SoundSemantics: Exploiting Semantic Knowledge in Text for Embedded Acoustic Event Classification

Md Tamzeed Islam
UNC Chapel Hill
tamzeed@cs.unc.edu

Shahriar Nirjon
UNC Chapel Hill
nirjon@cs.unc.edu

ABSTRACT

In this paper, we propose a fundamentally different approach to acoustic event classification that exploits knowledge from the textual domain to deal with a well-known pain point in audio event classification—i.e., the lack of adequate training examples. We show that by exploiting existing context-aware semantic representation of English words (e.g., Google’s word-to-vector [33]) that is generated from a massive amount of English texts on the Internet, it is possible to classify acoustic events even when there are a few or no training examples for a wide variety of sounds. Our approach works in application scenarios where a system wants to learn a number of predefined categories of acoustic events, but it does not have enough training examples per category, and/or for some categories, it does not have any training examples at all. We solve this problem by combining a robust audio representation step, followed by a cross-modal projection of the audio representation onto textual representation. Our approach is different from techniques such as *one-shot* and *data augmentation* that do not consider the cross-domain knowledge transfers. We develop a generic mobile application for audio event detection where a user can input a list of desired sound types, along with training audio clips for some of those sound types, and the system is able to recognize all types of sounds (at varying level of accuracy depending on the number of classes that do not have any training examples), which is not achievable by any existing audio classifier that we are aware of. We evaluate the performance of the proposed system on an empirical dataset [41] as well as by deploying the application in two real-world scenarios. The accuracy of the classifier lies between 60%-90% for a 6–10 class problem when the number of classes that do not have any training examples is varied between 2–5.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Computer systems organization** → **Embedded and cyber-physical systems**;

KEYWORDS

Zero-shot Learning, Audio Classification

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IPSN '19, April 16–18, 2019, Montreal, QC, Canada

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6284-9/19/04...\$15.00

<https://doi.org/10.1145/3302506.3310402>

1 INTRODUCTION

Recent advancements in machine learning [11, 20] have initiated a revolution in embedded sensing and inference systems. Acoustic sensing and inference systems have enabled a wide variety of applications such as speech recognition [28], speaker identification [32], music recognition [53], sleep quality monitoring [5], intrusion detection [6], elderly monitoring [50], activity recognition [4], and health vitals monitoring [38, 42]. Existing machine learning algorithms perform remarkably well in these applications, given that they enjoy the availability of labeled training data for all types of acoustic events that are involved in the application [37]. However, with the ubiquity of acoustic sensing systems, the diversity in their sound types as well as their intended purposes are evolving—making it hard for us to acquire a large corpus of labeled training data for all possible types of acoustic events that an application wants to detect. For example, in applications like sleep monitoring, home intrusion detection, and health monitoring, we may want to collect multiple instances of rare, inconvenient, and unwanted acoustic events such as snoring, breaking in, sneezing, and wheezing. Such requirements create challenges to developing personalized and environment-specific acoustic event recognition systems which primarily rely on user-contributed labeled data for training the classifier. Hence, the research community for long has sought after a solution that can classify audio using few or no training examples.

Audio classification has been a well-studied problem for decades. Dealing with the lack of training data has remained an open and active research problem since then. Recent work [9, 18, 43] in the audio recognition literature that address the problem of *learning from limited training data* mainly apply *data augmentation* [48] and *classifier fusion* [27] techniques. However, these techniques require a certain amount of representative training data to begin with. Furthermore, all existing audio recognition algorithms limit themselves to a fixed set of sound types, and thus when an audio clip from a completely new sound type is presented to them, they have no built-in mechanism to guess the correct label for that unknown sound. For instance, if a classifier learns only two classes {microwave oven, vacuum cleaner} and is tested with a sound of a {blender}, it will never be able to output anything other than {microwave oven, vacuum cleaner} or "sorry".

In this paper, we study the problem of *learning without examples* in the context of audio inference. The problem is popularly known as the *zero-shot learning* [10, 29], which is an active area of research in computer vision and image classification. To the best of our knowledge, we are the first to apply zero-shot learning to solve the general-purpose audio event classification problem. The central idea of zero-shot learning is to exploit information or learned knowledge from other sources such as textual descriptions.

For example, to describe a *tiger* to a person who knows *cats*, instead of showing them a tiger, we can describe that ‘a tiger is just like a cat but 20 times bigger.’ In this work, we exploit the context-aware representation of English words as an additional source of knowledge when classifying audio clips. Through an advanced audio to text domain projection algorithm, we blur the difference between an audio and its corresponding English word/phrase by representing them in the exact same vector space. We exploit the semantic relationship between English words to classify an audio clip that the system has never heard before. We combine these algorithmic steps to develop an advanced acoustic event classification system that we name—SoundSemantics.

Developing a system like SoundSemantics that projects audio representation onto the space of text representation is non trivial and poses several challenges that are addressed in this paper. First, we propose a new way to learn robust audio features by explicitly learning the similarity (and dissimilarity) between different sound types. We show that such a representation is robust, beneficial to later steps, and yields better features for acoustic data. Second, we propose a neural network architecture to merge text and audio domain so that acoustic data is embedded with the distributional characteristics of English words. This results in the first context-aware audio embedding and paves a way for classification without training data for audio. Third, we propose a two-way classifier that is capable of classifying audio clips from previously *heard* as well as *unheard* sound types. This makes SoundSemantics a generalized system for classifying all types of acoustic events.

To demonstrate the performance of the proposed algorithms in real-world scenarios, we develop a smartphone application that implements the end-to-end pipeline of SoundSemantics. A user of the application simply inputs a list of sound types along with zero or more training examples for each type, and the system is capable of learning and accurately classifying all sound types in real-time. We deploy the SoundSemantics app in two real-world scenarios (kitchen and street sound recognition) and show that the system is able to detect acoustic events with or without training samples with an accuracy of up to 90%. We also evaluate SoundSemantics’ performance on a popular empirical dataset and report the performance of the proposed algorithms.

The contributions of this paper are the following:

- We describe SoundSemantics, which is the first system that addresses the problem of learning without examples for general-purpose acoustic event classification.
- We propose a new approach to learn robust audio representation to preserve better semantic relationship between similar sounds. This representation by itself can be used in developing robust acoustic event classifiers when we have training data for all the classes.
- We propose the first algorithm that exploits semantic knowledge in text to enable classification of audio clips of sound types for which a classifier has never been trained for.
- We develop a light-weight mobile application that executes the audio classification algorithms in real-time and conduct empirical and real world evaluations to test the algorithms as well as the end-to-end performance of the mobile application.

2 CONCEPT APPLICATIONS

SoundSemantics is an advanced acoustic event recognition system that beats the state-of-the-art in scenarios where we want to recognize a wide variety of sounds, but collecting enough audio clips for each type of sounds is not feasible. In the machine learning literature, this is known as the *class imbalance* problem [19], where the number of training examples varies significantly from one class to another. Especially, in rare event detection problems, such as anomaly detection, fraud detection, and medical diagnosis, often we do not have enough training examples of the primary event of interest. This results in classifiers that suffer from high false negative rates and misclassify critical events. SoundSemantics solves this problem in a unique way, and accurately classifies acoustic events even if some of the sound types have a few or no training examples. This capability of SoundSemantics enables many indoor and outdoor applications that were not possible before.

