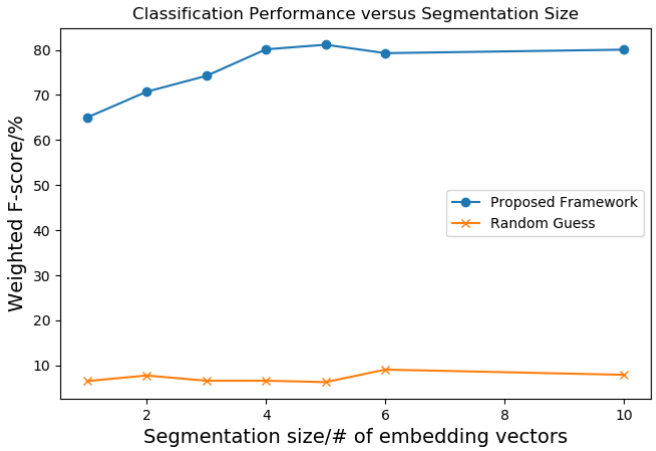

Fig. 4. F-score performance with different segmentation size per window. The performance was the worst when no segmentation was applied. As the segment size increased, so did the F-score; it stabilized around 80%. The random guess levels were around 7%.

To determine how the segmentation process can affect the classification performance, we compared the overall F-score under different sizes of embedding segmentation. The comparison is shown in Figure 4. As reference, we also plotted the random guess levels (around 7%). From the figure, we can see that the performance was the worst when no segmentation process was introduced (i.e. 1 embedding vector each segment), with an F-score of only 65%. By applying a bigger segment size, the F-score value significantly increased to over 80%. In addition, we can see that a unit segmentation length of 5 embedding vectors enabled the instances to capture enough information for classification. Further enlargement of the segmentation size no longer improved the overall recognition performance.

4.1 Transfer Learning and Domain Adaptation

From the perspective of transfer learning, our framework is actually a domain adaptation process where we tried to find a mapping between the source Youtube audio clips and the real-world recordings. Generally speaking, audio features from on-line videos can be very different from those of real-world collections for activity recognition. Interestingly, our classification network only yielded 53% validation and training accuracy on the random oversampled AudioSet embeddings. But the performance of our top-1 scheme reached over 80% on the ambient

recordings. Moreover, we have noticed that the validation performance on the AudioSet data could have been further improved by adopting deeper layers. However, increasing the depth of the model would no longer help to improve the performance on real-world data (it might even harm the performance). A possible reason is that ambient sounds from the real world (especially in home settings) can generally show different characteristics for classification compared to YouTube audio. Hence, a best model for the classification of the AudioSet data may not be the best choice for the classification of real-world sounds.

5 IN-THE-WILD STUDY

With the pilot test, we verified the feasibility of the proposed framework and determined the appropriate combination of oversampling and segmentation strategies with the proposed networks. However, several parameters in the study were controlled; (1) there was little environmental noise, (2) the audio samples were recorded by a smart phone nearby with almost no artificial or ambient disturbance during the processes, (3) the start and end points of the collection were carefully selected to ensure high quality recordings, and (4) there were almost no overlaps and co-occurrence among the activities. To generalize the study in more natural settings, we conducted an IRB-approved *in-the-wild* study with 14 human subjects in their actual home environments (7 males and 7 females, with the age between 20 to 30).

The *in-the-wild* study was performed by following a scripted scenario. A key advantage of this approach is that the process of following the script can simulate the continuous flow of human activities just as in natural home settings. All target activities were listed in advance in the form of instructions such as "first head to the bathroom, wash your hands and face" or "after juice is prepared, please warm some food using the microwave oven". Each subject followed the instructions written on a sheet of paper and freely performed the activities. We adopted the same off-the-shelf smartphone device (i.e., Huawei P9) for data collection. The smartphone was carried in the subjects' arms with a wristband so that the participants could perform the activities without needing to attend to the device. During data collection, an experimenter (i.e., one of the authors of the paper) followed the subjects at a distance while they were performing the activities (e.g., waited outside the room while the subject was cleaning); the experimenter was available to answer questions during the study and, critically, to label and time the target activities.

To incorporate variability factors in the tests, the experimenter occasionally chatted with participants during some of the activities such as watching TV, cooking or strolling. To simulate multi-tasking, subjects were allowed to perform some activities simultaneously such as washing and frying. Moreover, participants were encouraged to use their own devices or tools (e.g. their own vacuum cleaner, kitchen and toilet appliances) during data collection.

