WatchDog: Real-Time Gunshot Detection & Response

Satyarth Arora
sxa1654@miami.edu

Scott Pyle
scott.pyle@miami.edu

Matthew Sena
matthewsena@miami.edu

Zafiro V Paredes
zvp3@miami.edu

Department of Electrical and Computer Engineering
University of Miami
Coral Gables, Florida
May 8th, 2024

ABSTRACT

The increasing prevalence of shooting incidents in semi-open environments poses significant challenges for public safety and emergency response protocols. The WatchDog system is a novel solution designed to address these challenges by detecting and localizing gunshots in real-time. This research paper presents the development and evaluation of WatchDog, which utilizes audio data analysis through a sophisticated machine learning framework to provide rapid and accurate incident responses.

The system utilizes a combination of convolutional neural networks (CNNs) and custom models to process audio streams into spectrograms, which are then analyzed to detect the acoustic signatures of gunfire. Our models are trained on a diverse dataset of gunshot sounds, collected from various firearms and supplemented by environmental noise samples to enhance the system's robustness against false positives.

The architecture of WatchDog comprises multiple modules, including real-time audio processing, machine learning classification, and a user interface designed in Angular for efficient communication and interaction. The system's performance has been preliminarily tested, showing high accuracy and fast response times in gunshot detection, with the potential to substantially decrease the response time of first responders and increase the safety of individuals in affected areas.

Through this project, we demonstrate the application of advanced machine learning techniques and modern web technologies in creating a reliable and effective tool for enhancing public safety. The WatchDog system stands out for its ability to operate in real-time with high accuracy and minimal delay, proving its potential as a critical tool in emergency response efforts. Future work will focus on expanding the system's capabilities and improving its integration with existing emergency response protocols to provide a comprehensive solution for gunshot detection and response.

I. INTRODUCTION

In a study by the National Institute of Justice, there were a total of 170 shootings between 2023 and 2022 compared to the 12 shootings between 1966 and 1975 [1]. Similarly, there was an average of 51 deaths in shooting incidents between 2013 and 2019 compared to the average of 8 deaths in 1970 [2]. When evaluating the number of shooting incidents from 1966 and 2022, the trendline has a sharp exponential increase. In the data from the NIJ, 31% of shootings were in the workplace and 30% were in retail, bars, or restaurants [2]. Hence, around 60% of shooting incidents predominantly occurred in semi-open spaces that had groups of individuals. To address the issue of increasing shooting incidents in these environments, the WatchDog, a system specializing in detection and localization is created to assist in public safety and first responder protocols.

II. PROJECT PLANNING

The problem of shooting incidents in semi-open environments can be mitigated with a system that isolates and identifies the source of a gunshot.

A. Objectives

Some objectives to accomplish the main solution of developing the system are focused on survival, localization, response, and identification. Firstly, the system aims to increase the likelihood of survival for individuals involved in a shooting incident and reduce the number of victims. The quick and accurate localization of the shooter is essential to detain them. To increase the likelihood of survival, the system will aid in the identification of the perpetrator with the localization and reduce the overall response time needed to detain the perpetrator. Finally, the system will identify the location of the shooting find the type of weapon being used for logging an incident report and potentially find a path for individuals near the shooting to avoid while searching for safety.

B. Target User and Stakeholders

This system is intended for the use of any organization with a semi-open campus or environment which may be target of a mass shooter. Some stakeholders that need to be considered to succeed in reducing the number of casualties and victims of mass shootings are those involved in the semi-open campus or environment and those responsible for maintenance or safety

protocols and regulations.

The main stakeholders involved in the monitored location are the administration and support staff for the location. They have the responsibility to manage the system to ensure the functionality of the system, monitor the system as a part of their safety systems, and make the decision when a shooting as been detected to release the information to the first responders and the public in their communications. They would also have the responsibility of taking over the system to take control of the information it would report. Another set of stakeholders would be those most affected by a shooting incident, like the site's employees and any pedestrians or public on the campus. These individuals would not only be directly affected by the incident, but they would also be affected by any emergency policies and would desire real-time information while they seek safety. The media and communication systems of the environment are needed to distribute any available information during an incident.