2.1 Indoor Usage Scenarios

There are many indoor audio sensing and recognition scenarios such as sleep monitoring [5], bathroom activity monitoring [4], home appliance monitoring [12], asthma severity detection [42], music genre recognition [51], and context recognition for voice assistants [17], where using SoundSemantics can dramatically improve the performance as well as the intelligence of an application. Because each indoor acoustic environment is different, these systems typically rely on user-contributed audio clips to train the classifiers. However, collecting a large number of audio clips for each type of sound is an inconvenience to an end-user. In such cases, SoundSemantics is able to recognize sounds even if a user provides no sample audio for some of the sound types.

Figure 1: Kitchen activity monitoring application.

For instance, consider the simplified *kitchen activity monitoring* scenario in Figure 1 where a user wants SoundSemantics to monitor four types of kitchen appliances or activities such as {microwave oven, vacuum cleaner, dish washer, blender}. The user provides SoundSemantics the names of these four types of sounds. SoundSemantics asks the user to record one or more audio clips for each sound type. However, the user only records and provides audio clips of two of these types, i.e., {microwave oven, vacuum cleaner} and does not provide any audio clip for the other two, i.e., {washer, blender} as collecting dish washer and blender’s sound requires more work from our busy user. Despite the lack of sample audio clips, SoundSemantics is able to recognize all four types of sounds

at runtime, including the blender—for which SoundSemantics was not provided any audio samples during the training phase.

2.2 Outdoor Usage Scenarios

Like indoors, SoundSemantics has the potential to make outdoor sound recognition applications far more effective than they are today. There are many outdoor acoustic event recognition applications, such as street activity monitoring [3], violence detection [8], and pedestrian safety applications [7], where the number of sound types involved in the application is large, and the quality of the sound varies due to distance, background noise, and acoustic environment. It is not feasible to collect audio samples of all types of sounds under all possible combinations of distance, background noise, and acoustic context. The use of SoundSemantics simplifies the task of training acoustic event classifiers as it only requires a list of possible sound types and example audio clips of some. For instance, in a *street monitoring* application, it is relatively easier for a user to list a possible set of sounds that he/she may encounter (e.g., car engine and tire noise, honks, sounds of buses and trucks, human chatter and footsteps) than to actually record audio clips of all these types of sounds. Collecting training data for this application is not only cumbersome, but it may also raise privacy concerns as audio recording in public places is, in general, illegal. Using SoundSemantics, a user can record as many audio clips of as many types of sounds as possible, and the system takes care of learning the rest of the sound types automatically.

3 OVERVIEW OF SOUNDSEMANTICS

SoundSemantics is an advanced acoustic event recognition system that takes a short-duration audio clip and a list of possible sound types (i.e., *tag-list*) as input and processes the audio through an acoustic processing pipeline to classify the audio as one of those given sound types. The design of SoundSemantics is modular. Computationally expensive modules (e.g., one-time offline training of the classifiers) are run on a server, while the end-to-end audio classification pipeline – from sensing to classification – runs on embedded systems such as smartphones, tablets, and voice assistant devices like Amazon Echo and Google Home.

Figure 2: SoundSemantics acoustic processing pipeline.

Figure 2 shows the pipeline diagram containing the three processing steps of SoundSemantics: *Robust Audio Representation*, *Cross-Modal Projection*, and *Two-Level Classifier*. This section provides an overview of each of these steps and defers their algorithmic details to subsequent sections.

3.1 Robust Audio Representation

Audio signals from microphones are essentially a time series of numbers. Typically, the sampling rate of a microphone lies between 8 KHz to 48 KHz. This results in 8,000 to 48,000 dimension vectors every second, which is not practical for direct use in training or classification of audio data—especially in CPU and memory constrained

embedded systems. Hence, the first step in any audio classification system is to map raw audio data to a lower dimensional vector, commonly known as the *feature* vector, based on its time and frequency domain properties. A wide variety of audio features have been proposed in the literature. The list includes classic time and frequency domain features such as zero crossing rate, pitch, cepstral coefficients, spectral entropy, energy, flux, and roll-off [37]; as well as recent practices where a layer of a pre-trained neural network is considered as the learned features [30]. A limitation of all these features is that although they can distinguish examples from different classes, they do not preserve the inter-class relationship.

Figure 3: Preserving semantics in audio representation.

Figure 3 illustrates this concept using a simple three-class problem where two 2D feature representations of examples from the classes {blender, male voice, female voice} are shown. Both representations are able to distinguish examples from different classes. However, the left one does not consider the fact that male voice and female voice are semantically similar, and thus, they should be closer to each other. The representation on the right is a better representation in this regard as it ensures class separability and preserves inter-class semantic relationship.

In SoundSemantics, we employ a special type of neural network, namely the *siamese network* [23], to learn a robust audio representation that positions similar types of audio signals closer to each other and vis-à-vis. Such a representation helps SoundSemantics in the next step down the audio processing pipeline where we leverage semantic knowledge from English text to enable classification of unknown sounds. The details of robust audio representation is described in Section 4.

3.2 Cross Modal Projection

(a) Existing Algorithms are unable to Project Audio into Word Embedding Space, $Y \neq X$
(b) SoundSemantics Projects Audio into Word Embedding Space, $Y = X + \epsilon$

Figure 4: Cross-modal Projection maps audio to its corresponding label in word embedding space.

The secret recipe behind SoundSemantics' ability to classify completely unknown sounds is the *cross modal projection* step. Through this step, we blur the difference between an audio clip and its

corresponding English word, and make them (almost) equal in their feature representation. In other words, if an English word (e.g., the word ‘blender’) has a known vector representation (e.g., obtained using Google’s word-2-vec [33] on a large dataset like Wikipedia [16]), the goal of this step is to generate the exact same vector when SoundSemantics is presented with an audio clip of that word (e.g., the sound of a blender). The benefit of this *projection* from audio to text representation is that there are over 150 thousand English words for which researchers in the natural language processing field have created semantically aware vector representations, called the *word embedding*. Such an embedding preserves the contextual relationship among words and puts two words that are similar in meaning or are often used in the same context closer in the representation space. By projecting the audio representation to the word embedding space, SoundSemantics is able to generate meaningful and context-aware representation of any audio clip – irrespective of whether or not it has heard example audio clips of the same class before.

In Figure 4, we illustrate cross-modal projection using the example of a *blender*, whose word embedding is X . The audio representation of blender is passed through the cross-modal projection to obtain Y . In SoundSemantics, we make sure that $X \approx Y$. In traditional acoustic feature representation, however, the projection is different ($X \neq Y$), and hence unlike SoundSemantics, it’s not possible for traditional features to guess the word label of an unheard sound.

