Writing 4: Final SIBR Grant Application

Connor Kutz, Cole Weinhauer

Project Summary

Box 1: Overview, Key Words, and Subtopic Name

The desired outcome of the proposed activity will be an android application that can be used to act as an interface between users wearing headphones and incoming audio stimuli. Specific goals in terms of functionality include alerting users about upcoming heavy traffic and crime using GPS location services, changing the volume levels of your device based both on general and specific audio, like your name or screaming, honking, and other things of the nature, and finally creating customized profiles for specific locations so users can have preset conditions for the application settings that will activate whenever they enter a preset radius of that said location. Keywords and phrases include audio processing, audio filtering, machine learning, application development, and android mobile systems. The subtopic name is IT11. Human-Computer Interaction; Virtual Reality; Augmented Reality.
Keywords: Pedestrian Safety, Machine Learning, Audio Processing, Speech Detection, NLP, Android

Box 2: Intellectual Merit

This Small Business Innovation Research Phase I project will provide insight into the capabilities of smartphone sensors and the ability integrate headphone users into their environment. The overall plan for our application is to combine system applications, like microphones and location services, and machine-learning based algorithms to 1) detect and classify real world sounds and 2) give alerts to the user based on the sound. To do this, we created a python-based keras neural network model using an urban sounds dataset and then deploying our trained model onto our application. Active volume management was also implemented via a separate, perpetual service used to monitor and intelligently adjust the sound level sent to the user's audio output.

Box 3: Broader/Commercial Impact

The Application will fill a societal need: it will provide a system to warn pedestrians who are listening to music about various possible hazards, including crime and traffic. Many people, especially teens and young people, become trapped in a “bubble” while listening to music while acting as pedestrians in lively cities. In a study published in *Injury Prevention*, it was found that that both the number of pedestrians killed while listening to music has risen substantially and that in nearly 70% of the cases, the accident was fatal. Thus, the monetization route could be as simple as being a paid app on the app store. Targeted consumers could include parents who pay for it to run on their children's devices as well as young people who live in urban areas with large traffic.

Elevator Pitch

Our innovation will help change how pedestrians experience the world around them. Did you know that in a study published in *Injury Prevention*, it was found that that both the number of pedestrians killed while listening to music has nearly tripled in the years from 2004 and 2011, and that in nearly 70% of the cases, the accident was fatal. So, as people become more and more enthralled with their virtual worlds, we need a way for pedestrians to be able to know about possible hazards. Using a combination of sound processing, text-to-speech, and location services, we intend to create an app that provides information about traffic, crime rates, and even impending accidents to help its users stay safe even when listening to their favorite songs!

Users who purchase our app will get traffic, crime, and weather updates precautionarily when they enter a new area, as well as the ability to have your music turn down or off and get audio alerts in specific situations. The app will be responsive to cars honking and other stimuli that could be indicative of a future accident and give an audio alert regarding it. It will also be able to recognize the users specific name so that if someone addresses the user, the app will alert him or her and turn down music if it’s playing. These features set our app apart because there is no other app with this functionality.

There are 2 major consumer demographics we expect. Firstly, pedestrians in cities that have a large amount of commuters and traffic. Because of the quickness of traffic in these areas, we expect to need to be able to respond quickly to possible auto accident stimulus. The other major demographic is parents who will buy this product for their children to help bring them peace of mind about their children commuting, to and from school, sports practice, etc.

Commercial Opportunity and Societal Impact

The Commercial Opportunity

The market of our application is the subset of mobile device users who walk outdoors with music playing in their headphones. This is quite a large group of users. In a 2009 survey, The National Household Travel Survey estimated that there were around 40.9 billion walking trips done each year, so many people could benefit from this app and thus we can consider them in our addressable market. Specifically, in big cities there is a very large market. It was found that over 355,000 people enter Times Square daily as pedestrians. Obviously a few things drive the market. One could be gas prices: if they were lower, then we would see a decrease in pedestrians using the app (because there would be less pedestrians) and the opposite if the prices went up. Similarly, we could consider the temperature and weather as a driver in the same way: colder, rainier, and generally worse weather causes people to walk less, and nicer, more enticing weather will make people walk more.