In our script, most activities were required to be performed only once and the length was determined freely by the participants. The experimenter setup bacon, cucumbers or carrots in advance for activities 'frying food', 'chopping food' and 'squeezing juice'. For the class 'watching TV', participants were asked to watch 5 different channels for about 30 seconds each. For the 'enjoying music' activity, the subjects were asked to play their own piano or listen to relevant types of musics such as piano solo or symphonies chosen by themselves. Finally, female participants were not asked to perform the 'shaving' activity.

5.1 Results and Discussion

In total we obtained 32105 seconds (535 minutes) of audio data. Based on the labeled time stamps, we manually segmented the target activity data from the raw recordings. Overall, we identified that roughly 12078 seconds (201 minutes) of the clips were target activity-related, accounting for 37.6% of the total. The resulting sparsity is comparable to audio-based activity recognition in practice as not all home-related activities can generate specific sound features and audio-based frameworks are not suitable for them. We applied the best architecture

of the proposed framework (classification network with random oversampling) for the following evaluation. The segmentation length was set as 10 embedding vectors (9.6 seconds).

As a baseline, we first evaluated the proposed architecture by training a model with the aggregate of all the data collected in the *in-the-wild* study. We obtained this measure by performing 3-fold cross validation. In our compiled dataset, class 'TV' accounts for the most (2242 seconds) and class 'flushing' accounts for the least (166 seconds). By randomly oversampling the user data to equalize the number of samples across all user activities, we were able to obtain a new user dataset of 33630 seconds in size. To ensure the same input format for the classification network, we applied the VGGish extractor for embedding transform. For each validation run, 2/3 of the data was used to train the classification network while the remaining 1/3 was used for evaluation. Our framework yielded 85.55% averaged classification accuracy with 0.72% standard deviation.

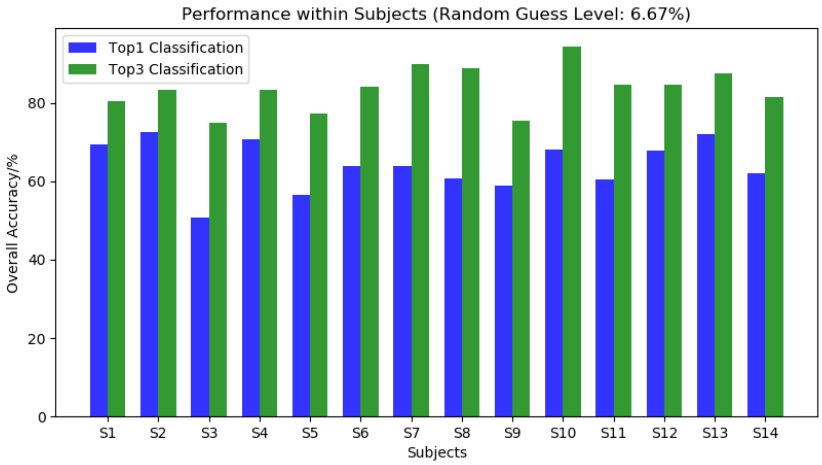

Fig. 5. The averaged top-1 and top-3 accuracies are 64.16% and 83.59% respectively for all subjects.

Next, we studied how the proposed framework trained on the AudioSet embeddings performed on the user test data (12078 seconds). Test results were first evaluated based on each individual participant. Figure 5 shows the overall classification performance for each subject. Because of the high inequality of segment length among the activities, we adopted the overall weighted average as the performance metric. In other words, for a given subject, the contribution of each tested instance to the overall accuracy is inversely proportional to the amount of tested data within that corresponding activity class. By weighting the instances, each activity class within the subject can then contribute equally to the overall performance. In our studies, the averaged top-1 classification accuracy was 64.16% for all tested subjects. In addition to the top-1 classification, we also evaluated the overall performance using a top-3 classification scenario given the co-occurrence of activities and the variability during the tests. In the top-3 classification, predicted labels with the top 3 highest confidence are considered as the final predictions, and a true positive can be counted if any of the 3 labels match the ground truth. It incorporates the variants of predictions due to possible similarity of sound features or concurrence of the actual activities. From the figure we can see that the top-3 performance was much better than the top-1 scenario, with an averaged accuracy of 83.59% for all 14 subjects.