3.3 Two-Level Classifier

There are two classes of sound types that SoundSemantics may encounter at runtime: *heard* and *unheard* classes. The *heard* classes refer to those sound types for which SoundSemantics has audio clips for training. The *unheard* class, on the other hand, refers to sound types for which SoundSemantics does not have any training audio clip. For example, in the kitchen monitoring scenario of Figure 1, {microwave, vacuum} are heard classes and {blender, washer} are unheard classes. During the classification step, at first, SoundSemantics employs a *novelty detection* algorithm to determine whether the input audio belongs to a heard class or an unheard class. After this determination, a *nearest neighbor* classifier is used to find the most probable sound type for the input audio. The use of a two-level classifier significantly improves the accuracy of SoundSemantics, because we propose a holistic approach to classify both *heard* and *unheard* classes together. The details of this two-level classifier is described in Section 6.

4 ROBUST AUDIO REPRESENTATION

The goal of this step is to obtain a robust feature representation of audio which explicitly learns the similarity and dissimilarity of examples from different classes.

4.1 Basic Approach

We employ a *Siamese* neural network [23] that takes two audio clips as the input and determines whether or not they are similar. A siamese network is a twin neural network architecture where two identical deep neural networks share their weights. The objective of this network is to learn a representation of the training data which projects similar data points closer while ensuring a certain margin

between data points from different classes. This is achieved by using an *loss function* during the training phase, which is discussed later in this section.

We train the network by providing it with pairs of audio clips as the input and a binary target that denotes whether or not these clips belong to the same class as the output. Once the training finishes, we take the output vector of one of the last few layers—which becomes the desired robust, inter-class similarity aware representation of an input audio clip.

A siamese network has advantages over other types of commonly used neural networks such as the convolutional neural networks (CNNs) as it learns the semantic representation better by explicitly learning a similarity metric. Furthermore, siamese networks are less susceptible to *class imbalance* [23, 46] problem, which is one of the main challenges that we address in this paper.

4.2 Analytical Formulation

We denote $G_\theta(X_i)$ as the projection of an audio clip by one of the twin networks G_θ , where X_i is a standard frequency domain representation (e.g., FFT or MFCC [37]) and θ denotes the learnable weights of the network. For two audio clips, X_1 and X_2 , we set their distance score, $Y = 0$ when they belong to same class, and $Y = 1$ when they belong to different classes. Finally, we define a contrastive [13] loss function as follows:

$$L(\theta) = (1 - Y) \frac{1}{2} \left| G_\theta(X_1) - G_\theta(X_2) \right| + Y \frac{1}{2} \left\{ \max \left(0, \Delta - \left| G_\theta(X_1) - G_\theta(X_2) \right| \right) \right\} \quad (1)$$

The first term in Equation 1 minimizes the distance between a pair of audio from same class whereas the second term maximizes the distance between a pair of audio from different classes. The term Δ represents the distance margin that the network maintains between audio clips from different classes.

4.3 Neural Network Architecture

As the architecture of G_θ , we use four layers of convolutional layers with max pooling and batch normalization. Each layer extracts feature map from the input data and as we go deeper into network more number of filters extract more information. We decided to go with four layers as we found that these layers were able to get adequate information for the later steps. For nonlinear activation, we use *relu* activation function. The output of the fourth convolutional layer is fed into two layers of dense layers which have *relu* activation in between them. We keep the last dense layer with 1000 neurons which is our feature dimension.

In Figure 5, we start with a filter size of 20×20 for the first convolutional layer. We keep the size slightly bigger to capture more information from both the time and frequency domain. We have experimented with larger sized filters and find that they result in slower convergence. We use pooling layers along with the conv layers with a stride of 2 and a kernel size of 2×2 . This is a common choice for pooling layer to sub-sample from layers to reduce the size as neighbor nodes are usually highly correlated. We also find that batch normalization boosts the accuracy as it handles different scaling factors of the input.

Figure 5: The architecture of the Siamese network used in robust audio representation. Here, f denotes the size of the filters and s denotes the stride length of the convolutions. Two audio clips are given as input and the last layer is fed into the contrastive function to learn the similarity between them.

Using the trained siamese network, we obtain the robust audio representation of the audio sample by forward propagation through any of the twin networks. Since the networks have identical weights, choosing any of them yields the same representation.

5 CROSS MODAL PROJECTION

In the previous section, we proposed a feature learning algorithm that learns to map audio data onto a feature space that projects similar data close and dissimilar data far. This gives us a robust audio representation which by itself can directly be used to train robust audio classifiers. However, that representation can not be used to perform zero-shot learning as it does not borrow knowledge from an external source. To achieve this, we propose an algorithm that integrates semantic knowledge from the text domain to the audio representation. This is done by projecting the audio representation onto the word embedding space such that each audio representation is mapped to the word embedding of its corresponding text.

The difficulty in this step, however, is that a word is spelled or written in exactly one way, whereas, its corresponding audio clip may have a lot of variations. In SoundSemantics, we solve this problem using a neural network to learn the projection from audio to text.

5.1 Neural Network Architecture

We use a two-layer fully connected neural network for cross-modal projection between the two data modality. The first layer has 500 neurons which are connected to the input layer with a dimension of 1000. For the first layer, we apply $\tanh()$ to model the non-linearity in input data. The output of the first layer is then fed into the last layer which outputs the projected embedding of the audio onto the word space. The number of neurons in the output layer is the same as the dimension of the word embedding we use. In this case, it is set to 50 to match with the dimensions of word embedding of [16].

For an audio sample, A and its robust representation, A_{feat} , the final output of the neural network is defined by the following equation:

$$A_{out} = \theta^2 g(\theta^1 A_{feat}) \quad (2)$$

Here, θ^l denotes the layer of neurons and l denotes the number of layers. If A_{feat} has $M = 1000$ dimensions, the number of neurons in the first layer is h^1 , and the number of dimensions of word embedding is d , then the layers have the following dimensions: $\theta^1 \in \mathbb{R}^{h^1 \times M}$ and $\theta^2 \in \mathbb{R}^{d \times h^1}$. Note that, g denotes the non-linear function $\tanh()$.

5.2 Loss Function

The target of cross-modal network is to project audio as close as possible to its corresponding label's word embedding. Therefore, when an audio sample's robust representation passes through the neural network, the target is to minimize the euclidean distance between the output of the neural network and the audio's class label's word embedding.

- *Mean Square Error (MSE) Loss*: Mean Square Error (MSE) [31] is a good choice of loss function for our network as we can use it to minimize the distance between projected audio embedding and its label. Let's represent the pre-trained word embedding of a word W as $E(W)$. For an audio sample A , and its class label y , following the MSE loss function we want to minimize:

$$L(\Theta) = \|E(y) - A_{out}\| \quad (3)$$

Here, $\|\cdot\|$ is the Euclidean norm and $\Theta = \{\theta^1, \theta^2\}$. We have omitted the division with vector size here for easy notation. One drawback of MSE loss is it does not consider its neighbour word labels when minimizing the distance between the audio projection and word embedding. This may lead to poor separability of embeddings between two audio classes which have close class label word embedding. In Figure 6(a), we demonstrate this fact by projecting audio of *blender* and *washer* in word space. We see from the figure that, MSE loss minimized the distance between the corresponding projection of audio and their class labels. But MSE loss failed to get a fair separation between the projected embedding of *blender* and *washer* audio clips.

