CSafe: An Intelligent Audio Wearable Platform for Improving Construction Worker Safety in Urban Environments

Stephen Xia
Columbia University
New York, New York, USA
stephen.xia@columbia.edu

Jingping Nie
Columbia University
New York, New York, USA
jn2551@columbia.edu

Xiaofan Jiang
Columbia University
New York, New York, USA
jiang@ee.columbia.edu

ABSTRACT

Vehicle accidents are one of the greatest cause of death and injury in urban areas for pedestrians, workers, and police alike. In this work, we present CSafe, a low power audio-wearable platform that detects, localizes, and provides alerts about oncoming vehicles to improve construction worker safety. Construction worker safety is a much more challenging problem than general urban or pedestrian safety in that the sound of construction tools can be up to orders of magnitude greater than that of vehicles, making vehicle detection and localization exceptionally difficult. To overcome these challenges, we develop a novel sound source separation algorithm, called Probabilistic Template Matching (PTM), as well as a novel noise filtering architecture to remove loud construction noises from our observed signals. We show that our architecture can improve vehicle detection by up to 12% over other state-of-art source separation algorithms. We integrate PTM and our noise filtering architecture into CSafe and show through a series of real-world experiments that CSafe can achieve up to an 82% vehicle detection rate and a 6.90° mean localization error in acoustically noisy construction site scenarios, which is 16% higher and almost 30° lower than the state-of-art audio wearable safety works.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; Embedded systems; • **Human-centered computing** → Ubiquitous and mobile computing.

IPSN '21, May 18–21, 2021, Nashville, TN, USA

ACM ISBN 978-1-4503-8098-0/21/05…$15.00
https://doi.org/10.1145/3412382.3458267

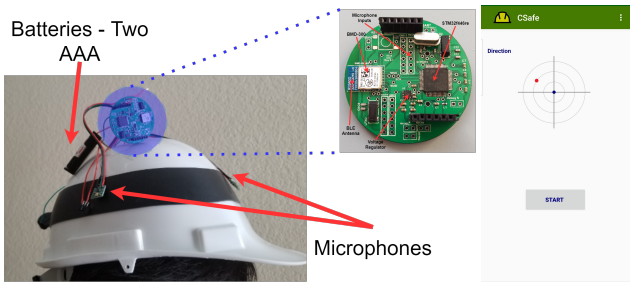


Figure 1: (Left) The CSafe embedded platform, consisting of an an array of four microphones integrated into a feature extraction system introduced in [1]. Two microphones are shown (the front and right microphones), while the other two are on the back and left side of the helmet. (Right) Screenshot of CSafe's smartphone system.

KEYWORDS

sound source separation, acoustic detection, adaptive filtering, wearables

ACM Reference Format:
Stephen Xia, Jingping Nie, and Xiaofan Jiang. 2021. CSafe: An Intelligent Audio Wearable Platform for Improving Construction Worker Safety in Urban Environments. In *The 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021) (IPSN '21), May 18–21, 2021, Nashville, TN, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3412382.3458267

1 INTRODUCTION

Vehicle-related accidents are one of the largest sources of construction worker injury. In fact, motor vehicle crashes are the number one cause of work-related deaths in the United States [2, 3]. These accidents arise in part because the worker is distracted by their work. To help reduce the number of worker-related vehicle accidents, we present CSafe, a low-cost wearable and smartphone platform, that can be easily integrated into common wearables such as helmets, hats, and

headphones. CSafe leverages an array of low-power microphones, signal processing, and machine learning classifiers to detect and localize oncoming vehicles, and alert construction workers in real-time.

There are two main challenges that make construction worker safety a difficult problem. The first challenge is that our wearable is battery-powered, resource-constrained and needs to run all of our noise filtering, vehicle detection, and vehicle localization machine learning classifiers in real-time. Specifically, our system needs to detect and alert the worker well in advance of when a vehicle would pass the user to give them ample time to react.

The second major challenge is that construction sites are very noisy. The power tools that construction workers will be operating by the side of the road can be orders of magnitude greater than the engine and tire sounds of an approaching vehicle, which will adversely affect any acoustic detection or localization algorithm we may decide to implement.

We take an audio-based approach to detect, localize, and alert users to oncoming vehicles in real-time and in a resource-limited wearable. Works that leverage sensors that provide more rich information about the environment, such as camera or LIDAR, have high computational and battery requirements, making it difficult to create a long-lasting battery-powered platform. Although audio does not provide as rich information about the environment as sensors like cameras, standard microphones provide 360 degree coverage and require less computation to sample and extract features. As such, we chose to create a hardware platform consisting of an array of microphones sampled by a low-power embedded platform. The embedded platform performs feature extraction and transmits wirelessly to the more powerful smartphone platform, which runs signal processing and machine learning algorithms to perform vehicle detection and localization, and send alerts to the user. Our embedded platform can be easily integrated into common wearables, such as headphones, helmets, and hats. In this work, we chose to integrate our hardware platform into a construction worker helmet, as shown in Figure 1.

To account for loud construction tools that will be present in a construction site we propose an energy-efficient sound filtering architecture that contains content-based separation and spatial separation in a feedback configuration, which utilizes known or learned models of sounds to iteratively remove noise and boost target sounds. Our goal is to show that our novel filtering architecture can be used in conjunction with existing vehicle detectors to boost performance in noisy construction site environments rather than serve as a complete replacement. We first develop a novel noise filtering algorithm called Probabilistic Template Matching (PTM). PTM is a low computation source separation algorithm that leverages statistical "templates" of noises to filter out these sounds. Next, we develop a novel noise filtering architecture that intelligently leverages PTM and multi-channel filtering methods to robustly filter out construction tool sounds from the environment. Our novel filtering architecture differs from existing works in sound source separation in that we intelligently leverage both single-channel source separation and multi-channel source separation in a feedback architecture to more robustly remove overpowering construction noise over time. We show that our novel filtering architecture can run in real-time on a low-resource embedded + smartphone platform and can improve vehicle detection by up to 12% more than other state-of-art source separation algorithms.

We make the following contributions in this paper:

- We create, **CSafe**, a low-cost, end-to-end wearable system to provide real-time alerts of oncoming cars to construction workers even in environments with tools that are orders of magnitude louder than approaching vehicles. We perform real-world experiments and show an improvement in vehicle detection by 16% and a $30°$ reduced localization error over existing systems.
- We develop a novel and light-weight single-channel source separation algorithm, called Probabilistic Template Matching (PTM) that uses learned statistical models or "templates" of construction tool sounds to filter them out.
- We develop an adaptive and selective noise filtering architecture that allows users and applications to select specific types of construction sounds to filter out. Our architecture integrates both PTM and multi-channel source separation methods in an adaptive feedback architecture to robustly filter out overpowering construction tool noise over time.
- We show that by applying our novel adaptive filtering architecture to vehicle detection, we can improve the overall vehicle detection accuracy by up to 15%. We also show that our architecture improves the vehicle detection rate by up to 12% more than other state-of-art audio source separation methods.

2 RELATED WORKS

Pedestrian safety has similar aims as construction worker safety, in that both attempt to reduce the number of vehicle accidents. However, the technical challenges between these two problems are very different. In construction worker safety, the presence of construction tools, that are many orders of magnitude louder than that of approaching vehicles poses a challenge that is typically not seen in pedestrian safety. In these scenarios, the sounds of loud tools effectively mask the sounds of oncoming vehicles.

Existing works on pedestrian safety can be broadly divided into two categories: communication-based and sensing-based. Communications based safety systems leverage developments in vehicle to vehicle, pedestrian, and infrastructure communication protocols, such as Dedicated Short Range Communication (DSRC) or even WiFi, to directly communicate the presence of vehicles to nearby people [4, 5]. However, these protocols are currently not natively supported by most smartphones, vehicles, and city infrastructure, making widespread adoption of these methods currently infeasible.

Sensing-based solutions equip users with sensors to detect and alert people of nearby vehicles. Image-based sensors, such as cameras and LIDAR, are commonly used in many applications, including vehicular safety. These solutions, leveraging machine learning and deep learning, have been shown capable of detecting nearby objects and predicting nearby vehicle movements [6–8]. However, these solutions both have high sampling and computational requirements, making it difficult to incorporate into a low-power wearable. A secondary issue is that both LIDAR and cameras provide rich sensory information, introducing user privacy concerns.

