Aware

Connor Kutz and Cole Weinhauer

School of Engineering and Applied Science
THE GEORGE WASHINGTON UNIVERSITY
Department of Computer Science

Problem Statement

In today's society, it has become commonplace to see pedestrians walking around city streets wearing headphones. This is a major distraction that can stop the wearer from hearing important noises which could save their lives.

Aware was designed to address and help to mitigate this problem distracted pedestrians.

What It Is

Aware is a multi-featured android app which should help keep pedestrians aware of their surroundings at all times.

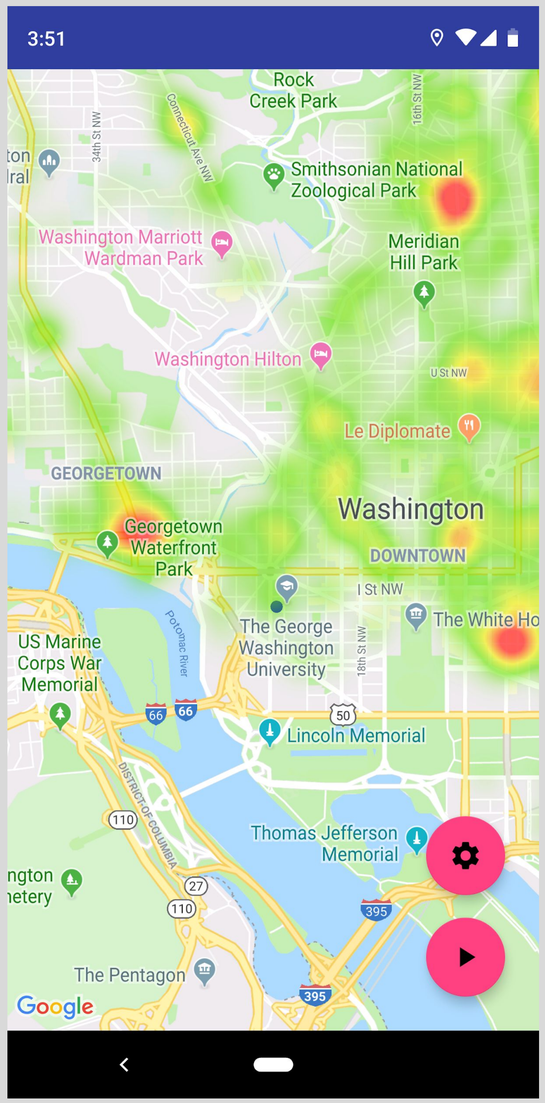
It does by providing the user with real time crime and noise recognition alerts. Aware takes advantage of location services to provide the user with alerts that are specifically relevant to them.

Interface Design

Aware's main components are separated into two parts. Using a Location Module and a Microphone Service, we are able to accomplish all of our functionality, including Noise Detection, Volume Management, and Crime Alerts.

Crime Density Service

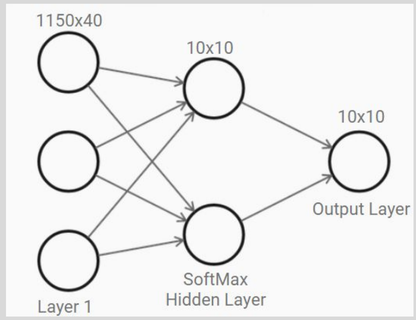
By utilizing DC's OpenData API, Aware can gather data from the last 30 days of crime incidents in the District of Columbia. With that data, a density heat map is displayed to the user.



Optionally, the user can decide to have alerts enabled so they are warned when walking into an an area with high crime density.

Noise Recognition Service

We designed our noise detection system to get real time classifications on Urban sounds. Our model was developed on Python using the Keras sequential model and achieved over 90% accuracy on an 80-20 training-testing split. We then use Tensorflow to deploy the model to Android, and we feed the microphone data into it to get the classification of sounds. We then alert the user if the system hears any of the following 3 sounds: Car honks, gun shots, and emergency sirens.



Alert Network

Using noise recognition alerts and a Firebase Realtime Database, alerts are sent to the cloud for processing and dispatch. If a dangerous noise is detected, for example, gunshots, other users within a predefined radius can be warned accordingly.

Since there is no need for human intervention, this is all done in real time. That makes Aware much faster than GW Alerts and other crime services.

Name Recognition

For the name recognition system, we ask the user to say their name when they first load up the app, and we use the Google text to speech services to get the name as text and save it on the app. Next, we feed the microphone recordings into the same text to speech service and if it returns with the users name, we alert the user.

Conclusions and Future Work

The crime hotspot identification system, the active crime alert system, and the active volume management all work. We are currently finalizing the noise dectionion system. We are still trying to find a reliable source for crime alerts, and the name recognition system has yet to be completed

One idea for future work is to finish implementation of the name recognition system, which which could be easily completed using Google's Text-to-Speech service. Another option is to add improvements the system for visually impaired users. This would allow Aware to also provide alerts for the blind

Major References

Android Studio Reference Documentation - https://developer.android.com/docs

Firebase Reference Documentation - https://firebase.google.com/docs/android/setup

Google Maps SDK Documentation - https://developers.google.com/maps/documentation/android-sdk/intro

Tensorflow API Reference Documentation - https://www.tensorflow.org/api_docs