- *Neighbour-Aware Loss*: To consider nearby labels, we propose a new loss function to project audio into word embedding space. The proposed loss function has two objectives. First, it minimizes the distance between the projected audio embedding and its corresponding audio class label. Second, it ensures that the distance

Figure 6: (a) MSE Loss fails to separate the cross-modal embedding of adjacent audio class *blender* and *washer* in word embedding space. (b) Neighbour-aware Loss projects the audio clips close to their corresponding labels in word embedding space and also keeps separation from adjacent classes.

between the projected audio embedding and its corresponding audio class label is less than the distance between the projected audio embedding and other class labels by a certain margin.

The second objective makes sure that the projected audio embedding of a class y is closer to $E(y)$ by a margin of α than other class labels $y' \neq y$. To formally define, we want the following condition to hold after projection:

$$\|E(y) - A_{out}\| + \alpha \leq \|E(y') - A_{out}\|, y' \neq y \quad (4)$$

Hence, the objective function becomes:

$$\begin{aligned} &\underset{\Theta}{\text{minimize}} \quad \|E(y) - A_{out}\| \\ &\text{subject to} \quad \|E(y) - A_{out}\| + \alpha \leq \|E(y') - A_{out}\|, y' \neq y \end{aligned}$$

Using this optimization function, we propose our *Neighbour-aware Loss* function as following:

$$\begin{aligned} L(\Theta) = & \|E(y) - A_{out}\| + \frac{1}{|adj|} \times \sum_{y' \in adj(y)} (\|E(y) - A_{out}\| \\ & + \alpha - \|E(y') - A_{out}\|) \end{aligned} \quad (5)$$

Here, $adj(y)$ denotes the neighbour or adjacent labels that our proposed loss consider. $|adj|$ denotes the number of classes in the $adj(y)$ set. We experimented to set $adj(y)$ in two ways: a) keeping all other labels other than y in the set, b) keeping the nearest n neighbours of y in terms of euclidean distance in embedding space. We find that the former gives a slight improvement over the later. But the second option yields faster training when the number of classes are high. Both MSE and the proposed neighbor-aware loss are convex functions. However, since the neural network itself is non-convex, $L(\Theta)$ is non-convex.

In Figure 6(b), we show the effect of neighbour-aware loss with our previous example of *blender* and *washer*. We see that, the proposed neighbour-aware loss is able to minimize the distance between projected audio embedding and their corresponding class label's embedding in word space and also maintain a margin of distance with audio embedding of neighbour classes. Thus, we get adequate separability between the projected embedding of *blender* and *washer* audio clips as desired.

6 TWO-LEVEL CLASSIFIER

SoundSemantics employs a two-level classifier to infer the most likely sound type for an input audio clip. The first level classifier

determines whether the input audio belongs to *heard* or *unheard* classes. The second level classifier makes the final determination of the most probable sound type for the input audio.

6.1 Heard vs. Unheard Determination

Recall that SoundSemantics is trained on examples from the *heard* classes only and the goal of the cross modal projection step is to create a projection of an input audio which is as close as possible to its corresponding textual label. Hence, the projection of audio clips from an *unheard* class are unlikely to be close to any class labels that are in our training data.

Using this hypothesis, we devise a simple threshold-based decision algorithm to determine whether an input audio clip belongs to an *unheard* class. For an input audio, u , it belongs to an *unheard* class if the following condition is true:

$$\min_{h \in H} \|E(h) - \rho(u)\| > \Pi \quad (6)$$

where, $\rho(u)$ is the audio embedding, H is the set of *heard* classes, Π is an empirically determined threshold, and $\|\cdot\|$ is the Euclidean norm. If this condition is false, u belongs to a *heard* class.

Figure 7: The accuracy of *heard* and *unheard* class detection depends on the threshold Π 's value.

The threshold, Π is empirically determined to maximize the accuracy of *heard* vs. *unheard* class detection over an empirical dataset. In Figure 7, we plot the accuracy for $\Pi \in [0.5 - 2.0]$. We observe that setting a high threshold fails to detect any class as *unheard* and the accuracy drops to almost zero for *unheard* classes. On the other hand, setting the threshold too small leads to poor results for *heard* classes as it determines every audio sample as *unheard*. Hence, there is a trade-off between the *heard* and *unheard* class detection accuracy. The optimum threshold is 1.25, for which, the classification accuracy of both the *heard* and *unheard* classes are around 85%.

6.2 Classification

Once an audio clip is determined to be from a *heard* or an *unheard* clip, we use a nearest neighbor classifier to get the class label. If the audio is detected as *heard* category, we consider labels from only the *heard* categories, and select the label that has the minimum Euclidean distance from the audio's projected embedding. Likewise, if the audio is detected as an *unheard* type, we consider labels from only the *unheard* categories, and select the label that has the minimum Euclidean distance from the audio's projected embedding.

7 IMPLEMENTATION NOTES

• *Mobile Phone Application.* We develop a mobile phone application to run SoundSemantics. The application records audio and classifies them using the models we have designed. We use Tensorflow Lite [1] to build the application to run our pre-built models. In Table 1, we list the run-time for inference of the two models. We see that the siamese network having four convolutional layers takes 112 ms on average for inference, which is about 4 times higher than the inference time for cross modal projection model having only two fully connected layers.

Table 1: Runtime of our algorithms

Operation	Execution Time(ms)
Siamese network inference	112
Cross Modal network inference	39

• *Training.* We train our models on a server machine and use the pre-built models in the mobile application. We use mini-batch with Adaptive-Moment-Estimation (Adam) [22] to train the models. We use Adam as it adapts its learning rate for faster convergence than an optimizer with a fixed learning rate algorithm such as Stochastic-Gradient-Descent (SGD).

8 EMPIRICAL EVALUATION

In this section, we empirically evaluate the proposed algorithms with well-known public datasets and report the effectiveness of different parts of SoundSemantics' pipeline. Note that our evaluation is primarily focused on SoundSemantics' capability in inferring acoustic events that do not have training examples. The data we use in the evaluation have common environmental noise and reverberation effects, and thus our neural network models and their evaluation implicitly included these artifacts. However, environmental effects vary in different acoustic environments and impacts any audio classification algorithm negatively. To deal with this, additional pre-processing steps such as noise removal, reverberation compensation, and augment training should be applied in conjunction with SoundSemantics.

8.1 Datasets

We use three datasets in the empirical evaluation: ESC-50, ESC-10, and AudioSet [14, 25].

ESC-50 and ESC-10 are two popular datasets for audio event recognition and have been used in several recent audio classification papers [2, 25]. These two datasets have labeled collection of 2000 and 400 environmental audio recordings, respectively. There are a total of 10 and 50 classes with 5 seconds long 40 examples per class for ESC-10 and ESC-50 dataset, respectively. The datasets are provided with five folds for easy benchmarking across all algorithms. The ESC-50 also divides the dataset into categories such as *Animals*, *Natural Soundscapes*, and *Interior*.

AudioSet is primarily used to learn priors for transfer learning [25] to recognize acoustic events on datasets like ESC-10 and ESC-50. AudioSet is weakly labeled and most of the files are multi-labeled. On average, there are five labels per file [14]. This makes AudioSet unsuitable for our system's evaluation as we focus on

single label classification problems. Hence, we select a subset of the classes and files from AudioSet which have at least 50 single-labelled files. This results in a dataset that has over 1600 audio clips from 33 classes.