Microphones that passively measure surrounding audio may not provide as much sensory information as sensors such as LIDAR or camera, but provide 360 degree coverage around the user and are low power. It has been shown in various communities that audio can be used to detect and/or localize a variety of different sounds [9–11], and has been used in a wide range of mobile and wearable applications to address concrete challenges [12–14]. [15] introduces a smartphone based system for detecting vehicles that requires the user to constantly hold out their smartphone, which is impractical for workers. [16] introduces a system that outfits vehicles with an ultrasound emitter and uses a smartphone to detect emitted chirp patterns. Requiring all vehicles to be outfitted with an ultrasound emitter makes widespread adoption of this method difficult. [1, 17–20] introduce systems that detect and localize vehicles using smartphone systems and microphones embedded into wearable systems. However, these works do not account for situations in which non-vehicle noise is orders of magnitude greater than that of approaching vehicles, making them infeasible for ensuring the safety of construction workers.

There are a variety of commercial products and research works that address construction worker safety, including jackets, helmets, and smartphone systems, that monitor fatigue and posture [21, 22] or provide cooling and heating relief [23]. These methods do not warn workers of oncoming dangers such as approaching vehicles. There are also works and products that deploy sensors (e.g. RFID or proximity sensors) on either the construction worker or large equipment. These sensors can then be used to quickly locate workers in case accidents occur [24] or can send an alert to the worker if s/he comes too close to dangerous equipment [25]. These works require an installation phase in the construction site and cannot account for passing non-construction vehicles.

3 OVERVIEW OF SOURCE SEPARATION AND SELECTIVE NOISE FILTERING

It is difficult to detect and localize vehicles using audio signals in real-time and on a resource-constrained platform. Our problem is made even more challenging because construction workers work in sites that are extremely noisy, with loud machinery and tools prevalent in the environment.

To clean and remove overpowering construction tool sounds from the environment, we consider incorporating audio source separation methods into CSafe. We recognize that there are decades of work on noise filtering and source separation and that there is no way for us to address all aspects on the topic. As such, we summarize these works into two broad groups: **spatial separation** and **content-based separation**, that we will discuss next.

3.1 Spatial Source Separation

Spatial source separation techniques rely on observing multiple observations of the environment with multiple microphones in placed in different locations. Because these methods require multiple microphones, these methods are also called *multi-channel source separation* methods. Some methods that fall into this category include beamforming [26–28], general adaptive filtering techniques (e.g. least means squares and weiner filtering) [29, 30], blind source separation techniques (e.g. independent component analysis) [31, 32]. With the exception of blind source separation techniques and adaptive filtering, *spatial source separation techniques generally require the location of the source to perform separation.* As such, many of these works assume the location of the source is known in advance, which is not true in a dynamic urban environment.

Traditional blind source separation techniques do not require the location of the source to be known in advance. However, these techniques generally perform poorly in real-world scenarios [33].

Adaptive filtering techniques, such as least means squares, filter out noise by observing a second correlated noise signal. In the context of audio wearables, we would require a two microphone setup where one microphone observes the environment for vehicles. In a construction setting, this microphone would also observe construction tools sounds. A second "noise" microphone would ideally observe just the noise of the construction tools, which would be used to clean the signal from the first microphone. However, the position of the microphones in a practical wearable would be close

together. If we were to integrate microphones into a helmet, then both microphones would observe relatively similar signals (e.g. the "noise" microphone would also observe the sound of passing vehicles). The denoising process would not only reduce the sound of construction tools, but that of passing vehicles, adversely affecting vehicle detection.

In light of these shortcomings for blind source separation and adaptive filtering, we decide to incorporate ideas in beamforming into our filtering architecture (Section 4.2).

3.2 Content-Based Separation

The second class of source separation algorithms is content-based separation, which uses learned knowledge and statistical models about specific sounds and noises in the environment to filter them out. Since these methods require data, but not multiple channels of audio, these methods are also known as *single-channel source separation* methods. Classes of techniques that fall into this category include but are not limited to dictionary learning (e.g. non-negative matrix factorization or NMF) and deep neural network methods [34–37].

Many deep learning architectures that perform source separation require millions of weights/parameters and a large amount of training data. For example, the network presented in [36] requires almost 2 million parameters to separate out two sources from a single channel of audio sampled at 16 kHz. Second, source separation neural networks must be trained using artificial mixtures. This is because neural networks require the ground truth signal to adapt weights for training; in our application, the ground truth signal would be the isolated vehicle sound. However, it is not possible to extract the isolated vehicle sound in a real-world mixture because the vehicle sound is corrupted by construction tool noise. Creating a network with these two requirements will not produce a robust and low-power solution for filtering out construction sounds and improving vehicle detection and localization. We present experiments to back this in Section 5.

Dictionary learning methods learn a set of bases or a "dictionary" that capture most of the important features of a sound type. When a new signal arrives another optimization is performed to discover the coefficients or weights of each basis or "word" in the "dictionary" that the observed signal is comprised of.

We observed while experimenting with dictionary learning methods, like NMF, that the separation quality can be very poor because dictionary learning seeks to precisely deconstruct an entire signal into a weighted sum of its learned "words". However, our learned "dictionary" may not contain a learned representation of all sounds present. For instance, if we learn a "dictionary" for construction sounds, and someone begins speaking, NMF would attempt to fit construction sound "words" into speech, which would yield poor results.

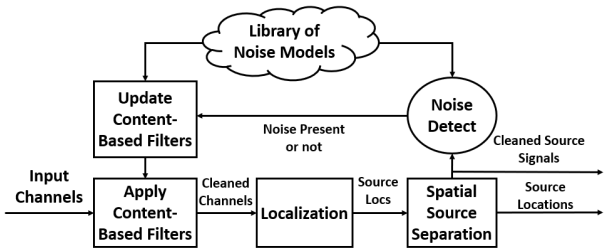


Figure 2: CSafe's full source separation architecture combining PTM (content-based separation) and spatial separation.

This brings up an assumption made by many content-based separation works: *there is a model available for all the types of sounds present in our observation.* This assumption is not always true in a dynamic environment, and it is not feasible to have a model of every possible sound in the environment. *These points motivate a need for a fast content-based separation algorithm that can separate and filter out noises for which we have models for, while leaving all other sources, which we may not have prior knowledge for, intact.*

In light of these shortcomings for content-based methods, we develop a novel light-weight single-channel source separation algorithm called Probabilistic Template Matching (PTM) that leverages learned models of construction sounds to filter them out. We integrate PTM into a novel filtering architecture described in Section 4.3 that uses a noise detector to control the level of filtering that PTM provides based on how dominant the construction sound is in the environment. In this way, our architecture only requires models of the noise (construction tools) and does not require knowledge of all other sounds in the environment (e.g. vehicles) as dictionary learning algorithms typically require. The algorithm is also similar to general adaptive filtering techniques, as discussed in Section 3.1, but does not require a second microphone in close proximity, that would likely observe and diminish vehicle sounds as well.

4 CSAFE FILTERING ARCHITECTURE

To perform robust urban and construction noise separation, we propose a novel adaptive filtering architecture, shown in Figure 2, that leverages both content-based separation and spatial source separation techniques. Our architecture incorporates both a noise detection module and a multi-source localization module required for both content-based and spatial separation and uses a feedback loop to adaptively learn better filter coefficients over time. We also propose a novel and light-weight content-based separation technique called *Probabilistic Template Matching* (PTM) that allows

users and applications to tune the amount of noise suppression that CSafe provides. Our full architecture allows users to inject their own recordings of nearby tool sounds for more robust noise filtering. In the following subsections, we first introduce our novel content-based separation algorithm, Probabilistic Template Matching. Next, we introduce how we integrate our novel content-based separation algorithm with our spatial separation module to create an adaptive filtering architecture that intelligently leverages both multi-channel and single-channel separation techniques to more robustly filter out construction sounds and improve vehicle detection and localization. All figures of waveforms, mixtures, and quantitative analysis presented throughout this paper were generated with real-world recordings of mixtures over the air rather than digitally mixing sources, which is commonly performed in many works that propose source separation algorithms. This is to ensure that our examples and methods are representative of real-world scenarios.

4.1 Content-Based Separation: Probabilistic Template Matching

We propose Probabilistic Template Matching (PTM) a lightweight content-based source separation algorithm to filter out construction noises, while reducing the amount of suppression of other interesting sounds, such as vehicles, in the environment. The algorithm uses "templates" of different noises commonly found in urban environments to statistically extract and filter them out. *PTM does not require knowledge of every source in the environment to perform noise filtering*, unlike in traditional dictionary-learning methods.