There are 3 major consumer demographics we expect to be interested in such an app. Firstly, pedestrians in cities that have a large amount of commuters and traffic. Due to the quickness of traffic in these areas, we expect to need to be able to respond quickly to possible auto accident stimulus. Similarly, due to typically higher amounts of traffic and crime in these areas, finding more up to date data in these areas is integral for the success of the app for these users. Another major demographic is parents who will buy this product for their children to help bring them peace of mind about their children commuting, to and from school, sports practice, etc. This requires that the user interface to be simple and easy to use. A final user demographic we are anticipating are blind/ visually-impaired people. Obviously, the reasons why they would use the application is trivial. Because of this we need audio instructions and cues to describe where certain buttons are located on the app.

As a business model, we plan to collect revenue from a one time app purchase. Since the app offers a function rather than entertainment, we think a one time payment makes the most sense. However, we also have the option to collect ad revenue with occasional advertisements upon startup of the app. Ads can be implemented visually and audibly as long as the ads don't have higher privilege than the audio messages alerting pedestrians. A third option we could look at is possibly trying to sell our application before it goes to stores to some larger company or entity. If this were the case, we could try to go to Android, the company itself, and see if they would be interested in purchasing it to be used a system application that comes pre-installed. Because of the way the app could be used to help the visually-impaired, it could be something that, if pre-installed, helps to swing consumers which product to buy.

Many apps that try to address pedestrian safety neither utilize the total, high processing functionality found in modern phones nor do they use sound processing as a way of predicting threats. Some use video processing to view for cars, some scan for bluetooth devices, or send alerts when users enter intersections, but very few focus on using sound and location to predict possible threats to the user. In terms of the competitive landscape, large scale improvement that would make our approach ineffective does not seem feasible. For such an improvement, a system would have to be developed that could actively track both vehicle drivers and pedestrians and makes inferences based on real-time behavior. On top of developing this complicated system, it would need to gain traction to be a viable solution to the problem and rival the functionality found in our app. The main difference between our app is it tries to combat both immediate threats (cars honking near you, hearing gunshots) as well as predicting them (crime alerts, weather and traffic updates)

In terms of actual revenue, we have to distinguish by plan how we intend to commercialize our product. Assuming we use just the app store purchasing plan, the revenue projections are very easy to estimate, it's just the revenue of a single app purchase multiplied by the number of users we expect. Obviously, we can do one of two things: we can make our app cost very little, such as the standard 99 cents seen on most apps today, to maximize our number of users who will buy it, or could maximize our app price so that even if we have a limited user base, we still can make considerable revenue. It is estimated that the first option would be more effective due to the market already having free options which are able to perform a fraction of the features found in our product, so it would be hard to convince a user to spend a fee that is significant for this product.

Societal and Global Impact

The broader societal need we are addressing is the alarming number of pedestrian deaths related to headphone use and distracted walking. In a study published by Injury Prevention, it was found that that the number of pedestrians killed while listening to music has nearly tripled in the years from 2004 and 2011, going from 16 to 47, and that in nearly 70% of the cases, the accident was fatal. This is due to the "bubble" that users get stuck in when they are both walking as well as typing on their phone, playing music, or scrolling and reading what's on your phone at that time, so we figure the best way to limit these deaths is to break this bubble by providing real-time audio alerts to the user.

If our product was widely used, it would result in a large decrease in pedestrian deaths due to distracted walking. As the number of pedestrian deaths continues to increase, a product like ours is becoming more and more of a necessity. Specifically, we would see decreases in pedestrian deaths of young commuters, our target demographic, since they are the most likely users of this app. Additionally, we hope to see train-related deaths decline significantly. Since accidentally getting hit by a train accidentally is essentially due to distracted walking, our app would be extremely effective in preventing this behavior. Finally, if our app was used widely we'd also see a decrease in traffic accidents and related-deaths, which would means that less tax dollars and time are spent trying to combat these issues.

The app poses no issues in terms of health or environment, and has no risk of being used unethically, so long as it isn't hacked into. Should the app be compromised, there would be a risk of user data being accessed which would obviously be problematic. For this reason, we are taking many measures to so secure user and other aggregate data. It is important that this app be accessible by young children. Thus, we want the UI to be simple and very easy to read. We will match colors to indicate if certain weather, traffic, or crime levels are good and nonthreatening (green) or are very threatening (red), or even if they are between the two (yellow). Also, we will use very simple to understand audio cues to ensure that young users can understand them. Similarly, as we intend for users who are visually-impaired to use it, we need audio cues that can also describe the layout of the application and perhaps even a tutorial session which goes over the buttons and provides haptic feedback, like the phone vibrating, should be implemented for these types of users.