Stakeholders that would maintain and additionally develop the system would be the developers, testers, and operators. They require consideration since they are essential to maintain the system to ensure maximum optimization and functionality and potentially reboot a system that failed or shut down. Compliance and financial officers are considered due to their responsibility to uphold legal and financial regulations for emergency responses. Finally, and most importantly, first responders would receive any information or report involving the incident that would assist in their response efforts. Additionally, their consideration is essential since this system has the potential of being implemented with any emergency protocols they may have in place,

C. Risk Analysis

Due to the nature of the system addressing such a sensitive matter, addressing any potential risks that could be brought up by the system is necessary to also implement solutions that would mitigate the risks. Four major risks that could be brought up by the system are misidentification, a false negative, mis location, and a system failure. Since the system has a detection aspect, the misidentification of a shooting incident has the potential of spreading a false alert causing panic and injury to the public. A way to avoid this issue a precautionary alert can be sent to an administrator of the organization for verification that there is a shooting incident. A false negative from the detection system that has failed to properly detect a gunshot is a risk on the other end of the spectrum that renders the system useless. To prevent such an issue from being present, the classification system can be optimized to fall within a reasonable threshold with little to non-false negative.

Mislocation of the shooting incident and the potential shooter is detrimental because the system is designed to work with first responders and their emergency protocols. Reporting incorrect information to the first responders can prevent them from neutralizing the perpetrator and getting pedestrians to safety. Therefore, a modular system with independent hardware designed for accuracy can provide a resilient system for if a unit fails, other units around it will be able to confirm or deny if there is an incident in the area. A system failure during a critical event is the most detrimental risk that could appear in the system since all safety aspects of the system, such as detection, surveillance, and localization are rendered useless. Deploying the system in a reliable could platform that is covered in multiple regions and zones can ensure functionality and a method for consistent safety surveillance.

III. TECHNICAL DESIGN

To complete the technical design of the system, the objectives are divided into components that will be comprised of the technical requirements needed to complete the deliverables.

A. Requirements

There are two sets of requirements for the system, non-functional and functional. Non-functional requirements focused on the stakeholders and the overall usage of the system and the functional requirements that are the expectations the system is required to meet technically.

Non-functional requirements consider the first responders, the administrators, and the employees that will be exposed to the system. For first responders, the system will be compliant with local laws and regulations, meet the operational requirements for first responders, and it will be able to integrate to the first-responder's operations, communication methods, and emergency procedures. For the organization's administrators and employees, there should be an ease of use and plenty of documentation for training and technical support for proper usage. The modular system should be quick to install and require low maintenance procedures. In regard to diversity and inclusion, there should be support and documentation in various languages.

Functional requirements for the system include the greatest amount of accuracy, proper and effective communication, and an efficient visualization of the information. For the detection and tracking sections of the project, a high accuracy shot detection for the gun type, incident location, and time is expected. Especially since the system depends on an accurate detection of a shooting incident to call for emergency services and protocols. Additionally, the system is expected to have an accurate tracking of the shooter with the modular hardware component and be able to deliver real-time performance. With real-time surveillance, the amount of time for more victims to be involved in a shooting incident is reduced.

For real time performance, the communication between the network and the hardware units is critical to be accurate. Therefore, it is expected that the system can communicate between the detection and information processes and the hardware that collects the information reliably to avoid any data distortion and misclassification or misidentification.

Finally, the section of the system that is seen by the user, the user interface, is expected to be able to effectively communicate the information collected by the hardware. This interface should be simple in the case of an emergency but display the proper amount of information to assist any emergency efforts. Furthermore, have the information displayed in a format understood by the public in the case it is released by the

organization.

B. Architectural Solution and Drivers

The architectural solution and design flow was designed to make the project scalable, reliable, and performance. The key factors influencing the architecture are the users for both the first responders and the public, the detection module, and the admin. WatchDog will utilize multiple architectures to accomplish these goals. As the project needs to have good performance and scalability, how the users receive the gunshot information as opposed to the program is important.