4.1.1 Probabilistic Template Matching. The main idea behind Probabilistic Template Matching (PTM) is to generate a filter, or a set of coefficients $\alpha_i(n)$ given a window of audio, where $\overrightarrow{X}(n) = [|x(\omega_1, n)|, |x(\omega_2, n)|, ..., |x(\omega_B, n)|]^T$ is the magnitude of the time-frequency representation of time window n, such that the probability of the filtered window $\overrightarrow{Z}_\Lambda(n)$ being an instance of a noise of class c_0 is minimized. The definitions of our inputs ($\overrightarrow{X}(n)$) and outputs ($\overrightarrow{Z}_\Lambda(n)$ and $\alpha_i(n)$) are summarized next.

$$\overrightarrow{X}(n) = [|x(\omega_1, n)|, |x(\omega_2, n)|, ..., |x(\omega_B, n)|]^T$$

$$\Lambda_n = diag\left(\alpha_1(n), ..., \alpha_B(n)\right)$$

$$\overrightarrow{Z}_\Lambda(n) = \Lambda_n \overrightarrow{X}(n)$$

B refers to the number of frequency bins in our time-frequency signal representation. The *diag* operator creates a diagonal matrix of size B x B, where the off-diagonal entries are all zero and the diagonal entries are filled with the filter coefficients, $\alpha_1(n), ..., \alpha_B(n)$.

We first make the assumption that the loud noise that we wish to filter out, c_0, can be described by a "template" represented by a Gaussian distribution:

$$c_0 \sim N\left(\overrightarrow{\mu_{c_0}}, \Sigma_{c_0}\right)$$

From this assumption, an observed signal containing noise c_0 at timestep n is generated by drawing a B dimensional vector from $N\left(\overrightarrow{\mu_{c_0}}, \Sigma_{c_0}\right)$. This vector is the time-frequency representation of the noise c_0 at timestep n, where each dimension corresponds to a different frequency component of the noise. The noise corrupted signal, $\overrightarrow{X}(n)$ would then be generated by adding in the other unknown signals (e.g. vehicle) from the environment. If c_0 has high energy over most other sounds in the environment, then the probability that our observed signal $\overrightarrow{X}(n)$ is an instance of noise class c_0, $P\left(\overrightarrow{Z}_\Lambda(n)|c_0\right)$, will be very high. Our goal is to generate filter coefficients $\alpha_i(n)$ that will reduce this probability.

However, if we minimize this probability without any constraints, all coefficients will tend to 0, cancelling out all sounds in the environment. To avoid this we introduce a novel constraint, yielding the following optimization problem shown in Equation 1.

$$\begin{aligned} &\arg\min_{\alpha_1,...,\alpha_B} P\left(\overrightarrow{Z}_\Lambda(n)|c_0\right) \\ &s.t. D\left(\overrightarrow{Z}_\Lambda(n)||\overrightarrow{X}(n)\right) < \beta \end{aligned} \tag{1}$$

$$D\left(\overrightarrow{Z}_\Lambda(n)||\overrightarrow{X}(n)\right) = \sum_{i=1}^{B}\left(\frac{\overrightarrow{Z}_\Lambda(n)_i}{\overrightarrow{X}(n)_i} - \log\frac{\overrightarrow{Z}_\Lambda(n)_i}{\overrightarrow{X}(n)_i} - 1\right) \tag{2}$$

The idea is still to minimize $P\left(\overrightarrow{Z}_\Lambda(n)|c_0\right)$ as much as possible, removing out as much of noise c_0 from our observation. The divergence constraint $D\left(\overrightarrow{Z}_\Lambda(n)||\overrightarrow{X}(n)\right)$ is in place to keep the amount of change between the filtered signal and the raw signal within a threshold β so that the filter coefficients do not completely remove all sounds from the environment. We use a static divergence constraint rather than another probabilistic constraint because we cannot assume that we have models of every possible sound in the environment. Making the assumption of knowing every sound in the environment is not feasible as there is an infinite number of potential sounds that could occur in the environment. Additionally, we chose to use the Itakura-Saito divergence metric, because of its equal weight on frequency bins with low and high energy, which is favorable for audio processing applications [38].

Optimizing over this loss function using Lagrange multipliers yields Equation 3.

$$L = \log\left(P\left(\overrightarrow{Z}_\Lambda(n)|c_0\right)\right) + \lambda D\left(\overrightarrow{Z}_\Lambda(n)||\overrightarrow{X}(n)\right) \quad (3)$$

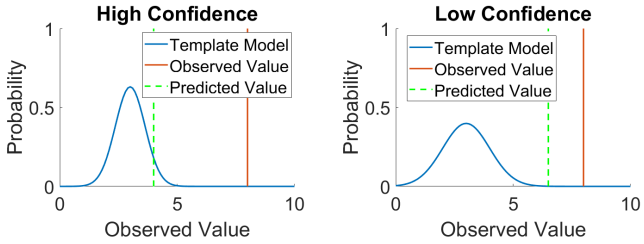
Finally, we arrive at the gradient update for each time window by taking the partial derivatives of our loss function L with respect to our filter coefficients $\alpha_i(n)$ and substituting it into the gradient update shown in Equation 4.

$$\alpha_i(n+1) = \alpha_i(n) - r\frac{\partial L}{\partial \alpha_i(n)} \quad (4)$$

One subtle point to note is that the learning rate r and the λ weight term are application tunable parameters that can be used to increase or decrease noise suppression. Higher levels of suppression will remove more noise, but will also leave a higher chance of removing out non-noise sounds from the environment. Conversely a lower suppression level will remove less noise, but will also remove less non-noise sounds from the environment. Through experimentation, we found values of $\lambda = 1e-5$ and $r = 1$ to consistently yield the best separation results and highest improvement in vehicle detection rate. We use these values in our evaluation and experiments in Sections 5 and 8.

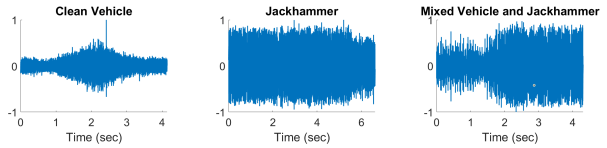
A visualization of the concept behind PTM is shown in Figure 3. In each of these four plots, we simplify our signal and models to one dimension ($B = 1$) for visualization purposes only. A probability distribution is shown in each of the four plots, corresponding to the Gaussian "template" probability distribution. Figure 3a shows a template in which we have high confidence in. This corresponds to a template with low covariance or variance. Since we are very confident in our model, PTM estimates and extracts out a noise value that is very close to the "template" mean. Figure 3b shows a template with low confidence. This means that for this dimension, we have observed values that fluctuate greatly, thus yielding a "template" with high variance. As such, PTM is more conservative and filters out less of the signal.

Figure 4 shows examples of a clean passing vehicle, a jackhammer, and the two sounds mixed over the air and recorded in a real-world setting. Sounds generated from moving vehicles are primarily from their engines as well as the friction of tires on the ground. As such, throughout this work, we primarily focus on detecting and localizing these sounds from an approaching vehicle. As a comparison to neural network methods, we cleaned the mixed signal using a state-of-art neural network based source separation algorithm (MMDenseLSTM [36]). We trained the MMDenseLSTM network using artificial mixtures of vehicle and construction sounds as described in Section 5, since it is not possible to train source separation neural networks using mixtures recorded in the real world.

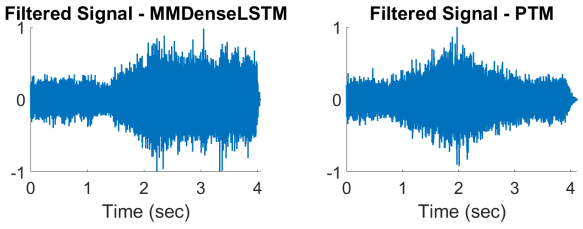


(a) (b)

Figure 3: a) Plot demonstrating the extraction process of PTM with a highly confident "template". We see that if we are more confident in the frequency component of our sound source (lower variance), PTM will extract coefficients or generated "predicted values" closer to the mean of the model template. b) However, if the model of the frequency component has higher variance (e.g. we are less confident in value of the frequency component), PTM will extract coefficients further away from the mean of the template.



(a) Vehicle. (b) Jackhammer. (c) Vehicle and Jackhammer.



(d) Neural network [36]. (e) PTM.