8.2 Performance of Robust Audio Representation

In Section 4, we use a contrastive function (Equation 1) to learn representation of audio. Contrastive function directly learns to map similar audio closer and dissimilar audio farther in feature space. This gives our proposed Siamese network based robust audio representation advantage over traditional classifier network based feature learning. Here, we compare Siamese based robust audio representation with a traditional classifier network based feature learning, by evaluating the clusters generated by mapping audio samples in feature space. Better representation results in more separability between clusters of different classes and less variance in a cluster of same class. For this comparison, we used ESC-10 data set. We generated feature for each of the five folds using SoundSemantics' robust audio representation pipeline and convolutional neural network based classifier. Both of the networks have same number of layers and same parameters for fair comparison. Moreover, they were trained with same number of epochs.

Table 2: Our proposed Robust Audio Representation preserves better clustering properties than feature extracted from a convolutional neural network based classifier.

Algorithm	Intra-cluster Centroid Distance	Inter-cluster Variance
Convolutional Network	0.36	0.029
SoundSemantics' Robust Audio Representation	0.59	0.017

We compared the features learnt by the two algorithms using two metrics. First, the mean distance between the centroids of the clusters for different classes. Second, the inter-cluster variance among each class. From Table 2, we find that the mean distance of the centroids for SoundSemantics is 0.59, on the other hand the mean distance for convolutional network is 0.36. This shows that features generated by SoundSemantics is able to differentiate between the classes more efficiently than features generated by convolutional network. Then, we compare the quality of the clusters by comparing that the inter-cluster variance for SoundSemantics and convolutional network are 0.017 and 0.029, respectively. Therefore, we conclude that, our robust audio representation algorithm learns better representation than traditional convolutional networks.

8.3 Performance of Proposed Neighbour-Aware Loss

We compare our proposed neighbour-aware loss with MSE loss by comparing their accuracy in classifying audio events with training examples. We used ESC-10 dataset and ESC-50 dataset. For ESC-10 we did the experiment for all the folds. Moreover, we did the experiments with the categories listed in ESC-50 and reported the mean accuracy with variance as error bar in Figure 8. We refer the

category based ESC-50 dataset as ESC-50 Categorized. Note that, in a category the class labels are much closer in embedding space than a general ESC-10 dataset. For example, in Animal category some of the classes are : dog, cat, pig. They all are very close to each other in word embedding space which makes SoundSemantics' cross-modal projection based classification harder. In Figure 8, we see that our

Figure 8: SoundSemantics' proposed neighbour-aware loss yields better result in classifying heard classes. The effectiveness of neighbour-aware loss reflects more in ESC-50 Categorized dataset. In categorized dataset the class labels are closer in embedding space which makes the scenario harder for MSE loss. SoundSemantics' proposed loss function comes as winner because it is aware of the neighbor words.

proposed loss has an accuracy of 83%, whereas MSE loss has an accuracy of 82% for ESC-10 dataset. However, for categorized ESC-50 dataset where the labels are closer to each other in embedding space our proposed neighbour-aware loss has an average accuracy of 76% and MSE has an average accuracy of 72%. So, we see a performance boost of 4% here. As our proposed loss considers neighbour words when projecting audio to word embedding space, it performed better when the labels in the dataset are closer to each other compared to MSE loss.

8.4 Performance on Unheard Class

For evaluation, we consider five scenarios. The scenarios have 2, 3, 4 and 5 unheard classes respectively. Each scenario was evaluated atleast three times with different combinations of classes in both heard and unheard classes. For ESC 50 dataset, the unheard classes were selected from each category by making sure that the category has some classes in the heard category. For AudioSet we picked the unheard classes by selecting from the closest labels of the classes in the heard category.

In Figure 9, we show the accuracy of SoundSemantics for different number of unheard classes. We observe that for two unheard classes, the mean accuracy of SoundSemantics is around 90% for the ESC dataset and 76% for the Audioset. The drop of accuracy for Audioset is due to weak labeling and presence of noise in the audio clips. For both datasets, the accuracy drops as the number of unheard classes increases. The reason is that the accuracy of unheard sound detection is dependent upon the heard classes. An unheard sound is detected successfully only when there are heard sounds which are similar in acoustic properties as well as their labels' contextual properties in the text domain. The heard classes act as representatives of the corresponding unheard classes. As more classes fall into the unheard category, their representative classes in the heard category decreases. Therefore, we see that the accuracy drops to 60% when five classes are in the unheard category.

Figure 9: SoundSemantics' accuracy in unheard audio event classification decreases from 90% to 60% for 2 to 5 unheard classes in the ESC dataset. For Audioset, due to weak labeling and noise, the accuracy drops to 76% for 2 unheard classes.

8.5 Performance on Heard Class

SoundSemantics is a generalized acoustic event recognition system that is expected to encounter examples of both *heard* and *unheard* classes at runtime. To test SoundSemantics' accuracy on *heard* classes, we compare its performance with baseline algorithms such as Convolutional Neural Network (CNN) and Random Forest. Both of them are trained with same number of epochs. We use more epochs when using only CNN as it does not have the extra step of cross-modal projection. Note that all of these algorithms have the same number of layers and same design of network for fair comparison. We use the ESC-10 dataset in this experiment and perform a five-fold cross validation using the folds that the dataset comes with. We also report the performance of the best classifiers for the dataset, i.e., EnvNet [49] and ConvRBM [44].

Figure 10: SoundSemantics is able to classify heard audio events better than baseline algorithms because of its robust audio representation and cross-modal projection.

In Figure 10, we see that the mean accuracy of SoundSemantics for heard classes is 82.25% which is higher than that of a CNN with our cross-modal semantic projection (79.4%), a regular CNN (77%), and random forest (71.8%). EnvNet and ConvRBM achieves 84.9% and 86.5% accuracy, respectively. Note that both algorithms use raw audio as the input to the network unlike SoundSemantics which feeds time-frequency domain representation as the input to the network. These papers [44, 49] learn the frequency filterbanks from raw audio data, and thus learns better representation and eventually attains a higher accuracy. SoundSemantics' Siamese network-based robust representation and cross-modal projection steps are generic and can be used with the techniques proposed in these papers to achieve similar or higher accuracy in heard class detection.

8.6 Performance with Distractor Words

While SoundSemantics performs well in detecting unheard audio classes it requires the unheard label names beforehand. In this

experiment, we wanted to see SoundSemantics' performance when the unheard labels are not only limited to the expected classes, but has some extra random labels also. We refer these extra words in the unheard labels as *distractor words*. We performed two sets of experiments for a scenario with two unheard classes. In the first, experiment the distractor words were chosen randomly and in the second experiment the distractor words were chosen from the k -nearest neighbours of the expected class label in word embedding space. For example, if we had *cat* sounds in unheard class then some of the nearest neighbour words were: *dog*, *rabbit*, *pet*, *mouse* etc. Note that, both of these experiments were done with multiple pairs of unheard class and only the average is reported here.

Figure 11: Random word injection in unheard class labels does not have effect on SoundSemantics' performance in unheard class detection. Adding closest words with respect to unheard labels in embedding space causes a certain drop in SoundSemantics' accuracy.

