
Audio-Visual Fusion Assisted Disaster Rescue

Duggan, Will

dwduggan@mail.sc.edu

Lewis, Cole

lewiscg@mail.sc.edu

Mallari, Daniella

dmallari@mail.sc.edu

December 2, 2022

Abstract

As the Earth's climate changes, our species is experiencing a rapid increase of volatile and dangerous weather events that often tragically conclude in disaster where a great loss or injury of life and property is experienced. This is happening more frequently in more areas now than ever before. We are attempting to solve the problem of how best to effectively navigate disaster-stricken environments safely, effectively, and quickly to determine where rescue resources should be allocated most efficiently towards locating and saving victims. To meet this challenge, we developed a system that implements fusion of both a visual, human-detection model and an audial, human-generated sound-detection model to determine potential victims within the immediate surroundings. The results at this stage of development proved to be a foundation upon which future work can be done, however, are underwhelming given available hardware, time constraints, and naive method of sound source localization.

and visual inputs from a microphone and camera respectively in order to make positional inferences. This is done through having our human-generated sound-detection model check detection results from our computer-vision, human-detection model to better make inferences as to where a potential disaster rescue victim may be located relative to the microphone and camera apparatus. Directional inference is then performed if our sound detection model determines that it detected somebody. The loudest average input channel is then found and mapped to a cardinal direction before being written to standard output.

1.2. Broader Impact

While currently ineffective, our project contributes towards providing better ways to engage in rescue efforts as our world becomes increasingly subject to calamitous events. Given more time, funding, access to hardware, and room to learn, we feel as if our project could be of real use in helping save lives in the wake of disaster. By providing human or autonomous operators a means by which to extend sensory-based rescue assistance from anywhere in the world, the research done here joins, in spirit, a body of scientific work dedicated towards bettering humanitarian endeavors now and into the future.

1. Introduction

1.1. Problem Statement

The problem at hand that we are attempting to solve is how to safely, efficiently, and quickly locate disaster victims within volatile and dynamic environments after a disaster has occurred. Conditions such as building or infrastructure collapse, immobilization or injury, as well as other obstacles that may impede camera input such as smoke or dust all contribute towards the need for a holistic, "multi-sensory" approach that requires multiple forms of input in the search for victims. We set out to accomplish this in our project by fostering within our system a dependence upon both audio

2. Contextual Work

3. Data

Given that there are two different models, two different sets of data are used. The visual model is pre-trained on an object-detection dataset with labeled people. The data in this dataset are labeled images. The labels included the class of the object as well as bounding-box labels for where the person is in the frame. Before being passed to the model, the pixel values of each frame are standardized to a scale between 0 to 1. A standardized scale during training of the

model allows the model to converge faster. The sound classification task used a combination of two different datasets, the ESC-50 [CITATION FOR DATASET] environmental sound classification dataset and some samples from the 8k Flickr audio corpus [CITATION FOR DATASET]. The ESC-50 dataset consisted of samples of 50 different semantic classes ranging from cow, to mouse clicking, to helicopter. 10 of these classes were considered “human, non-speech sounds” ESC-50. Each class contained 40 recordings, 5 seconds in length and sampled at 16kHz. Using these samples meant that there were 400 examples of non-speech human sounds and 1600 samples of non-human environment sounds. To create a dataset with two equally weighted classes, examples from the 8k Flickr audio corpus were added to the human sounds class so that both classes had 1600 samples. The 8k Flickr audio consists of recordings of speakers captioning multiple images verbally with no particular length, sampled at 16kHz. The data for the sound classification model was preprocessed before training. To ensure that there was no error in the lengths of the recordings, Librosa was used to load and resample the audio clips to a consistent 16kHz. The average length of the recordings was found to be 54000 samples. Librosa was again used to cut each recording to a consistent length of 64000 samples based on the average found. The samples were then converted to spectrograms to be used for classification in the sound classification model. The spectrograms were then converted to grayscale and resized to a height and width of 128 pixels. The purpose of this was to decrease the amount of information that would need to be processed by the network and to have consistent input sizes.

4. Methodology

Our project consists of two models which interact via a shared buffer of timestamps to determine if somebody was detected by the camera and our sound model simultaneously. In Figure 3 below, we outline and detail how each component works together before explaining how each component works individually.

4.1. Visual Recognition Model

Originally, we had sought to develop an object-detection model using YOLOv3 and TensorFlow that would be trained to recognize various human body parts, to ensure that human detection would not only be limited to instances in which a full body shot can be taken. We chose to use the YOLOv3 algorithm because it is well implemented and well documented. As time went on and we experimented with training runs of the model, we realized that training that model would be a challenge. One reason for this was because our group had to use an existing implementation of the model. YOLOv3 is built off of a Darknet backbone,

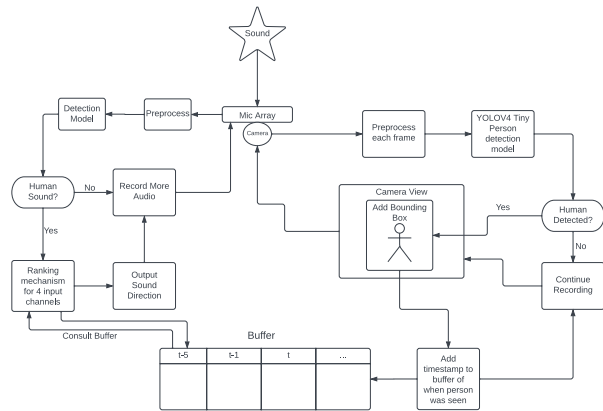


Figure 1. High level overview of our entire system and how each model is integrated with each other.