Figure 4: Examples of filtering results of a vehicle sound mixed over-the-air in the real world with a jackhammer sound. We played the vehicle sound (4a) and jackhammer sound (4b) through two different speakers and recorded the mixture with a 4-channel uniform circular microphone array to generate the mixed waveform (4c). We see from visually inspecting the filtered waveforms that the MMDenseLSTM neural network approach (4d) was not able to remove as much of the jackhammer noise as PTM. (4e).

We show the filtered results in Figures 4d and 4e. Through visual inspection, we can see that PTM is able to recover more of the characteristics of the clean vehicle sound than

the tested neural network method. A large part of the neural network's poor real-world performance is because it is not possible to use real mixtures to train a neural network for source separation. Using artificial mixtures may not capture all of the intricacies of the mixing process over the air in a real environment. One of PTM's strengths over neural network-based approaches is that it only requires models and data from a single sound source (e.g. the construction sound) and does not require sound mixtures (e.g. vehicle + construction sound) and the isolated cleaned signal (e.g. vehicle) that would require us to use artificial mixtures during training. Since CSafe is a construction worker safety platform, PTM and CSafe's overall filtering architecture will only benefit the system if it can improve vehicle detection. We present results that support this in Section 5.

Although we represent distributions of our frequency domain noise "templates" as Gaussians, we show in Section 5 that our adaptive noise filtering architecture, that utilizes PTM, can still improve vehicle detection more than state-of-art source separation methods.

As a final note, we made the assumption that the "template" of our noise c_0 was available. This implies a need for a noise detector that can detect the presence of the noise and can provide a correct "template" that PTM can use to filter out noise. We describe and address both of these points next.

4.1.2 Noise Detection and Template Learning. One common assumption in many content-based separation works is that the sound they are trying to separate is present in the audio stream. This assumption is not always true in real and dymamic scenarios. Hence, a noise detector is required to determine whether to perform noise separation or not. The second concern is how to learn and obtain "templates" of noise to use for filtering, as described in Section 4.1.1. *We incorporate a noise detector to solve both the requirement of detecting the presence of noise in the environment and as a method for learning and providing templates required for PTM.*

In general, sound event detectors operate as follows:

$$P(X \in c) > \beta$$

X is the input representation of the signal (e.g. frequency spectrum), and c is the class of sound we are interested in detecting. If the probability that our input observation is an example of a noise of class c is greater than some threshold β, then we would detect sound c in this window.

We create our noise detector in a similar fashion and choose to use a Gaussian mixture model (GMM) to model this probability distribution for each class of noise. GMMs model a probability distribution using a linear combination of Gaussian distributions to model sub-populations within the data. Each Gaussian can be described with a mean and a covariance matrix. The mean value is the most probable value that our features will take on if our signal is indeed a sound of the specific class we are trying to detect; this is another way of saying that the *mean values of the Gaussian distributions that make our GMM noise detector can be used as templates for PTM.* The covariance is a measure of uncertainty in our template and will also be used in PTM as described in Section 4.1.1. In this way, we not only create a noise detector, necessary for intelligently applying content-based source separation, we can also leverage the way GMMs model data to provide and learn templates required for our content-based separation algorithm, PTM.

4.1.3 Generalizability. Sound source separation, like many problems that have been addressed with machine learning and deep learning, suffers immensely from lack of generalizability. When we refer to generalizability, we refer to the ability of our models and algorithms to separate out and deal with unseen examples of our target noise. We are not referring to the ability of our architecture to denoise all different types of sounds. We concern ourselves only with loud construction sounds, as these are the sounds in environment that are most likely to overpower the sound of the engines and tires of passing vehicles. Other urban noises such as passing animals or the sounds of walking are generally lower in volume and will not be as consistently present; a well-trained vehicle detector can more easily account for these scenarios than situations where, for example, a worker is continuously operating a loud jackhammer.

In CSafe, we provide pretrained models of common power tool sounds in construction sites. However, we realize that such models could not possibly account for every single type of tool. As such, we allow users the option to record sounds in the environment of their work, allowing CSafe to build template and noise detection models on the spot that are tailored to the current work environment.

4.2 Spatial Separation: A Filter Bank Approach

Most spatial source separation methods require the location of the sound source in advance. Since there could be multiple sources in the environment that we may need to identify (e.g. vehicles) or filter out (e.g. loud construction sounds), we first need to identify and localize these sources before performing spatial separation.

4.2.1 Multiple Source Localization. A common method to perform localization is to estimate the relative delay of a sound source arriving between multiple microphones in an array and use these estimates to triangulate and estimate the direction of arrival. A power-based metric, such as cross-correlation, is commonly used to accomplish this. First, the cross-correlation is computed between different

microphones at different time shifts. The shift with the highest cross-correlation is estimated as the relative delay. These relative delays between microphones can then be passed into a machine learning classifier or directly used to triangulate the direction of the source. This architecture was used in a similar audio safety platform [1]. The biggest drawback in using a power-based metric and selecting the greatest peak is that it will tend to localize the loudest sound source. In construction sites, this is often the tool that the worker is operating, not the approaching vehicle. As such, it is necessary to consider methods that can estimate the location of multiple sources in the environment.

There are numerous works that introduce methods to perform multiple source acoustic localization, including MUSIC/ESPRIT and their variants [39, 40]. They all work by generating and analyzing a probability distribution of sources present at each direction of interest. This probability distribution is generated by comparing the phase difference between microphones in an array to the expected phase difference given the locations of each microphone in an array and a sound source coming from a specific direction. Once this probability distribution across all directions is generated, then a peak detection algorithm is employed to detect significant peaks and sources. The exact details and algorithms employed at each step varies from method to method.

We adopt the algorithm presented in [26] to perform multiple source localization due to a simplification that reduces the computational complexity of generating the probability distribution of the presence of sources across different directions. Algorithms like MUSIC and ESPRIT generate probability distributions by taking a subspace approach, by first dividing the energy found in a single frequency, f, to different directions based on the probability of a source appearing at each direction. This is repeated and aggregated for every frequency of interest. Generating a probability distribution across every frequency significantly increases computation, making it difficult to use such algorithms for real-time applications. The algorithm presented in [26] makes a simplification by assigning all the energy of a specific frequency to just a single direction, which reduces computation.

Localization algorithms generally do not consider the content or the class of sounds we are localizing. This means that as long as a single sound is loud enough, our source localization module can detect and localize this source. After our filtering architecture reduces the energy of construction tool sounds, we expect to observe greater energy from vehicle sounds, which would be detected and localized. This means that *our source localization module that we incorporate to localize noise sources in the environment will also double as our vehicle localization module.*

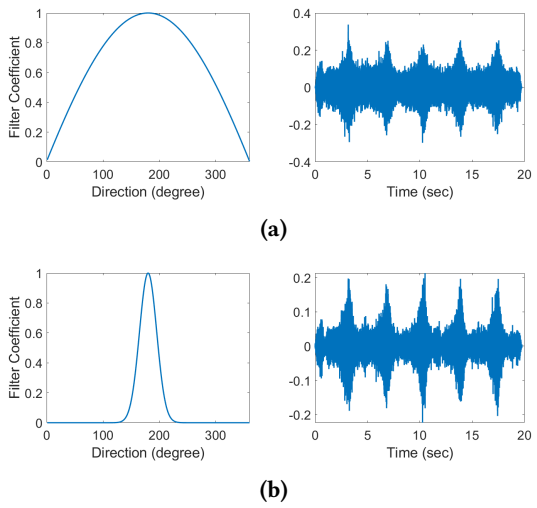


Figure 5: a) (Left) A wider and less directional filter. (Right) More energy of the original signal is preserved. b) (Left) A more directional filter. (Right) More energy from other directions are filtered out at the cost of greater signal distortion. The signal shown in both scenarios were recorded with a four microphone uniform circular array by playing a vehicle and jackhammer sound with separate speakers on repeat. The filtered signals (right plots) were generated by applying filters (left plots) to diminish the construction tool sound.

4.2.2 Filter Bank Spatial Separation. Spatial separation methods such as the DUET algorithm and beamforming create filters to apply to microphone channels that diminish frequency components that are not in phase with a signal coming from the direction of the sound source [41]. Adaptive beamforming methods achieve this by continuously updating the filter based on a cost function that captures and improves certain quality measures of the received signal (e.g. SNR) [27]. *To reduce computational requirements over an adaptive approach and allow for application and user tunability, we take a static approach, where we pre-generate a filter bank that we apply onto channels based on how close each frequency component aligns in phase with a signal coming from the direction of the sound source.* Figure 5 demonstrates our proposed approach.