In Figure 11, we see the result with random words and nearest neighbour words as distractor words. With random words, the accuracy does not change even with 50 distractor words. This proves that the words have well spanned embedding and SoundSemantics is robust with random words injection. However, with nearest words added the accuracy drops around 10 distractor words. This is because, the distractor words are too close to the unheard class labels in the embedding space. We notice that from 10 to 50 distractor words, the accuracy does not dip very much and the reason behind this is further inclusion of nearest distractor words do not have any effect as they are far enough from the unheard class labels in embedding space.

9 REAL WORLD EVALUATION

We deploy SoundSemantics phone application in two real world scenarios involving both indoor and outdoor usages and evaluated our system's performance with other state of the art acoustic event classification algorithms. In this section we report the results from these two real world applications.

9.1 Kitchen Monitoring

We use SoundSemantics in an indoor scenario for kitchen monitoring. In this application, a user monitors his/her kitchen using SoundSemantics. The user provides a total of 10 acoustic event names which are listed in Table 3. The number of samples and total duration of audio are also listed in the table.

We consider three different training scenarios to stress out SoundSemantics' performance under different conditions. First, we consider that the user is giving 8 out of 10 classes' audio examples to SoundSemantics during training. So the number of classes in

Table 3: Dataset for Kitchen Monitoring.

Category	Examples	Count	Length
Appliances	dryer, utensil, basin, pouring, plate, blender, vacuuming, heater	320	1600 s
Pet	dog, cat	80	400 s

heard and unheard categories are 8 and 2, respectively. Second, we consider where 6 out of 10 classes' audio examples are given to SoundSemantics during training. And, the last scenario is the hardest, where SoundSemantics has only 2 classes' audio samples during training, that makes 8 out 10 classes in unheard category. For each scenarios, we randomly picked the classes for heard and unheard categories but making sure that classes from both *Appliances* and *Pet* have been selected. In Figure 12, we report the performance of SoundSemantics along with two baseline algorithms : convolutional neural network(CNN) and random forest for all three aforementioned scenarios. For each scenario, we consider three cases varying the ratio between samples from heard and unheard class in test dataset in the following ways: a) $\frac{\#heard}{\#unheard} = .25$, b) $\frac{\#heard}{\#unheard} = .5$ and c) $\frac{\#heard}{\#unheard} = .75$. Here, # represents the number of samples.

In the first scenario (Figure 12(a)) where only 2 classes were in unheard category, we see that SoundSemantics' accuracy is over 70% for all cases as there are enough representative classes in training samples. For the 1st case where only .25 fraction of test samples are from heard category baseline algorithms fall behind by a long margin (less than 20% accuracy) as they can not classify any of the classes from unheard category. The situation gets better for the baselines as the fraction of heard classes increase in test samples. The maximum accuracy of convolutional network approaches around 60% for the case with .75 fraction of test classes in heard category. For this case, SoundSemantics' accuracy(77%) still beats baseline algorithms by more than 15%. For the second scenario (Figure 12(b)), with 4 classes in unheard category the scenario becomes harder as unheard category classes have increased. Still, SoundSemantics was able to classify around 57% of the test samples even in the extreme case with only quarter fraction of test samples from heard category. In this case, the baseline algorithms accuracy are around 22% and 23%. The third scenario (Figure 12(c)), is the most challenging one as it deals with only two classes in the heard category. When the fraction of test samples which belongs to heard category is .25 we see that SoundSemantics' performance drops to 28% as it can not classify most of the unheard classes with such limited training data. Although it still outperforms conv-net(23%) and random forest(24%). With less proportion of test samples from unheard category, SoundSemantics' performance gets better as expected and reaches 77% when there are three-quarter of the test samples from heard category.

9.2 Street Monitoring

SoundSemantics was also deployed in outdoor scenario for street event recognition. Here, we showed the usecase of SoundSemantics in street event recognition using audio such as: car engine sound, honk or people talking etc. There are total 6 classes in this scenario and in Table 4, we described the data for this application.

Figure 12: SoundSemantics achieves better accuracy than baseline algorithms in monitoring kitchen for all scenarios with all proportion of heard vs unheard class in test samples. (a) Eight out of 10 class are in heard category. SoundSemantics’ accuracy remains over 70% for all cases. (b) For six out of 10 class in heard category, SoundSemantics’ accuracy drops from 75% to 58% with increasing proportion of test samples coming from unheard classes, whereas the best baseline’s accuracy drops to 23% . (c) For, two out of 10 class in heard category, SoundSemantics achieves 77% accuracy when majority of test samples are from heard category but drops to 28% with more test samples from unheard category.

Figure 13: SoundSemantics performs better than baseline algorithms for all cases in street monitoring. (a) When four out of six class are in heard category, SoundSemantics has more than 70% accuracy for all cases. On the other hand, baseline algorithms’ accuracy drops below 20% for some cases. (b) For three out of six class in heard category SoundSemantics outperforms all baseline algorithms for different cases. (c) When two out of six class are in heard class, for .25 fraction of test samples coming from heard category SoundSemantics’ performance drops to 51% which is still two times more than baselines’ performance.

Table 4: Dataset for Street Monitoring

Category	Examples	Count	Length
Vehicle	car, airplane, honk, siren	160	800 s
Noise	talking, shouting	80	400 s

For evaluation in outdoor environment, we again consider three scenarios. Scenario 1: out of 6 classes, 4 classes were in heard category and the rest of the 2 classes are in unheard category. Scenario 2: both heard and unheard categories have 3 classes each. Scenario 3: heard category has only 2 classes whereas unheard category has 4 classes. For each scenario we again vary the proportion of heard vs unheard classes to .25, .50 and .75 respectively.

In Figure 13, we report the result of SoundSemantics for all the cases in the three scenarios we mentioned. In Scenario 1 (Figure 13(a)), where only 2 classes are in unheard category, we find that SoundSemantics has accuracy of over 70% for all the cases. On the other hand, the baseline algorithms’ accuracy drops under 20% when most of the samples are coming from unheard category. In scenario 2 (Figure 13(b)), for SoundSemantics we see an accuracy of 75% for case 3 with majority of the samples in test cases coming from heard classes. However, when the ratio of heard classes in test data get decreased in case 1 and 2, the accuracy drops. Yet, SoundSemantics’ performance is better than baselines by a margin of greater than 30%. For scenario 3 (Figure 13(c)), where only two classes are in heard category, the accuracy for the case where 75% of test samples are from heard classes reaches up to 83% for

SoundSemantics. For heard category the classification has become a binary problem, and thus causes boost in accuracy for this case. However, for case 3, the accuracy drops and gets to 51% where 75% of the test samples are from unheard categories. For this case, baselines achieve a maximum accuracy of 23%.

10 DISCUSSION

• *Zero-Shot Learning Assumption.* We assume that audio classes with similar acoustic properties have semantically similar class labels. We use this assumption to learn the non-linear mapping function to project audio signals onto the word embedding space. When there are no semantically similar training examples corresponding to an unheard class, a zero-shot learner fails to recognize examples from that unheard class. However, in a large dataset, the chances that there is no semantically similar training example to an unheard class is relatively low. This is why zero-shot learning performs better on larger and diverse datasets than smaller and skewed ones.