One of the global issues we could encounter is the app will only be in English. Thus, if you don't speak English, you won't be able to use the app. Similarly, that could make it hard to access if you live in another country, even if you speak English. However, implementation of language support for other countries would be exceedingly simple and the system would still work under the exact same protocol. Google’s locations services API will work on any country, and as long as crime and weather data could be gathered in these areas, the app could run anywhere and be used by anyone.

Technical Discussion and Plan

Design

Included below is an image representation of our application. For this section, we will go through each subsystem and talk about the design as well as its innovations

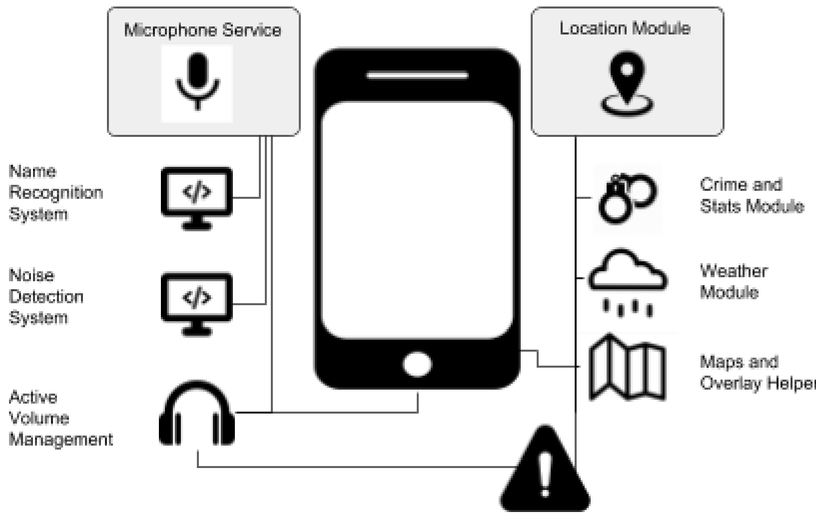

Figure 1: System Overview

Sound Recognition and Classification System: This system is in charge of developing the model we will use to classify our real-time sounds. the sound recognition system is a Python-based machine learning algorithm. The system utilizes the UrbanSounds8K Dataset as developed by J. Salamon, C. Jacoby and J. P. Bello. This Dataset supplies 3 things for each entry: a WAV file that is composed of the sound of the event, the metadata of that file, and a classification of the sound. From here, we will get the Mel-frequency cepstral coefficients (MFCCs). A Mel-frequency cepstrum is a representation of the short-term power spectrum of a sound, based on a cosine transformation of a log power spectrum on a nonlinear mel scale of frequency, and the MFCCs are the 40 values that are used to model the Mel-frequency Cepstrum. We will then use these values to train our model. The one we will be using is the sequential neural networks model provided by the Keras library. We will use a train-test split of 80-20 on our data to validate our model. Upon running the model and validating it with the test data, the model was accurate typically in the mid-70s percentage. After developing the model, we have to freeze it and turn it into a .pb file that can be deployed on our application. We developed a small-python script to do this. Finally, we have to create .so and .jar libraries for the model, which was done by executing a .sh script, and then put the libraries within our applications gradle. After this, we have a python-developed model we can invoke on our android device to classify sound.

Name Recognition System: This system's responsibilities are to 1) distinguish whether a sound is a user's name and 2) deciding what to do if it is/isn't the name. To do this, we intend to have a user enter in their name initially when first starting to use the system. We will compute the MFCCs as we did before, as well as use speech-to-text and create a list of all words that the system interprets the user as saying when we are configuring the name. Now, when a sound is encountered by the system, it will compute the MFCCs of it, get its speech-to-text result, and then pass them to the system, which will compare them to the archived values of the user's name and see if they are close enough to distinguish a match. If the system has a hard time deciding, we can use the fall-back of the speech-to-text results to decide if the user's name was said. In the instance it is found to be the user's name, we will lower the volume so the user can hear who called their name.

Active Volume Management System: One feature of our design is that audio levels are actively adjusted to match your environment. This means that when in a quiet environment, the media volume level will be a quieter setting. Then, if you go into an area with a high level of ambient noise your volume will increase to remain audible. As the majority of our users will use our product while listening to music, this feature acts as a convenience to ensure that media can be heard over the ambient noise level. This feature is also a safety design to ensure that users are able to hear the alerts announced by the app. This is all done by a persistent, threaded service which gathers average noise level data. When in increase or decrease in ambient noise is detected, volume output is adjusted accordingly.