For every direction d, we generate a confined Gaussian window centered around direction d, with variance σ^2. This variance term can be used to tune the amount of suppression of other directions provided by the filter. We apply this filter by scaling the energy of each frequency in the spectrum by the filter coefficient corresponding to the direction assigned to that frequency. The direction assigned to each frequency is generated while performing source localization, as described in Section 4.2.1. Figure 5a shows an example of

a filter with a higher variance and wider beamwidth, which suppresses out less energy from frequencies that do not align with the direction. While a filter with a more narrow beam and lower variance, shown in Figure 5b suppresses more energy from other directions, but may also remove more energy corresponding to other sounds in the environment (e.g. vehicles).

For all experiments and in the CSafe wearable platform presented throughout this paper, we choose to scan across $d = 24$ directions divided evenly across $360°$, yielding a $15°$ granularity. We select this parameter because it provides enough localization granularity for vehicle localization and noise separation while remaining low-cost enough to maintain real-time performance.

4.3 Full Filtering Architecture

In this section, we bring together all the different components proposed and introduced up until now to form CSafe's adaptive filtering architecture for robustly filtering out common urban construction noises from the environment. Figure 2 shows the full noise filtering architecture of CSafe. First, we sample a window of audio from our microphone array and compute each channel's FFT. Next, the content-based separation filters learned from our adaptive PTM algorithm during the feedback loop is then applied to clean up the audio channels. The individual channels are then provided as input to the source localization algorithm to obtain source locations. The source locations and cleaned microphone signals are then provided as input to the spatial separation module. The spatial separation module separates the individual sources in the environment. Each separated source is then passed to the noise detector to determine which sources are noise. Then, the noise sources are passed into the PTM module, where our content-based separation filters are adapted and applied to remove all detected noise in the environment, completing the loop. Sources that are not detected as noise are then fed into a vehicle detector to determine if there is a vehicle present. We use a 50 tree random forest as our vehicle detector, just like the vehicle detector used in the state-of-art audio safety platform presented here [1].

5 CSAFE FILTERING ARCHITECTURE REAL-WORLD EVALUATION

In this section, we compare the improvements in vehicle detection provided by CSafe's construction noise filtering architecture, introduced in Section 4, with existing noise filtering algorithms and a state-of-art source separation neural network. Throughout this section, all of our figures and quantitative analysis is performed on recordings of sounds in the real-world, not artificially mixed signals as is commonly done in many works presenting sound source separation methods. We also utilize two datasets for evaluation:

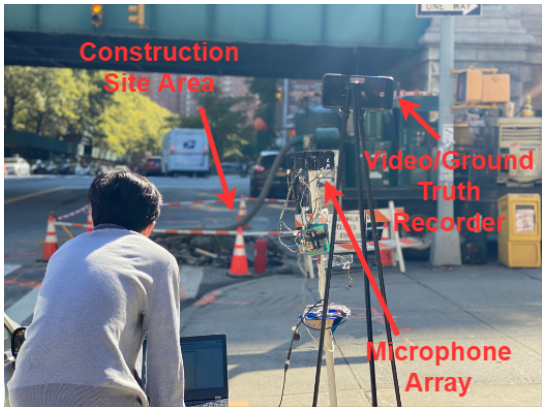


Figure 6: Experiment setup at a construction site.

- *Training dataset*: Consists of 175 audio clips of common construction sounds divided into jackhammer, drilling, hammering, sanding, sawing, and vacuuming. Each clip is 10 seconds long, yielding 30 minutes of construction noise recordings. We also added an additional 5 minutes of audio clips containing vehicles passing by for a total of 35 minutes of audio. All clips were extracted from labeled YouTube clips found in the Google Audioset dataset [42].
- *Construction site dataset*: To generate this dataset we created a four microphone uniform circular array with a diameter of 15 cm, which is around the average width of a human head [43], and took audio recordings from a real construction site. Our experimental setup and construction site is shown in Figure 6. We recorded a total of 40 minutes of audio, during which 76 vehicles passed by. The noises prevalent in this site were jackhammering, drilling, and vacuuming sounds.

To show the improvements in vehicle detection that PTM and CSafe's noise filtering architecture can provide, we present comparisons with 3 other algorithms: the linearly constrained minimum variance (LCMV) beamformer [27], hierarchical alternating least squares nonnegative matrix factorization (HALS NMF) [44], and the state-of-art source separation neural network MMDenseLSTM [36]. These algorithms were chosen as representative algorithms of a content-based separation algorithm (HALS NMF), a spatial separation algorithm (beamforming), and a neural network based algorithm (MMDenseLSTM). There is a training phase required for all of the algorithms except for the adaptive beamforming algorithm.

We trained the MMDenseLSTM network using artificially mixed construction and vehicle sounds generated from the *training dataset* for 12,000 epochs using a batch size of 1

Table 1: Confusion matrix metrics of vehicle detectors under different filtering schemes in a construction site environment.

	True Pos.	True Neg.	False Pos.	False Neg.	Vehicles Detected
CSafe	95%	94%	6%	5%	20/20
CSafe - Generic	89%	96%	4%	11%	20/20
CSafe - Spatial	85%	94%	6%	15%	17/20
MMDenseLSTM [36]	84%	92%	8%	16%	17/20
NMF HALS [44]	83%	94%	6%	17%	16/20
LCMV Adaptive BF [27]	84%	95%	5%	16%	17/20
Nonfiltered	80%	92%	8%	20%	16/20

Table 2: True positive (recall) for vehicle detection broken down by SNR and tool type of the environment.

	Drill (1.6 dB)	Vacuum (-5.5 dB)	Jackhammer (-8.6 dB)
CSafe	99%	96%	93%
CSafe - Generic	95%	93%	87%
CSafe - Spatial	91%	87%	83%
MMDenseLSTM [36]	72%	90%	84%
NMF HALS [44]	83%	92%	83%
LCMV Adaptive BF [27]	87%	84%	81%
Nonfiltered	83%	82%	76%

clip. It is not possible to use real mixtures to train a neural network for source separation because neural networks require the exact ground truth signal to adapt weights during training. The ground truth signal in this case would be the isolated vehicle sound during the period where both sounds are occurring. However, if the sounds are both occurring at the same time, then the isolated signals of both sounds would not be available.

We trained the HALS NMF algorithm using the *training dataset* to learn a 50 bases dictionary of "words" to separate vehicle and construction sounds.

To gain better insight into CSafe, we evaluated three modes of operation. First, we evaluated CSafe using only the spatial separation module; we denote this as CSafe - spatial. Next, we add in the content-based separation module to see how adding this next module could improve vehicle detection. There are two modes of operation for this module, as mentioned in Section 4.1.3. In the first mode, the user does not record noises from the environment for separation and uses an existing noise detector; we denote this mode as CSafe - generic. To train this construction noise detection and separation model, we use the construction sounds from the *training dataset* and create a 20 mixture GMM. The second mode of operation is where the worker records a segment of the loud tool s/he will be operating to use for detection and separation; we denote this as the default CSafe mode. To train the noise detector and source separation model for this mode, we take a small 10 second segment where only the tool sound is present from clips in the *construction site dataset* to create a 5 mixture GMM. Since there were three periods of different tools (jackhammer, vacuum, drill), we train three models for these individual sounds and apply the corresponding model (e.g. if a jackhammer is in use, we use the the jackhammer model learned from the environment).

Finally, to train and evaluate the vehicle detector, we used the *construction site dataset* with an 80%/20% train/test split. 56 out of the 76 recorded segments where vehicles were present were used for training and 20 segments were used for testing. All clips were divided and processed into 250ms windows with 50% overlap.