• *Necessity of User-Provided Tag-List.* The user-provided additional labels do not have any influence on the training phase. They are only used in the classification step after the training has been completed. If we do not have these additional labels, then the search space becomes too large (i.e., as big as having all the words in our database for a language) in the classification stage. We have demonstrated this in Section 8.6 where we added extra labels (i.e., distractor words) along with user-provided additional labels. We find that the

inclusion of similar semantic words with additional labels causes a dip in the accuracy of unheard audio class detection.

Figure 14: SoundSemantics’ performance in recognition of an unheard audio event is contingent upon the acoustic and word embedding similarity between that event and classes in heard category.

- *Heard-Unheard Synergy.* SoundSemantics’ accuracy on unheard sound classification depends on the characteristics of heard classes. If the training classes have representative acoustic and semantic representations of rare audio events, SoundSemantics is able to infer it. Hence, as long as this assumption is satisfied, SoundSemantics can recognize rare, inconvenient, or unwanted events. For example, to detect the sound class *waves* that has no training examples, we need to have some class in the heard category that is close to *waves* in terms of acoustic properties as well as word semantics, e.g., *wind*. However, without the presence of such classes in the heard category, the performance of *waves* detection without examples drops. To visualize this, we conduct an experiment where we want to classify the unheard sounds of *waves*. For training, we use one class from the following set interchangeably in the heard category: {*rain*, *wind*, *thunderstorm*, *fire*}. We want to see the effect of these classes for *waves*’ zero-shot classification. We use the mean Euclidean distance between the MFCC features of the audio samples of the classes as a metric for the *Spectrogram-level distance* and the Euclidean distance between word embeddings as *Word-level distance*. We sort the classes {*rain*, *wind*, *thunderstorm*, *fire*} in terms of both spectrogram-level distance and word embedding-level distance from *waves*. In Figure 14, we show the sorted classes on the X and the Y axes based on the spectrogram-level and word embedding-level distances. All distances are normalized by their maximum. We see that, as the spectrogram level distance increases, the classification accuracy for *waves* drops. We find that, for *fire*, the accuracy drops to 33% as it is very distant from *waves* in both the spectrogram and the word level.

11 RELATED WORK

Audio classification has been a well known problem for decades. There are some approaches to deal with limited training data for audio classification. Some of the past works dealt with limited training data problem by generating new training examples with the help of perturbation acoustic characteristics of existing examples, e.g., frequency warping [17], modifying tempo [21], and adding simulated reverberation [9], and noise [36]. These works fall in the category of *data augmentation*. Another direction of works use classifier fusion [34], where outputs of multiple classifiers are combined to improve the overall accuracy. These techniques vary from simple voting and averaging [26], to ranking [15], to learning decision templates [15]. Very recent works [25] deal with the

problem of limited data with transfer learning [35] from other large scale audio dataset. Another branch of work [45] proposes classification algorithm for weekly-supervised dataset where the presence or absence of an audio event in a clip is marked but the absolute position of the audio event is not labelled. However, our problem is significantly different from these works as these papers do not consider the scenario when absolutely no training examples are present. Ours’ is the first work to deal with zero shot learning for audio data by leveraging semantic knowledge from textual domain.

In image domain, learning from limited or no data has been explored recently. [23, 52] focus on learning suitable image representation to classify images from very few or only one examples. These works are totally different from ours as we want to recognize audio events without any training data. The other branch of learning with limited data focuses on zero shot learning for image classification. The earlier practices of zero shot learning [29, 39] for image classification problem infer the labels of unseen classes using a two step algorithm. First, the attributes of the sample is inferred and then the class label is predicted from an attribute database, which has most similar attributes with the image. Recent works [47, 54] have explored the mapping between image features and semantic space by projecting image features into word embedding space. Similar line of works [10, 40, 55] project image feature into semantic space then searches the nearest class label embedding using ranking loss. Recent works such as [24] uses auto-encoder to generate embedding for image features by reconstructing the images from the projected embedding. Although these papers propose zero shot learning method for images, none of them addresses the problem for audio domain and the associated challenges with it such as audio representation learning. Our work is the first paper to propose a zero shot learning method for audio classification where we overcome the challenges for cross-modal learning between text and acoustic domain. To the best of our knowledge, we present the first system with capabilities of inferring acoustic events without training examples and report extensive evaluation of such system under varying scenarios.

12 CONCLUSION

This paper presents the first system for acoustic event classification without training data. We propose a novel context-aware audio representation embedded with semantic knowledge from textual domain. Using this novel feature mapping, we created a classification algorithm capable of inferring acoustic events with or without training examples. We evaluated our proposed system SoundSemantics with public audio dataset and presented the benchmarking for different parts of the whole pipeline in the paper. We also implemented a mobile application and deploy our system in two real world applications. We compared our system’s performance with other state of the art audio classification algorithms and showed that our system is capable of inferring acoustic events without any examples with an accuracy of 60% – 90% for different scenarios.

ACKNOWLEDGMENT

This paper was supported, in part, by NSF grants CNS-1816213 and CNS-1704469.