Crime Alert and Data System: The crime and safety alerts feature is one of the most important pieces of our final product. This is the part of the app which warns the user of possible dangerous situations in their vicinity, for example, a personal robbery on a nearby street. The goal of this feature is to actively warn the user of dangerous crime incidents so they can take action to stay safe and avoid the possible suspect. Another aspect of this feature is to obtain and display crime density information via a MapView to preemptively show the user dangerous areas that should be avoided. These two pieces are implemented in different ways. First, for time sensitive and location based alerts, a polling service will check a database containing rapidly updated crime reports for relevant results. When there is an alert, it will be broadcasted via text-to-speech to the user through the audio output stream along with a short notification with extra details. For aggregate data of crime incidents and density, there is a service to perform a daily update and recalculation of the most dangerous areas to avoid. This is then shown in the map activity as a heat map.

Key Objectives

One big key objective our system is trying to accomplish is both the fast and correct classification of sounds. Essentially, our system gets most of its functionality from classifying sounds, so it will need to be accurate to have a successful system. Similarly, since we will try to classify every sound the microphone hears, we need to compute the MFCCs in computationally effective way.

Another such objective is a more broad one: can we effectively gather the sounds in the real-world environment. One immediate worry is that combining sounds with the general noise of cars, people, and other things will cause our sound recognition model to have problems classifying. Thus, during our next phase, we will spend time testing the system. If we have errors, the initial solution we could try is adding ambient noise to our urban sounds data and then getting the MFCCs so that it models better in the real world. Another solution we could attempt if this problem occurs is trying to remove the noise from the real-time audio we get, but this would be significantly harder.

Finally, we need to see if we can accurately and quickly get our crime updates as well. For our product to be successful in keeping its users safe, the crime alerts will need to be timely. In the real world, a delay of only 1 minute could put users in a dangerous situation, so the product will be vigorously tested to ensure that our service consistently and quickly performs as expected.

Technical Milestones

For this product to be successful, we need to overcome a few technological hurdles. One obstacle is gaining the ability to constantly and seamlessly listen to and analyze the user's surroundings. One of the challenges posed by this goal is gaining user permission for audio recording and processing. Often times, users are wary of allowing apps to use the phone's microphone so our reasoning must be clearly explained to the user upon requesting permission.

Similarly, having a constant service dedicated to audio processing runs the risk of being resource intensive, so it is important for us monitor CPU, memory, battery usage, etc. The only way for this app to be marketable is if it can be made to run in an unobtrusive and subtle way. We're already combating this by deploying our sound recognition model to the application after we have already trained it, and also precalculating MFCCs values.

Another milestone we need to hit is accurate classification of the real-time sounds. This is integral for our project as classifying the sounds the user hears provides most of the functionality of the application. Because of this, we want our model to correctly classify things 70% of the time, which we feel is a respectable amount and is below what our model is classifying now (without noise though, so that should cause a decrease in accuracy). Also, because of the nature of the application, we will try to have our false positives (saying there is a threat when there isn't) make up a much larger percentage of the inaccuracies as compared to our false negatives (giving no alert after a car honk, a gunshot, etc).

R&D Plan

September 2018

- Create Repository
- Develop initial feature test harness
- Create project webpage
- Find urban sounds dataset

October 2018

- Start a service with thread to use with Microphone
- Implement microphone libraries to activate constant listening
- Begin developing sound recognition system
- Set up android profiler to analyze resource usage

November 2018

- Implement location services to poll for user's current location
- Create a Map View for visualized crime, weather and, traffic
- Add ability to show data overlays to map
- Use service to actively adjust volume to compensate for ambient noise level
- Use android profiler to guarantee low resource intensivity
- use OpenData API to get be able to get http crime information
- Finish sound recognition model, deploy it to application

December 2018

- be able to receive time-sensitive crime updates
- be able to update and interface with user over media audio stream
- Create a service which polls for local and urgent crime updates
- Begin classifying real world sounds with model

January 2019

- Begin working on name recognition system
- Test sound recognition on outside audio
- Update/ retrain sound recognition model for improved results

February 2019

- Begin developing UI
- Look into using Firebase for user-authentication
- Fine tune sound recognizer, name recognizer
- Look at/ begin localized cache

March 2019

- Apply final tweaks to volume adjustment algorithm
- Finalize UI
- Begin working on server-side

April 2019

- Finalize server, Firebase authentication, and local cache

May 2019

- Clean up issues/ errors