Low Cost Gunshot Detection using Deep Learning on the Raspberry Pi

Alex Morehead[1], Lauren Ogden[2], Gabe Magee[3], Ryan Hosler[4], Bruce White[5], George Mohler[4]
[1]Computer Science, Mathematics, & Physics Department, Missouri Western State University; [2]Computer Science Department, Columbia University;
[3]Computer Science Department, Pomona College; [4]Computer & Info. Science Department, IUPUI
[5]AstroSensor.com

Introduction

Many cities using gunshot detection technology depend on expensive systems that ultimately rely on humans differentiating between gunshots and non-gunshots, such as ShotSpotter. Thus, a scalable gunshot detection system that is low in cost and high in accuracy would be advantageous for a variety of cities across the globe, in that it would favorably promote the delegation of tasks typically worked by humans to machines.

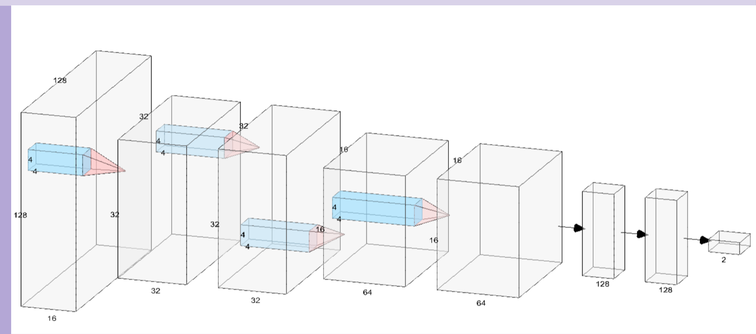
The Data

We obtained our data from two places: free internet databases such as Freesound and SoundBible and a repository of sounds recorded using a microphone connected to a Raspberry Pi microcomputer. In addition to this, we used a generative adversarial network (GAN) as well as sound augmentations to create additional samples of gunfire sounds and to prevent our models from overfitting to our compiled dataset.

Time Shift	Pitch Change	Speed Change	Volume Change	Background Noise Addition
Shifts a sound sample to the left or right by a randomly chosen amount less than 50% of the length and then fills in silence as needed.	Changes the pitch of a sample by a randomly chosen factor between 70% and 130%.	Alters the playback speed of a sample by a randomly chosen amount between 70% and 130%.	Decreases the amplitude of a sample according to a uniformly random variable.	Introduces random background noise into a sample while making sure that no gunshots are added into a sample that does not originally contain gunshots.

CNNs

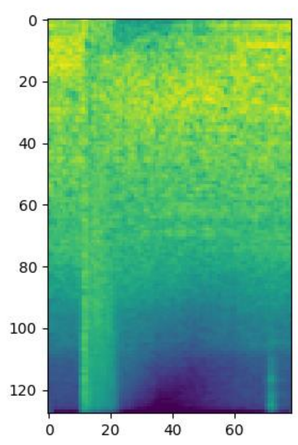
Convolutional Neural Networks (CNNs) are neural networks designed to locate, model, and accurately predict patterns present in input data such as a colored image.

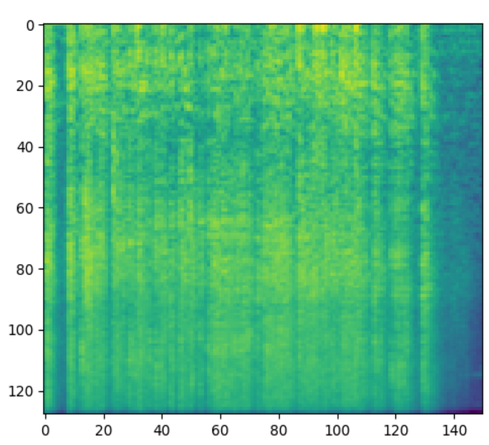


They do so by iteratively sliding over small regions of data and translating any inherent properties in a region over to a proceeding network layer. This process is repeated up until the output layer which generates a prediction.

Spectrograms

Spectrograms are visual representations of the frequency and amplitude of sound over a specified span of time. For our project, we created three models, a 1D architecture that looks at sound represented as an array of frequency values and two 2D architectures that instead analyze sound represented as spectrograms. For the time series model, each entry simply corresponds to a frequency measurement in a time series, whereas for the spectrogram model the entry for each specific frequency-time cartesian coordinate is an amplitude value.





Methodology

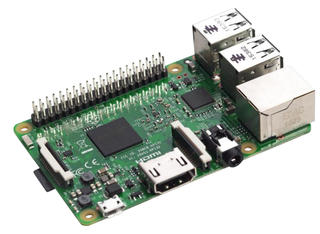
Development Process

Training

Three CNNs were trained on a variety of sound data to recognize gunshots. While we had labels for sounds other than gunshots, we grouped them into a singular group “other”. Then each of these samples were preprocessed to turn them into spectrograms if a given model was trained on them. The models were trained for 100 epochs or until the target metric of a training session, accuracy in this case, did not change for fifteen epochs.

Deployment

Each model was then deployed to a Raspberry Pi 3 Model B+ with an SMS modem attached. The models were loaded in TensorFlow Lite (TFLite) format as opposed to hierarchical data format (H5) for memory and performance concerns. Our pipeline has three active threads – one to put audio from a stream onto a queue, one to analyze sound data pulled from said queue, and one to send an SMS alert message to a predetermined list of phone numbers.



Gunshot Detection Results

We found that all our models performed well on a validation set. The best model, however, was found to be a combination of using all three of our models together by implementing a majority-rules algorithm which dispatches alerts if at least two models positively identified a gunshot sound.

	1D Convolutional Neural Network (44100 x 1)	2D Convolutional Neural Network (128 x 64)	2D Convolutional Neural Network (128 x 128)	Convolutional Neural Network Ensemble
Accuracy	99.4%	99.4%	99.4%	99.5%
Precision	98.0%	97.1%	97.4%	97.9%
Recall	96.6%	97.6%	97.6%	98.0%
F1 Score	97.3%	97.4%	97.5%	97.9%
AUC	98.2%	98.6%	98.6%	98.9%
IoU	94.7%	94.9%	95.1%	96.0%

Significance & Future Work

The findings generated by this research project add to the current base of knowledge regarding sound-based applications of CNNs and simultaneously provide insight into a compelling, new avenue of public safety measures. Ideally, a feature we would next like to implement in our pipeline would allow for a robust localization of gunshots using a group of three or more Raspberry Pi units positioned in proximity to each other to accommodate triangulation for discerning where a gunshot might have occurred.

Acknowledgements

We gratefully acknowledge the support of NSF grants REU-1659488, SCC-1737585, and ATD-1737996 for funding this summer research project.