The event-driven architecture is perfect for the users then. All the recorded audio data is not pertinent to the users. Nor are any false-positives or any gunshots detected far away from them. The event-driven architecture employs a "publisher-subscriber" model for the users (the subscribers) to wait on the admin (the publisher) to send pertinent and important information to them. The admin is the middleman in all of this, set in between the users and the detection module. The admin utilizes the observer behavioral pattern. An observer directly works with an event-driven architecture to define the subscription mechanism to correctly notify the subscribers. In this case, it will be observing the detection module to look for it to send a positive gunshot detection through. The admin will then properly distribute this to the users as said before. Finally, the detection module will employ the opposite of the event-driven architecture, the pipe-filter architecture. This architecture will have a constant stream of information being poured through it. Here, the information is the audio data from the microphones.

These architectural solutions directly build up the system to be better scalable and reliable in the future. The event-driven architecture for the users makes sure the system and clients do not get bogged down and confused with information that does not pertain to them. While the pipe-filter architecture of the detection module is still crucial so there is constant observation and detection of any gunshots that occur in range of microphones.

C. General Design

The general design of the system consists of multiple modules, each processing and storing information to complete all the objectives. There is a detection module dedicated to processing and classifying the audio input to decide whether there is a shooting incident and the gun used. Other information, such as which module unit to find the general location and the time of the audio is sent to the front-end interface and the video management module. The video management module will take in the camera feed and send the data to the front-end after the detection module has detected a gunshot, providing a live video feed of the incident. The front-end will send all the available information and with the activation of the detection module, send all available information to the tracking module to produce a report. The front-end is essential to communicate all the information to all the stakeholders and complete all the use-cases for each type of user.

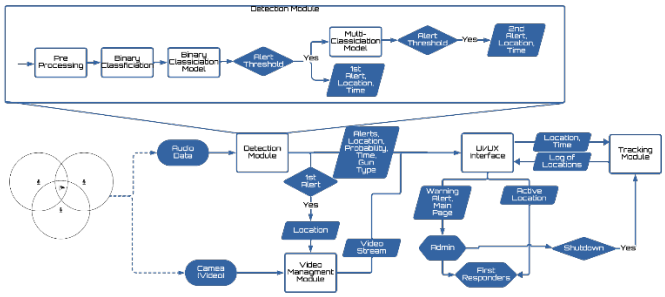


Figure 1:Overall Module Architecture Design

A use case diagram for the primary stakeholders can be seen in Figure 2: the administrators, first responders, and the public that includes employees and any visitors. Due to the privileges of the administrator, they have the capabilities to modify and monitor individual units, grant permissions to the other stakeholders, and have the final decision for the alerts to reach the first responders. First responders would then access any available information in the reports and monitor the map for surveillance and localization of the perpetrator. Finally, the public could be given access to the map through any method of communication for knowledge of where to avoid and where to seek shelter and get help.

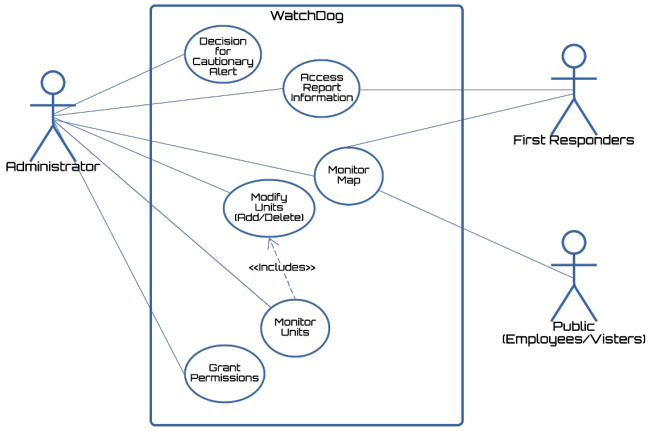


Figure 2: Use case diagram for main stakeholders.

The sequence diagram in Figure 3 shows the interactions of the main stakeholders with the WatchDog system. After WatchDog has detected a gun shot, the first alert is sent to the administrators for confirmation. After the second alert, the first responders are also sent the alert and provided any information in a report. Through the organization's communication efforts, the administrators will approve public access so the public can have the map view of where the incident is occurring.