Table 1 shows the confusion matrix metrics for the vehicle detector under the different source separation and noise filtering schemes. The confusion matrix metrics measure the portion of 250ms windows that fall into each category. For instance, a 94% true negative rate means that the detector was able to correctly reject the presence of vehicles in 94% of windows where a vehicle was not actually present. The table also records the number of vehicles (out of the 20 passing vehicles used for testing) that the detector successfully detected. First, we see that the true negative and false positive rates of all the methods are relatively similar (> 90% and < 10%) respectively. This means that the detector is able to correctly identify periods where no vehicles are present very well (true negative) and does not mistakenly detect a vehicle when no vehicles are present (false positive). The differences are pronounced when we look at the true positive rates. The true positive rate is the percentage of windows where a vehicle is present in the environment and the vehicle detector is able to detect that vehicle. We see that when there is no filtering involved, the detector was only able to detect a vehicle in 80% of windows where a vehicle was actually present. We see that the LCMV beamformer, HALS NMF algorithm, and the MMDenseLSTM neural network were able to improve the detection rate to around 84%. CSafe - spatial also achieves a similar performance. This is because CSafe - spatial is using only the spatial separation module, which performs separation using similar concepts as beamforming. However, when we add in CSafe's content-based separation module, we see noticeable improvements in the true positive detection rate. Using a pretrained model of construction tool sounds (CSafe - generic) improved the the true detection rate to 89%. Further, if the worker decides to record the tool he is using to use for construction tool filtering (CSafe), the true positive rate improves even further to 95%. This is a 15% improvement over just using a vehicle detector with no filtering. Since more windows where vehicles are present are correctly identified, CSafe also improves the number of vehicles detected as shown in the same table.

Table 2 further breaks down the true positive detection rate by the signal-to-noise ratio (SNR) of vehicles to the construction tool sounds. There were three prevalent construction noises in the environment: jackhammering (-8.6dB), vacuuming (-5.5dB), and drilling (1.6dB). Only one construction tool noise was present at any given moment. Though

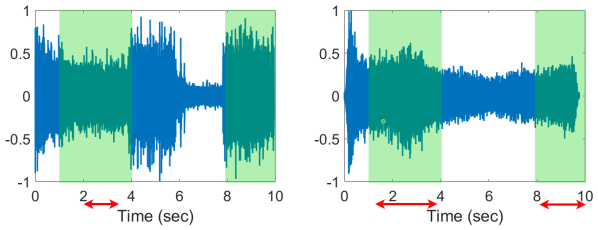


Figure 7: The two plots show a recording of an approaching vehicle in presence of a power tool sound when: a) No noise filtering is applied, and b) CSafe's noise filtering architecture is applied. The green segments highlight the ground truth for when a vehicle is present and the red arrows highlight the segments where the vehicle detector detects the presence of a vehicle. We see in this example, that the vehicle detector can detect the presence of vehicles earlier in each segment after applying our noise filtering architecture.

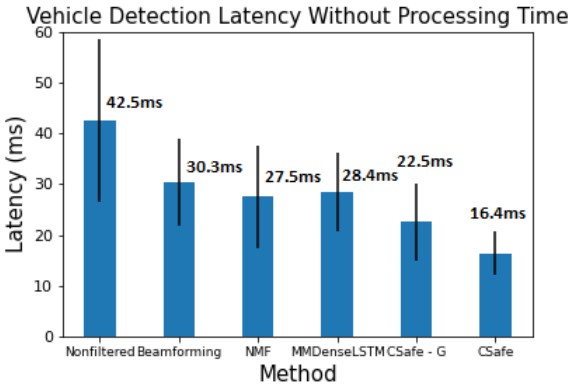


Figure 8: Average latency of vehicle detection of our vehicle detector without applying noise filtering and after introducing noise filtering. We see that CSafe's noise filtering architecture is able to reduce detection latency from 42.5ms, without any filtering, down to 16ms, which is a greater reduction than any of the other methods tested.

a bigger construction site may see many more tools being used at once, the microphones in the CSafe wearable platform will often observe only a single strong tool, which is the tool that the worker is currently operating or is closest to. As such, filtering out just this single tool can still account for many scenarios in construction worker safety. We obtain estimates of the SNR of these environments by taking the ratio between the average power of passing vehicles when no construction sounds were present with the average power of the construction tools when no vehicles were present.

In Table 2, CSafe followed by CSafe - generic attained the highest true positive detection rate across all SNRs, even when the power of the construction tool (jackhammer: -8.6dB) is almost an order of magnitude higher. An interesting point is that both HALS NMF and the MMDenseLSTM network see a decrease in performance between when the SNR improves from -5.5dB to 1.6dB. This is because at 1.6dB, vehicle sounds overpower construction sounds. Blindly applying a content-based filtering technique will result in signal degradation and distortion if the construction noise is low or not present at all. The CSafe noise filtering architecture does not suffer from this problem because it includes a construction noise detector that adaptively tunes the amount of construction tool sound to filter out.

Another subtle improvement that our noise filtering architecture provides is a reduction in detection latency. We illustrate this in Figure 7. The signals presented in this figure is that of a vehicle sound being played on repeat and mixed in the real-world with a sound of a power tool. Figure 7a shows the raw recorded signal, while Figure 7b shows the signal after applying our novel filtering architecture. First, we note that the peaks corresponding to the passing of the vehicle is much more noticeable in Figure 7b now that the majority of the construction sounds have been filtered out. Next, we placed each clip through our random forest vehicle detector. The segments highlighted in green are segments where the vehicle is audibly present. Segments highlighted in red show the time frames where our vehicle detector detected the presence of the vehicle. We can see that after applying our noise filtering architecture, we can detect a greater portion of the time frame that the vehicle is present. Additionally, we see that without applying our noise filtering architecture, the detector is only detecting the vehicle at its loudest point when the vehicle is passing by. Detecting a vehicle and alerting the user at this point is already too late because we need to give the user enough time to react. After applying our noise filtering algorithm, we see that the vehicle is detected much earlier before the peak when the vehicle is passing the user, giving the user much more time to react. In this example, the detector was able to detect the vehicle 90ms after the vehicle comes into audible range without applying any filtering, while the detector was able to detect the vehicle in 15ms when our noise filtering architecture is applied. This is critical in our worker safety platform, where every millisecond of reduced latency allows that much more time for the user to react to oncoming dangers.

To further quantify this improvement in latency, we repeated the steps described above for Figure 7 for every sample in our *noisy dataset* and plot the mean and variance in detection latency for each tested method in Figure 8. Without any filtering, our vehicle detector was able to detect vehicles

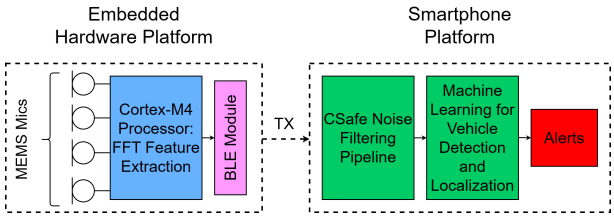


Figure 9: CSafe's full system architecture spanning the embedded and smartphone sub-platforms.

with an average of 42.5ms after the vehicle comes into audible range. Each separation method was able to decrease this lag, but CSafe - generic was able to bring the average delay down to 22.5ms. CSafe's non-generic mode was able to bring this delay even further down to 16.4ms, which is lower than any other method, allowing users that much more time to react after receiving an alert. We note that these latencies were generated using computer implementations of each algorithm while we experimented with different source separation algorithms to use in the final CSafe wearable system. A latency analysis of the full CSafe wearable platform is presented in Section 7.2.

6 CSAFE PLATFORM

In this section we introduce the CSafe wearable and smartphone platform. We also discuss our dataflow for the entire system spanning noise filtering, vehicle detection, and vehicle localization.

6.1 System Architecture

Figure 9 shows the full system architecture of CSafe spanning the embedded hardware platform and smartphone system. The hardware platform houses and samples from the microphone array. The FFT of each window sampled is then computed and these features are sent to the smartphone platform via Bluetooth Low-Energy (BLE), a low-power wireless transmission protocol. The features from the microphone array are then passed onto our novel adaptive noise filtering architecture (Section 4.3) that filters and removes urban and construction noise sounds from our signals. The outputs from this module are the filtered non-construction sound sources currently present as well as their current location with respect to the user. These sources are then passed to our vehicle detector to determine if any of the sources are vehicles. We use a 50 tree random forest detector, just like the vehicle detector employed by this [1] audio safety platform for pedestrian safety. Finally, an alert containing the direction of vehicles nearby is sent to the user if any of the sources are detected to be vehicles.

6.2 Embedded Hardware Platform

The CSafe embedded hardware platform was shown in Figure 1. We integrated the same embedded circuit consisting of a Cortex-M4 microcontroller and BLE module, presented in this work on pedestrian safety [1], into CSafe along with an array of four low-power MEMS microphones. These components are integrated onto a construction helmet, with the total cost of the major electrical components coming out to less than 20USD. We note that though the embedded circuit is the same, almost every other aspect, from the architecture and algorithms is novel because of the unique challenges present in construction worker safety.