To evaluate the performance of individual activity classes, we also summarized the class accuracies across all tested subjects. We calculated the average values for both the top-1 and top-3 classification, and Figure 6 and Figure 7 present the statistics for both settings. Instead of directly applying confusion matrices, we adopted a similar

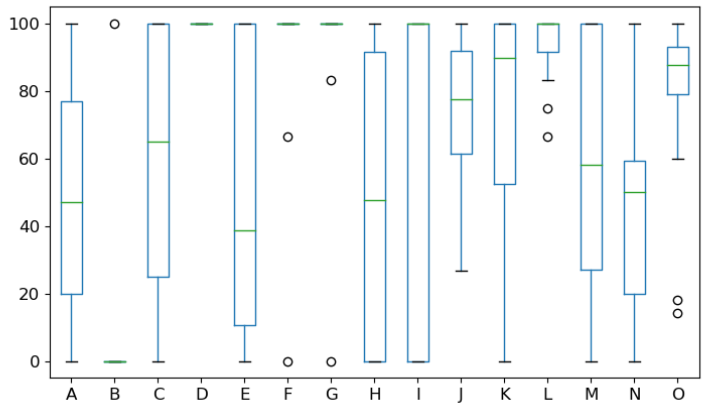

Fig. 6. Top-1 classification accuracy for individual activity classes (A:Bathing/Showering; B:Flushing; C:Brushing Teeth; D:Doing Shaver; E:Frying; F:Chopping; G:Microwave Oven; H:Boiling; I:Squeezing Juice; J:Watching TV; K:Playing Music; L:Floor Cleaning; M:Washing; N:Chatting; O:Strolling)

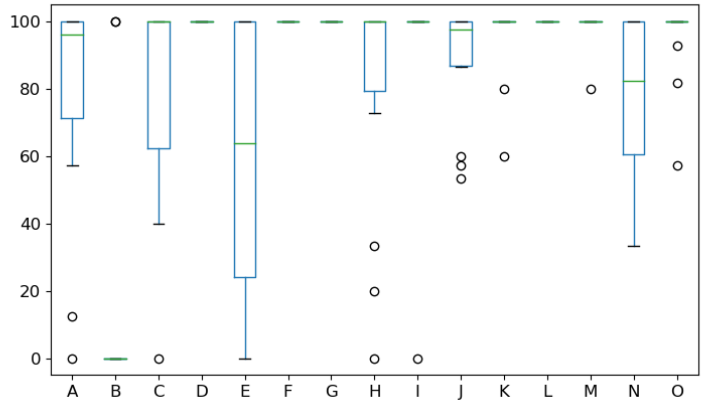

Fig. 7. Top-3 classification accuracy for individual activity classes (A:Bathing/Showering; B:Flushing; C:Brushing Teeth; D:Doing Shaver; E:Frying; F:Chopping; G:Microwave Oven; H:Boiling; I:Squeezing Juice; J:Watching TV; K:Playing Music; L:Floor Cleaning; M:Washing; N:Chatting; O:Strolling)

weighted approach for the analysis. That is, tested instances from each subject were assigned with weight that was inversely proportional to the amount of data within them. This enables samples from different subjects and different tested environments with varying data size to contribute equally to the overall performance of the target classes. In addition, the figures also indicate the deviations of the class accuracies away from the mean. A smaller deviation represents a more stable performance of the predictions and further implies a stronger robustness of the framework towards variants in the actual tests. As indicated from the figures, 'shaving', 'chopping food' and

'using microwave oven' showed the best performance with almost 100% averaged class accuracy and almost zero deviation. Class 'floor cleaning' was also of satisfactory results due to its clear and unique sound features. On the contrary, however, most of the flushing activities were misclassified by the framework. We hypothesize this occurred because the process of flushing was too short given the segmentation length. Also, the sounds of water flushing can largely overlap with those of the washing or frying activities. Also in the figures, some activities such as 'frying food', 'boiling water', 'squeezing juice' and 'brushing teeth' deviate highly from the average. This is reasonable because the modalities of cooking and boiling can vary in practice depending on the choice of the cooking tools and cooking styles among the participants. The performance of kitchen activities was also affected by usage of hoods by some of the participants. The brushing activity could mainly be affected by the noise of toilet fans. Especially, we noticed that our framework failed to recognize almost all brushing activities with electric toothbrush possibly due to the lack of relevant training samples in the AudioSet. If comparing the results in both figures, we can also see that the performance of most activities increased significantly from the top-1 scenarios to the top-3 scenarios, reaching nearly 100% mean accuracy with much smaller deviations. This implies the existence of activity co-occurrence and overlaps of acoustic features among distinct activities such as simultaneous chatting with outdoor strolling or a music show on TV, which are also commonly seen in the natural home settings.