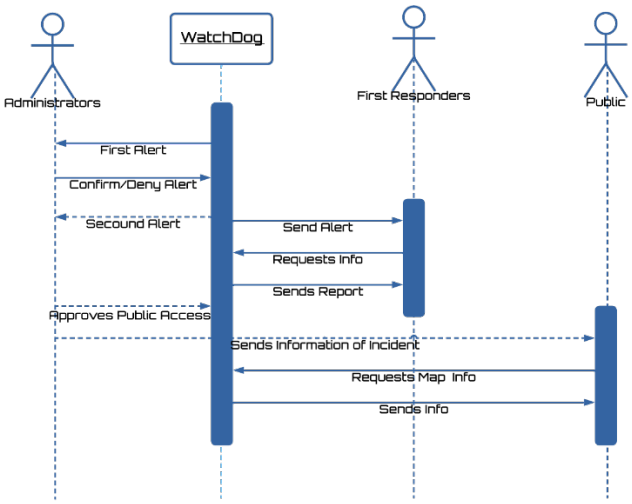


Figure 3: Sequence diagram when system detects a gunshot.

With a thorough analysis of the interactions between the stakeholders and the system as well as all the components necessary to complete the objectives, we can begin to develop each component.

IV. DEVELOPMENT & IMPLEMENTATION

Due to the constraints of this project, only three major components are built as a proof of concept: the detection module and the UI/UX interface. The detection module is composed of two components, the pre-processing of the audio input into spectrograms and the binary classification model deciding if it was a gun shot or not with the spectrograms. The output of the model is a possibility percentage sent to the front-end which will evaluate the number and present it appropriately. These three components produce a simple pipeline architecture from the initial stream-lined audio input to the spectrograms, to the possibility, to the front-end representation.

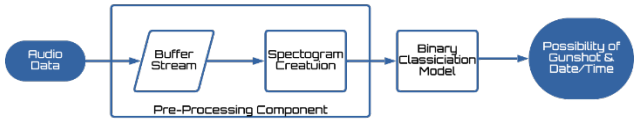


Figure 4:Archtecture of the detection module.

A. Pre-Processing

The pre-processing component of the WatchDog system is crucial for ensuring that audio data is properly prepared for gunshot detection. This section outlines the steps taken to transform incoming audio streams into a format suitable for analysis by the machine learning classification system.

a.) Audio Streaming and Packet Handling

Audio data is captured in real-time and streamed across a network in the form of small packets. As these packets arrive, they are immediately placed onto an initial queue designated for raw audio packets. This queue acts as a buffer, collecting audio data until there is enough content to form a more substantial audio segment.

b.) Audio Clip Assembly

Once sufficient packets are accumulated to create a 2-second audio clip, a process is triggered to stitch these packets together. This stitching process ensures that packets are aligned correctly to maintain the integrity of the sound, avoiding any gaps or overlaps that could affect the accuracy of the gunshot detection. The completed 2-second audio clip is then transferred to a secondary queue, where it awaits further processing.

c.) Spectrogram Computation

The next step in the pre-processing pipeline involves converting the 2-second audio clips into spectrograms. A spectrogram is a visual representation of the spectrum of frequencies in a sound as they vary with time, which is particularly useful for identifying distinct sounds such as gunshots within noisy environments (See appendix B for example spectrogram). This conversion is handled by a dedicated process that monitors the queue for new audio clips, processes each one to generate its spectrogram, and then removes the clip from the queue to maintain efficiency.

d.) Detection Model Queueing

Once a spectrogram is generated, it is placed onto another queue specifically for analysis by the detection model. This queue serves as a buffer that holds all pending spectrograms until they can be evaluated by the machine learning system.

e.) Inference and Thresholding

The final step in the pre-processing component is the inference process performed by the detection model. As each spectrogram reaches the front of the model queue, the detection model assesses it and outputs a score ranging from 0 to 1, representing the likelihood of the presence of a gunshot in the audio clip. If the score exceeds a predetermined threshold, it indicates that a gunshot has likely occurred, and the system proceeds with the appropriate alert protocols. Spectrograms with scores below this threshold are considered non-indicative of gunshots and are discarded.

The effectiveness of the pre-processing component is critical as it directly impacts the speed and accuracy of the gunshot detection capabilities of the WatchDog system. By efficiently managing audio streams, converting them into analyzable formats, and ensuring timely delivery to the detection model, this component helps maintain the system's responsiveness and reliability in critical situations.