REFERENCES

- [1] Tensorflow. <https://www.tensorflow.org/lite/>.
- [2] AYTAH, Y., VONDRICK, C., AND TORRALBA, A. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems* (2016), pp. 892–900.
- [3] BLUMSTEIN, D. T., MENNILL, D. J., CLEMINS, P., GIROD, L., YAO, K., PATRICELLI, G., DEPPE, J. L., KRAKAUER, A. H., CLARK, C., CORTOPASSI, K. A., ET AL. Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus. *Journal of Applied Ecology* (2011).
- [4] CHEN, J., KAM, A. H., ZHANG, J., LIU, N., AND SHUE, L. Bathroom activity monitoring based on sound. In *International Conference on Pervasive Computing* (2005), Springer, pp. 47–61.
- [5] CHEN, Z., LIN, M., CHEN, F., LANE, N. D., CARDONE, G., WANG, R., LI, T., CHEN, Y., CHOUDHURY, T., AND CAMPBELL, A. T. Unobtrusive sleep monitoring using smartphones. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*.
- [6] CLAVEL, C., EHRETTE, T., AND RICHARD, G. Events detection for an audio-based surveillance system. In *2005 IEEE International Conference on Multimedia and Expo* (2005), IEEE, pp. 1306–1309.
- [7] DE GODOY, D., ISLAM, B., XIA, S., ISLAM, M. T., CHANDRASEKARAN, R., CHEN, Y.-C., NIRJON, S., KINGET, P. R., AND JIANG, X. Paws: A wearable acoustic system for pedestrian safety. In *Internet-of-Things Design and Implementation (IoTDI), 2018 IEEE/ACM Third International Conference on* (2018), IEEE, pp. 237–248.
- [8] DEDEOGLU, Y., TOREYIN, B. U., GUDUKBAY, U., AND CETIN, A. E. Surveillance using both video and audio. In *Multimodal Processing and Interaction*.
- [9] DICKERSON, R. F., HOQUE, E., ASARE, P., NIRJON, S., AND STANKOVIC, J. A. Resonate: reverberation environment simulation for improved classification of speech models. In *Proceedings of the 13th international symposium on Information processing in sensor networks* (2014), IEEE Press, pp. 107–118.
- [10] FROME, A., CORRADO, G. S., SHLENS, J., BENGIO, S., DEAN, J., MIKOLOV, T., ET AL. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems* (2013), pp. 2121–2129.
- [11] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., AND BENGIO, Y. *Deep learning*, vol. 1. MIT press Cambridge, 2016.
- [12] GUVENSAN, M. A., TAYSI, Z. C., AND MELODIA, T. Energy monitoring in residential spaces with audio sensor nodes: Tinyyears. *Ad Hoc Networks* 11, 5 (2013), 1539–1555.
- [13] HADSELL, R., CHOPRA, S., AND LECUN, Y. Dimensionality reduction by learning an invariant mapping. In *null* (2006), IEEE, pp. 1735–1742.
- [14] HERSHEY, S., CHAUDHURI, S., ELLIS, D. P., GEMMEKE, J. F., JANSEN, A., MOORE, R. C., PLAKAL, M., PLATT, D., SAUROUS, R. A., SEYBOLD, B., ET AL. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on* (2017), IEEE, pp. 131–135.
- [15] HO, T. K., HULL, J. J., AND SRIBARI, S. N. Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence* 16, 1 (1994), 66–75.
- [16] HUANG, E. H., SOCHER, R., MANNING, C. D., AND NG, A. Y. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Annual Meeting of the Association for Computational Linguistics* (2012).
- [17] ISLAM, M. T., ISLAM, B., AND NIRJON, S. Soundsifter: Mitigating overhearing of continuous listening devices. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (2017), ACM, pp. 29–41.
- [18] JAITLEY, N., AND HINTON, G. E. Vocal tract length perturbation (vtp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language* (2013).
- [19] JAPKOWICZ, N., AND STEPHEN, S. The class imbalance problem: A systematic study. *Intelligent data analysis* 6, 5 (2002), 429–449.
- [20] JORDAN, M. I., AND MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science* 349, 6245 (2015), 255–260.
- [21] KANDA, N., TAKEDA, R., AND OBUCHI, Y. Elastic spectral distortion for low resource speech recognition with deep neural networks. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*.
- [22] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [23] KOCH, G. Siamese neural networks for one-shot image recognition.
- [24] KODIROV, E., XIANG, T., AND GONG, S. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345* (2017).
- [25] KUMAR, A., KHADKEVICH, M., AND FÜGEN, C. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 326–330.
- [26] KUNCHEVA, L. I. A theoretical study on six classifier fusion strategies. *IEEE Transactions on pattern analysis and machine intelligence*.
- [27] KUNCHEVA, L. I., BEZDEK, J. C., AND DUIN, R. P. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern recognition* (2001).
- [28] LAMERE, P., KWOK, P., GOUVEA, E., RAJ, B., SINGH, R., WALKER, W., WARMUTH, M., AND WOLF, P. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong (2003), vol. 1, pp. 2–5.
- [29] LAMPERT, C. H., NICKISCH, H., AND HARMELING, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (2014), 453–465.
- [30] LEE, H., PHAM, P., LARGMAN, Y., AND NG, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems* (2009), pp. 1096–1104.
- [31] LEHMANN, E. L., AND CASELLA, G. *Theory of point estimation*. Springer Science & Business Media, 2006.
- [32] LU, H., BRUSH, A. B., PRIYANTHA, B., KARLSON, A. K., AND LIU, J. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *International conference on pervasive computing* (2011), Springer, pp. 188–205.
- [33] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint*.
- [34] MILUZZO, E., CORNELIUS, C. T., RAMASWAMY, A., CHOUDHURY, T., LIU, Z., AND CAMPBELL, A. T. Darwin phones: the evolution of sensing and inference on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (2010), ACM, pp. 5–20.
- [35] MITHUN, N. C., MUNIR, S., GUO, K., AND SHELTON, C. ODDS: real-time object detection using depth sensors on embedded GPUs. In *IPSN* (2018).
- [36] MORALES, N., GU, L., AND GAO, Y. Adding noise to improve noise robustness in speech recognition. In *INTERSPEECH* (2007), pp. 930–933.
- [37] NIRJON, S., DICKERSON, R. F., ASARE, P., LI, Q., HONG, D., STANKOVIC, J. A., HU, P., SHEN, G., AND JIANG, X. Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services* (2013), ACM, pp. 403–416.
- [38] NIRJON, S., DICKERSON, R. F., LI, Q., ASARE, P., STANKOVIC, J. A., HONG, D., ZHANG, B., JIANG, X., SHEN, G., AND ZHAO, F. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems* (2012), ACM, pp. 43–56.
- [39] NOROUZI, M., MIKOLOV, T., BENGIO, S., SINGER, Y., SHLENS, J., FROME, A., CORRADO, G. S., AND DEAN, J. Zero-shot learning by convex combination of semantic embeddings.
- [40] PALATUCCI, M., POMERLEAU, D., HINTON, G. E., AND MITCHELL, T. M. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems* (2009), pp. 1410–1418.
- [41] PICZAK, K. J. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, ACM Press, pp. 1015–1018.
- [42] RA, H.-K., SALEKIN, A., YOON, H.-J., KIM, J., NIRJON, S. S., STONE, D. J., KIM, S., LEE, J.-M., SON, S. H., STANKOVIC, J. A., ET AL. AsthmaGuide: an asthma monitoring and advice ecosystem. In *Wireless Health* (2016), pp. 128–135.
- [43] RUTA, D., AND GABRYS, B. An overview of classifier fusion methods. *Computing and Information systems* 7, 1 (2000), 1–10.
- [44] SAILOR, H. B., AGRAWAL, D. M., AND PATTI, H. A. Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification. *Proc. Interspeech 2017* (2017), 3107–3111.
- [45] SALEKIN, A., EBERLE, J. W., GLENN, J. J., TEACHMAN, B. A., AND STANKOVIC, J. A. A weakly supervised learning framework for detecting social anxiety and depression. *ACM Interactive, Mobile, Wearable, and Ubiquitous Technologies*, (2018).
- [46] SCHROFF, F., KALENICHENKO, D., AND PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 815–823.
- [47] SOCHER, R., GANJOO, M., MANNING, C. D., AND NG, A. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems* (2013), pp. 935–943.
- [48] TANNER, M. A., AND WONG, W. H. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association* 82, 398 (1987), 528–540.
- [49] TOKOZUME, Y., USHIKU, Y., AND HARADA, T. Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282* (2017).
- [50] TOREYIN, B. U., DEDEOGLU, Y., AND ÇETIN, A. E. Hmm based falling person detection using both audio and video. In *International Workshop on Human-Computer Interaction* (2005), Springer, pp. 211–220.
- [51] TZANETAKIS, G., AND COOK, P. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing* 10, 5 (2002), 293–302.
- [52] VINIYALS, O., BLUNDELL, C., LILLICRAP, T., WIERSTRA, D., ET AL. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems* (2016), pp. 3630–3638.
- [53] WANG, A. The shazam music recognition service. *Communications of the ACM* 49, 8 (2006), 44–48.
- [54] XIAN, Y., AKATA, Z., SHARMA, G., NGUYEN, Q., HEIN, M., AND SCHIELE, B. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 69–77.
- [55] ZHANG, Z., AND SALIGRAMA, V. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 4166–4174.