The embedded hardware platform samples from the four microphone channels and computes the FFT for windows of 250ms with 50% overlap for each channel. These features are then transmitted to the smartphone over BLE. This provides enough granularity for CSafe to reliably detect vehicles with low latency while satisfying BLE's bandwidth limitations.

For this work, we embed our hardware platform into a helmet commonly worn by construction workers, but note that our platform can be easily incorporated into many other kinds of wearables, such as hats and headphones, for various kinds of users.

6.3 Smartphone Platform

Once the smartphone platform receives the frequency spectra of the microphone array, the smartphone executes our novel adaptive noise filtering architecture, introduced in Section 4.3. The output to the noise filtering architecture is the separated sound sources and corresponding locations. The outputs from the noise filtering architecture are then passed onto the vehicle detector. If any of the separated sources are detected as vehicles, a visual alert is sent to the user through the smartphone system indicating the direction and distance of the vehicle from the user. Since construction workers are often busy working and may not be able to look at their phones, we also send an audio alert to the user's headset/audio-enabled ear protection and provide haptic feedback through the smartphone system.

We decided to leverage the smartphone platform to execute most of the algorithms found on CSafe, as shown by the uneven partition of computation between the embedded hardware platform and the smartphone system in Figure 9. This is because there is a low amount of computational resources available on our Cortex-M4 based hardware platform when compared to much stronger processors available on modern-day smartphones. Additionally, we allow users to record audio clips of urban and construction noise in the immediate environment to generate models that are tailored to the current environment, allowing our architecture to more robustly filter construction tool sounds.

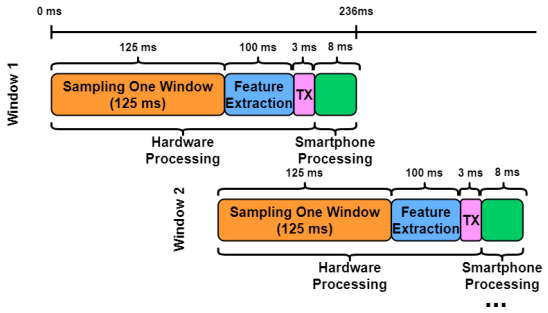


Figure 10: Breakdown of latency of each component of CSafe. CSafe completes one cycle of data sampling and computation in 236ms, which is on par with the reaction time of an average person.

7 SYSTEM EVALUATION

7.1 Power Consumption

CSafe's embedded hardware platform sees a 69 mA current draw off of a 3.3V power source. This allows CSafe to run for 14.5 hours off of two standard AAA batteries, connected in series, each with a 1000 mAh capacity, before recharging. This duration is more than enough for frequent daily use.

7.2 Latency

We measured the execution time of every component in our data pipeline, as shown in Figure 10. CSafe first samples 250ms windows of audio from its microphone array with 50% overlap. This means that CSafe calculates and transmits features every 125ms. The feature computation and wireless transmission to the phone takes 100ms and 3ms to execute, respectively. The CSafe smartphone system executes its entire pipeline, including the noise filtering pipeline, vehicle localization + detection, and sending user alerts in less than 8ms. This yields a full end-to-end latency, from when CSafe begins sampling a window to when CSafe is able to send vehicle presence and location alerts to the user, of 236ms. This is on par with the average human reaction time, allowing users enough time to react to oncoming vehicles.

8 CSAFE PLATFORM REAL-WORLD EVALUATION

In this section, we evaluate CSafe through a series of real-world experiments. We evaluated two aspects of CSafe: vehicle detection accuracy and localization accuracy. We evaluated CSafe in the same environment shown in Figure 6. To obtain the ground truth vehicle presence and location, we record all scenarios with an additional video recorder and sync the recorded video and audio with output logs from CSafe. In all experiments we compare CSafe with the state-of-art PAWS [1] audio safety platform, developed for pedestrian safety. In the following experiments, we analyzed 30 vehicles that passed by for the PAWS system. We also analyzed 30 vehicles each for CSafe and CSafe - generic.

Table 3: Confusion matrix metrics comparing both modes of CSafe with a state-of-art pedestrian safety system [1]

	True Pos.	True Neg.	False Pos.	False Neg.	Vehicles Detected
CSafe	82%	96%	4%	18%	29/30
CSafe - Generic	78%	99%	1%	22%	30/30
PAWS [1]	66%	99%	1%	34%	21/30

8.1 Vehicle Detection

Both CSafe and CSafe - generic detected a high percentage of vehicles (29 out of 30 and 30 out of 30, respectively). The PAWS system was only able to detect 21 out of 30 vehicles. Table 3 compares the confusion matrix metrics for vehicle detection of CSafe and PAWS. The confusion matrix metrics list the percentage of audio frames, rather than vehicle counts, that were categorized as a true positive, true negative, false positive, and false negative.

We see that for CSafe, both true positive and true negative rates are very high (CSafe - generic: 78% and 99%; CSafe: 82% and 96% respectively), while PAWS has a much lower true positive rate (66%) because of the presence of the loud construction tools obfuscating the sound of the oncoming vehicles. CSafe improves the true positive vehicle detection rate over PAWS by 16%.

The usability of the system is greatly affected by the false positive rate, which is the percentage of windows where a vehicle is not present, but the detector incorrectly detects a vehicle. If the false positive rate is too high, a user may become annoyed and less likely to heed the alerts of the system later on. We see that the false positive rates of both PAWS and CSafe are relatively low.

Finally, the false negative rate should be very low to avoid missing too many vehicles, which could be life threatening. We see that due to the added overpowering noise of construction tools and its lack of a mechanism to deal with this unexpected noise, PAWS's false negative rate is very high, at 34%. Because of their noise filtering mechanisms to filter out the loud construction sounds in the environment, both modes of operation for CSafe achieve a much lower false negative rate of around 20% or lower.

Overall, the high true positive rate and low false negative detection rates contribute significantly to CSafe's improved performance over PAWS as an audio safety platform suitable for construction worker safety and other scenarios beyond pedestrian safety.

Table 4: Localization error comparison between CSafe and a state-of-art pedestrian safety system, PAWS [1]

	Avg. Error (degree)	Std. Dev. Error (degree)
CSafe	6.90°	5.70°
CSafe - Generic	11.30°	10.07°
PAWS [1]	38.7°	18.60°

8.2 Vehicle Localization

Table 4 compares the average direction of arrival localization error between CSafe, CSafe - generic, and PAWS in degrees. We see that the average error rate of PAWS is much higher than that of both modes of CSafe despite PAWS using a localization algorithm with more granularity. This is because PAWS uses a cross-correlation based method to estimate relative delays between microphones in its array. As mentioned in Section 4.2.1 these methods are only capable of capturing a single source in the environment. Most of the time, the source that is captured is the loudest sound in the environment as the sound with the highest energy influences the values of the cross-correlation function the greatest. Since the construction tool is often the loudest sound in the environment, rather than the approaching vehicle, PAWS will tend to localize the sound of the construction tool rather than the vehicle, leading to high localization errors. On the other hand, CSafe's novel adaptive noise filtering architecture is able to filter out most construction sounds and localize multiple targets in the environment. This reduces the effect of the overpowering construction noises in the environment on the detection and localization of oncoming vehicles, which leads to higher localization accuracy.

9 CONCLUSION

We present CSafe, a low-power, wearable, audio safety platform for construction worker safety. CSafe uses an array of low-power microphones integrated into an embedded hardware platform along with an accompanying smartphone system to detect + localize oncoming vehicles and provide alerts to users. The key difference between construction worker safety and general urban or pedestrian safety is the presence of construction noises that are often orders of magnitude louder than that of oncoming vehicles, greatly reducing the effectiveness of general audio-based urban safety platforms. To address this challenge, we introduce a novel noise filtering called Probabilistic Template Matching (PTM) that is integrated into a novel adaptive noise filtering architecture that leverages both single-channel and multi-channel source separation techniques with a feedback loop to more robustly filter out common overpowering construction and urban noises from the environment. We show that by applying our novel filtering architecture, we can improve our vehicle detector's detection rate by more than 10% compared with no filtering and other state-of-art source separation algorithms. Finally, we show through a series of real-world experiments that CSafe improves upon vehicle detection rate by up to 16% and reduces localization error by almost $30°$ in noisy construction environments over other state-of-art audio safety systems.