In multi-class activity recognition, a common formulation is to characterize several target classes of interest plus a so-called NULL class, which represents the part of the signal where none of the target classes are observed. The NULL class is typically difficult to recognize because it is composed of all activities deemed not interesting. In our analysis, we evaluated the model performance for the proposed 15 classes but did not train a proper NULL class. A question that naturally arises in this case is how the classifier behaves when provided with a non-target class, i.e., garbage input. To answer this question, we randomly sampled around 20% (4429 seconds) of non-target audio and labeled it as belonging to a NULL class. The non-target audio was obtained from participants as they transitioned from one activity to another in the study. We found that the NULL instances were most likely predicted to belong to class 'chopping', with averaged confidence of 20.27%. A possible explanation is that the sound of chopping was similar to the sound of footsteps; study participants were frequently walking inside the home when transitioning from one activity to another. More importantly, we observed that the predicted distributions of confidence levels for the non-NULL classes, the 15 target activities, had an averaged true confidence level *over* 20%. In other words, 20% confidence in a prediction can be seen as a threshold separating a target class from a NULL class. This finding suggests that we might be able to successfully discriminate activities we are interested against those we are not, even though our model was not specifically trained with a NULL class.

Because of the difference in terms of evaluation metrics and test conditions, it was challenging to directly compare the performance of our framework against related work. As reference, Rossi et al. [31] combined a semi-supervised or manual filtering of outliers with the Gaussian Mixture Model (GMM) to classify 23 acoustic contexts. They extracted the MFCC features from the Freesound dataset with the sequence length of 30 seconds for training. The best top-1 classification and top-3 classification performance were 57% and 80% respectively only if with manual filtering of the outliers. Hershey et al. [14] trained two fully connected networks with and without the embedding extraction process to classify the AudioSet [12] categories. They adopted the mean Average Precision (mAP) as the performance metric and obtained the best mAP of 0.31 only if taking the embeddings as input. Kong et al. [17] completed a similar test using an attention model from a probability perspective, achieving mAP of 0.327 and AUC of 0.965. The state of the art by Laput at al. [20] reported the classification performance from several perspectives. Their best model achieved 80.4% overall accuracy for 30 context classes recorded in the wild, but the framework relied on a mixed process of audio augmentation and combination of sound effect libraries for training. When using only online video sounds (i.e. the AudioSet [12] data), their framework yielded the best overall accuracy of 69.5% when check-pointed on the test set and 41.7% when tested directly with real-world sounds. Correspondingly, our framework was not developed based on any feature augmentation and

semi-supervised learning processes. The overall classification accuracy of our model was 81.1% for 15 activity classes in the lab study. Our top-1 and top-3 performance was 64.2% and 83.6% respectively based on multi-subject tests of 14 participants in their actual home environments.

5.2 Privacy Concerns

A key issue in sound-based activities of daily living recognition is privacy. As pointed out by Christin et al. [7], audio-based computing frameworks can lead to privacy threats due to the recording of confidential conversations or sound patterns that uniquely reveal activities and locations. Hence, dedicated efforts are required for the preservation of user privacy. At a minimum, the raw audio data clips should be deleted from the mobile device and processing pipeline right after feature extraction so that sensitive information does not persist within the framework [25]. An approach that should be considered is the extraction and utilization of audio features that cannot be used to reconstruct the original recorded audio [38]. Also, extracted audio frames can be randomly split or aggregated in the form of statistics on the server [11]. Although our current audio framework does not currently implement these audio-based privacy protecting measures, this is an area we plan to explore in future work.

6 CONCLUSION

The collection and annotation of ground truth user data is often time-consuming and laborious in the field activity recognition. This paper presents a novel audio-based framework that uses large-scale on-line YouTube video soundtracks to train activity recognition models. Our framework combines transfer learning, oversampling and a deep learning architecture without the need for feature augmentation or semi-supervised methods. To evaluate the performance of this framework, we conducted pilot and *in-the-wild* studies showing that our proposed framework can recognize 15 common home-related activities with promising performance and robustness to environmental variability.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Alvina Anjum and Muhammad Usman Ilyas. 2013. Activity recognition using smartphone sensors. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*. IEEE, 914–919.

[3] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. 2016. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*. 892–900.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Jianfeng Chen, Alvin Harvey Kam, Jianmin Zhang, Ning Liu, and Louis Shue. 2005. Bathroom activity monitoring based on sound. In *International Conference on Pervasive Computing*. Springer, 47–61.

[6] François Chollet et al. 2015. Keras. https://keras.io. (2015).

[7] Delphine Christin, Andreas Reinhardt, Salil S Kanhere, and Matthias Hollick. 2011. A survey on privacy in mobile participatory sensing applications. *Journal of systems and software* 84, 11 (2011), 1928–1946.