which is implemented in C++. This made re-implementing the model a challenge for our group, so we chose to use an existing implementation. This came with its own set of challenges though, as if we wanted to retrain the model, we would have to change our data to fit with the implementation. YOLOv3 implementations are also a couple of years old, meaning that there were also many bugs that needed to be fixed from libraries that had changed over time. After fixing the bugs and getting the training to run, we had issues finding hardware that could train the model within a reasonable time. Training on a laptop took approximately 12 hours for 50 epochs, which was far slower than needed. Additionally, many training runs would need to be performed with different hyperparameters to find an optimal model. When moving to a cloud GPU environment, Chameleon, to continue training with faster hardware, the size of the dataset and millions of parameters contained within the model proved yet again far too slow to train, taking approximately 15 hours to train each iteration with the provided GPU. After having trained the model a few times on Chameleon, we realized that the time spent struggling was not worth the possibility of a completely non-functional visual detection model. Our team found a pre-trained human detection YOLOv4-tiny model in our research [model link] that proved capable of consistently detecting people at a high enough speed that we felt confident it could stand up to the task. Having already been quantized and in TFLite format as well as able to detect people even when only viewing an arm or leg, this model accomplished what we had set out to do as well as assuage any fears of being limited by people being completely. This model has clear advantages over the model we were developing in that it is faster, able to make inferences at approximately 16-17 frames per second when using the PlayStation Eye and a laptop with a Radeon Pro 560X graphics processing unit, rather than the

approximately 11-12 frames per second previously. This performance gain is likely to prove even greater when deployed on hardware dedicated to running machine learning models.

5. Experiments

We subjected our system to a test consisting of 20 trials set in a near-silent environment in which a test subject spoke at a consistent distance of approximately one meter at varying cardinal directions to gauge both the accuracy of the sound-recognition model and our means of sound source localization. We ran the test for 20 trials using two distinct methods of sound source localization: inferring which direction the sound originated from by selecting the channel whose amplitude was the highest at any given point in the five-second interval, and selecting the channel who at any given point in the recording detected the highest amplitude. The results of this test are shown in Figure 2 below, displaying the accuracy of the entire system as a whole throughout the test on both sound source localization methods. Results were further broken down by the accuracy of our sound model alone, determined by whether or not it could correctly classify the test subject's voice as human-generated or not, and are shown in Figure 3 below.

Unfortunately the sound model and means of sound source localization both performed poorly. However, there are many factors at play which we believe contributed to the overall performance of our system, further evidenced by a 74% accuracy rate of the sound model when making inferences over its test set.

The greatest consideration is that of the hardware upon which our system was implemented. The PlayStation Eye was designed in 2007 as an accessory to the PlayStation 3. Alongside a camera, the PlayStation Eye contains a 4-microphone, omnidirectional, linear array with a maximum sample rate of 16 kHz, far lower than the industry standard of even a CD, which stands at 44.1 kHz. Acoustic hardware is very nuanced and liable to be influenced by minute factors that require identical conditions to replicate; the training set for our sound model was carefully curated to provide an accurate benchmark for human and non-human sounds alike, likely with much better recording hardware and in more advantageous conditions. The size of the microphone array, only 76 mm in width, only gives each microphone approximately 19 mm of distance apart from each other when accounting for the size of the casing. The PlayStation Eye's limited size and relatively simple audio controller provides the ideal environment for each microphone to lose independence as sound input becomes muddled as to which microphone is uniquely detecting a specific sound. The performance of the system as a whole depends on the performance of each component, as direc-

tional inference fails without the accurate function of both the visual and audial model. The failure of a single model or detection cascades across the entire system and renders it ineffective.

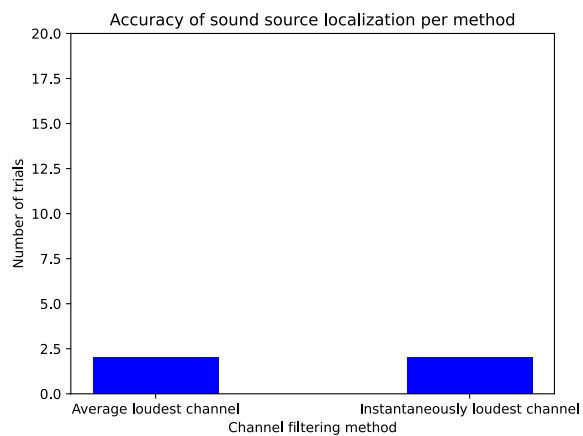


Figure 2. Sound source localization experiment. On the y-axis is the number of trials in which the system successfully detected which our test subject stood and spoke from, and on the x-axis is labeled which method was used.

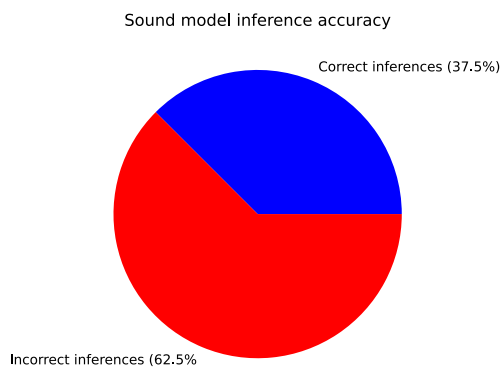


Figure 3. Sound source localization experiment. On the y-axis is the number of trials in which the system successfully detected which our test subject stood and spoke from, and on the x-axis is labeled which method was used.