ACKNOWLEDGMENTS

This research was partially supported by the National Science Foundation under Grant Numbers CNS-1704899, CNS-1815274, CNS-11943396, and CNS-1837022. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Columbia University, NSF, or the U.S. Government or any of its agencies.

REFERENCES

[1] Stephen Xia, Daniel de Godoy Peixoto, Bashima Islam, Md Tamzeed Islam, Shahriar Nirjon, Peter R Kinget, and Xiaofan Jiang. Improving pedestrian safety in cities using intelligent wearable systems. *IEEE Internet of Things Journal*, 6(5):7497–7514, 2019.

[2] Governor's Highway Safety Association. Pedestrian traffic fatalities by state: 2019 preliminary data. https://www.ghsa.org/resources/Pedestrians20, February 2020.

[3] The National Institute for Occupational Safety and Health. Motor vehicle crash facts. https://www.cdc.gov/niosh/motorvehicle/resources/crashdata/facts.html, September 2019. [Online].

[4] P. Ho and J. Chen. Wisafe: Wi-fi pedestrian collision avoidance system. *IEEE Transactions on Vehicular Technology*, 66(6):4564–4578, June 2017.

[5] Myounggyu Won, Aawesh Shrestha, and Yongsoon Eun. Enabling wifi p2p-based pedestrian safety app, 2018.

[6] Tianyu Wang, Giuseppe Cardone, Antonio Corradi, Lorenzo Torresani, and Andrew T. Campbell. Walksafe: A pedestrian safety app for mobile phone users who walk and talk while crossing roads. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems; Applications*, HotMobile '12. ACM, 2012.

[7] Shengyan Zhou, Jianwei Gong, Guangming Xiong, Huiyan Chen, and Karl Iagnemma. Road detection using support vector machine based on online learning and evaluation. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010.

[8] Shyr-Long Jeng, Wei-Hua Chieng, and Hsiang-Pin Lu. Estimating speed using a side-looking single-radar vehicle detector. *Intelligent Transportation Systems, IEEE Transactions on*, 15(2), 2014.

[9] Jean-Marc Valin, François Michaud, and Jean Rouat. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robotics and Autonomous Systems*, 55(3), 2007.

[10] Pradeep K Atrey, Namunu C Maddage, and Mohan S Kankanhalli. Audio based event detection for multimedia surveillance. In *Acoustics, Speech and Signal Processing, 2006. Proceedings. International Conference on*. IEEE, 2006.

[11] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. Acoustic event detection in real life recordings. In *Signal Processing Conference, 2010 18th European*. IEEE, 2010.

[12] Stephen Xia and Xiaofan Jiang. Pams: Improving privacy in audio-based mobile systems. In *Proceedings of the 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet*

of Things, pages 41–47, 2020.

[13] Shahriar Nirjon, Robert Dickerson, John Stankovic, Guobin Shen, and Xiaofan Jiang. smfcc: exploiting sparseness in speech for fast acoustic feature extraction on mobile devices–a feasibility study. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, pages 1–6, 2013.

[14] Dezhi Hong, Ben Zhang, Qiang Li, Shahriar Nirjon, Robert Dickerson, Guobin Shen, Xiaofan Jiang, and John A Stankovic. Demo abstract: Septimu—continuous in-situ human wellness monitoring and feedback using sensors embedded in earphones. In *2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 159–160. IEEE, 2012.

[15] Sugang Li, Xiaoran Fan, Yanyong Zhang, Wade Trappe, Janne Lindqvist, and Richard E. Howard. Auto++: Detecting cars using embedded microphones in real-time. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3), September 2017.

[16] M. Won, A. Shrestha, K. Park, and Y. Eun. Safercross: Enhancing pedestrian safety using embedded sensors of smartphone. *IEEE Access*, 8:49657–49670, 2020.

[17] Stephen Xia, Daniel de Godoy, Bashima Islam, Md Tamzeed Islam, Shahriar Nirjon, Peter R Kinget, and Xiaofan Jiang. A smartphone-based system for improving pedestrian safety. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–2. IEEE, 2018.

[18] Daniel de Godoy, Xiaofan Jiang, and Peter R Kinget. A 78.2 nw 3-channel time-delay-to-digital converter using polarity coincidence for audio-based object localization. In *2018 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–5. IEEE, 2018.

[19] Daniel de Godoy, Bashima Islam, Stephen Xia, Md Tamzeed Islam, Rishikanth Chandrasekaran, Yen-Chun Chen, Shahriar Nirjon, Peter R Kinget, and Xiaofan Jiang. Paws: A wearable acoustic system for pedestrian safety. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 237–248. IEEE, 2018.

[20] Rishikanth Chandrasekaran, Daniel de Godoy, Stephen Xia, Md Tamzeed Islam, Bashima Islam, Shahriar Nirjon, Peter Kinget, and Xiaofan Jiang. Seus: A wearable multi-channel acoustic headset platform to improve pedestrian safety: Demo abstract. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 330–331, 2016.

[21] Wonil Lee, Edmund Seto, Ken-Yu Lin, and Giovanni C Migliaccio. An evaluation of wearable sensors and their placements for analyzing construction worker's trunk posture in laboratory conditions. *Applied ergonomics*, 65:424—436, November 2017.

[22] Sungjoo Hwang and SangHyun Lee. Wristband-type wearable health devices to measure construction workers' physical demands. *Automation in Construction*, 83:330 – 340, 2017.

[23] Ergodyne. Dry evaporative cooling vest. https://www.ergodyne.com/chill-its-6685-dry-evaporative-cooling-vest.html. Accessed: 2020-10-16.

[24] Triax Technologies. Manage your workplace safety with a connected jobsite. https://www.triaxtec.com/. Accessed: 2020-10-16.

[25] Ibukun Awolusi, Eric Marks, and Matthew Hallowell. Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices. *Automation in Construction*, 85:96 – 106, 2018.

[26] Anastasios Alexandridis, Anthony Griffin, and Athanasios Mouchtaris. Capturing and reproducing spatial audio based on a circular microphone array. *JECE*, 2013, January 2013.

[27] O. L. Frost. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 60(8):926–935, 1972.

[28] L. Griffiths and C. Jim. An alternative approach to linearly constrained adaptive beamforming. *IEEE Transactions on Antennas and Propagation*, 30(1):27–34, 1982.

[29] A. V. Oppenheim, E. Weinstein, K. C. Zangi, M. Feder, and D. Gauger. Single-sensor active noise cancellation. *IEEE Transactions on Speech and Audio Processing*, 2(2):285–290, 1994.

[30] Jordan Cheer and Stephen J. Elliott. Multichannel control systems for the attenuation of interior road noise in vehicles. *Mechanical Systems and Signal Processing*, 60-61:753 – 769, 2015.

[31] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411 – 430, 2000.

[32] Siow Yong Low, S. Nordholm, and R. Togneri. Convolutive blind signal separation with post-processing. *IEEE Transactions on Speech and Audio Processing*, 12(5):539–548, 2004.

[33] Michael Syskind Pedersen, Jan Larsen, Ulrik Kjems, and Lucas C. Parra. A survey of convolutive blind source separation methods, 2007.

[34] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):191–199, 2006.

[35] E. M. Grais and H. Erdogan. Single channel speech music separation using nonnegative matrix factorization and spectral masks. In *2011 17th International Conference on Digital Signal Processing (DSP)*, pages 1–6, 2011.

[36] N. Takahashi, N. Goswami, and Y. Mitsufuji. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 106–110, 2018.

[37] Jalal Taghia and Jalil Taghia. One-channel audio source separation of convolutive mixture. In Tarek Sobh, editor, *Advances in Computer and Information Sciences and Engineering*, pages 202–206, Dordrecht, 2008. Springer Netherlands.

[38] P. Enqvist and J. Karlsson. Minimal itakura-saito distance and covariance interpolation. In *2008 47th IEEE Conference on Decision and Control*, pages 137–142, 2008.

[39] P. Gupta and S. P. Kar. Music and improved music algorithm to estimate direction of arrival. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 0757–0761, 2015.

[40] R. Roy and T. Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989.

[41] Scott Rickard. *The DUET Blind Source Separation Algorithm*, pages 217–241. Springer Netherlands, Dordrecht, 2007.

[42] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[43] L. L. DU, L. M. Wang, and Z. Zhuang. [Measurement and analysis of human head-face dimensions]. *Zhonghua Lao Dong Wei Sheng Zhi Ye Bing Za Zhi*, 26(5):266–270, May 2008.

[44] Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(3):708–721, 2009.