[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] Ethan Fast, William McGrath, Pranav Rajpurkar, and Michael S Bernstein. 2016. Augur: Mining human behaviors from fiction to power interactive systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 237–247.

[10] Frederic Font, Gerard Roma, and Xavier Serra. 2013. Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 411–412.

[11] Raghu K Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F Abdelzaher. 2010. GreenGPS: a participatory sensing fuel-efficient maps application. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 151–164.

[12] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 776–780.

[13] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*. Springer, 878–887.

[14] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 131–135.

[15] Derek Hao Hu, Vincent Wenchen Zheng, and Qiang Yang. 2011. Cross-domain activity recognition via transfer learning. *Pervasive and Mobile Computing* 7, 3 (2011), 344–358.

[16] Kyuwoong Hwang and Soo-Young Lee. 2012. Environmental audio scene and activity recognition through mobile-based crowdsourcing. *IEEE Transactions on Consumer Electronics* 58, 2 (2012).

[17] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D Plumbley. 2017. Audio Set classification with attention model: A probabilistic perspective. *arXiv preprint arXiv:1711.00927* (2017).

[18] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. 2011. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* 12, 2 (2011), 74–82.

[19] Nicholas D Lane, Petko Georgiev, and Lorena Qendro. 2015. DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 283–294.

[20] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. In *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 213–224.

[21] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic sensors: Towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3986–3999.

[22] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5. http://jmlr.org/papers/v18/16-365.html

[23] Alexander Liu, Joydeep Ghosh, and Cheryl E Martin. 2007. Generative Oversampling for Mining Imbalanced Datasets.. In *DMIN*. 66–72.

[24] Hong Lu, Wei Pan, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. 2009. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 165–178.

[25] Emiliano Miluzzo, Nicholas D Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B Eisenman, Xiao Zheng, and Andrew T Campbell. 2008. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 337–350.

[26] Long-Van Nguyen-Dinh, Ulf Blanke, and Gerhard Tröster. 2013. Towards scalable activity recognition: Adapting zero-effort crowdsourced acoustic models. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 18.

[27] Long-Van Nguyen-Dinh, Mirco Rossi, Ulf Blanke, and Gerhard Tröster. 2013. Combining crowd-generated media and personal data: semi-supervised learning for context recognition. In *Proceedings of the 1st ACM international workshop on Personal data meets distributed multimedia*. ACM, 35–38.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[29] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. 2005. Activity recognition from accelerometer data. In *Aaai*, Vol. 5. 1541–1546.

[30] Mirco Rossi, Sebastian Feese, Oliver Amft, Nils Braune, Sandro Martis, and Gerhard Tröster. 2013. AmbientSense: A real-time ambient sound recognition system for smartphones. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*. IEEE, 230–235.

[31] Mirco Rossi, Gerhard Troster, and Oliver Amft. 2012. Recognizing daily life context using web-collected audio data. In *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 25–28.

[32] Sebastian Säger, Benjamin Elizalde, Damian Borth, Christian Schulze, Bhiksha Raj, and Ian Lane. 2018. AudioPairBank: towards a large-scale tag-pair-based audio content analysis. *EURASIP Journal on Audio, Speech, and Music Processing* 2018, 1 (2018), 12.

[33] Justin Salamon and Juan Pablo Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters* 24, 3 (2017), 279–283.

[34] 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.
[35] Muhammad Shoaib, Stephan Bosch, Hans Scholten, Paul JM Havinga, and Ozlem Durmaz Incel. 2015. Towards detection of bad habits by fusing smartphone and smartwatch sensors. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE, 591–596.
[36] Edison Thomaz, Irfan Essa, and Gregory D Abowd. 2015. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1029–1040.
[37] Edison Thomaz, Cheng Zhang, Irfan Essa, and Gregory D Abowd. 2015. Inferring meal eating activities in real world settings from ambient sounds: A feasibility study. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 427–431.
[38] Danny Wyatt, Tanzeem Choudhury, and Jeff Bilmes. 2007. Conversation detection and speaker segmentation in privacy-sensitive situated speech data. In *Eighth Annual Conference of the International Speech Communication Association*.
[39] Hanghang Tong Xing Su and Ping Ji. 2014. Activity Recognition with Smartphone Sensors. *Tsinghua Science and Technology* 19, 3 (2014), 235–249.
[40] Koji Yatani and Khai N Truong. 2012. BodyScope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 341–350.

Received November 2018; accepted January 2019