B. Machine Learning Classification

The detection module's detection method was originally inspired by the CNN model used in combination with a Raspberry Pi to classify gunshots from other similar sounds [3], although this project introduces many notable improvements.

a.) Data Collection and Preprocessing

For the WatchDog system, our dataset includes 10,000 unique gunshot samples sourced from over 30 types of firearms.

These samples were aggregated from several public databases: [4], [5], and [6]. Our negative samples, crucial for training robust machine learning models, primarily consist of environmental sounds collected from [7] and [8], along with additional recordings made around our university campus.

Each audio file in our dataset is standardized into a 2-second clip at a 44,100 Hz sample rate and transformed into a 256x256 spectrogram. To enhance the model's ability to generalize across different audio conditions, we employ common audio augmentation techniques such as noise injection, time stretching, and pitch shifting.

b.) Network Architectures

We evaluated the performance of four convolutional neural network (CNN) models for this task (see appendix A for diagrams):

i. **SimpleCNN:** A baseline model with three convolutional layers.
ii. **ResNet18 and ResNet34:** These models are deeper, with residual connections to aid in training deeper networks without degradation.
iii. **EnhancedCNN:** Our custom model incorporates dilated convolutions, which expand the model's receptive field and are optimized for real-time performance with minimal computational resources.

c.) Training and Evaluation

All models were trained using the same data splits and hyperparameters to maintain consistency across evaluations. We utilized cross-validation to ensure the generalizability of our models. The models were evaluated based on their classification performance, ability to compute on-the-fly, and resource efficiency.

The training process involved several steps:

i. **Initialization:** Models were initialized and configured for training on the available computing device.
ii. **Data Loading:** Data loaders were prepared with batch processing and normalization.
iii. **Optimization:** An AdamW optimizer was used with a learning rate scheduler to adjust the learning rate over epochs.
iv. **Loss Functions:** We used binary cross-entropy loss for its effectiveness in binary classification tasks like gunshot detection.

This structured approach allows us to maintain a clear overview of the machine learning pipeline, ensuring that each component is optimized for the task of real-time gunshot detection. The modularity of our codebase also facilitates easy updates and scalability for future enhancements of the WatchDog system.

C. User Interface

Angular is a widely used front-end framework known for its robustness and flexibility in building modern web applications/interfaces. In this section, we delve into the interface components that constitute our Angular application, and how each plat a crucial role in enhancing user interactions and connectivity to our gunshot detection efforts.

a) Button Component

The button component enhances user interaction by providing visually appealing buttons with different styles, including success, danger, and warning. Evering Angular's template syntax and Bootstrap styling classes, the component ensures consistency and flexibility in button design. Custom logic enables click events to trigger specific actions within the application, enhancing user engagement and workflow efficiency. This design will allow command center/security personnel to signify if the alert from the binary classifier is a true or false indication of a gunshot to communicate with first responders.

b) Card List Component

The Card List component efficiently displays dynamic content in a structured format, enhancing the presentation of information within the application. Utilizing Angular's components architecture and data binding techniques, it dynamically populates card content based on user data or external sources. Event handlining mechanisms allow users to interact with individual cards, providing a seamless browsing experience. As the Binary classifier passes readings within gunshot range, threshold design will highlight unit cards with green, yellow, red indicating the level of threat. Utilizing our multi-class identifier will pass weapon types identified by our learning model.

c) Navbar Component

The Navbar component serves as a crucial element for navigation and user interface controls within the application. It incorporates touch event handling to ensure responsiveness across various devices, enhancing accessibility and usability. Custom styling and animations elevate the visual appeal of the navigation menu, providing an intuitive and engaging user experience.

d) Video Model Component

The Video Modal component facilitate the integration of video playback/live stream capabilities within the modal dialog, enriching multimedia content presentation within the application once linked with a camera/video streaming device that can be selected upon alert. Secure embedding of video content from external sources, and custom logic controls modal behavior, allowing users to interact with video content seamlessly within the application.

V. RESULTS & TESTING

Results are essential to ensure the functionality of WatchDog. Optimization results for the machine learning model are essential to stay within accuracy thresholds. Deployment results are essential to complete the proof of concept for a client receiving information as it is sent out by the detection module.

A. Machine Learning Models

The evaluation of our machine learning models reveals significant insights into their performance across various metrics crucial for the WatchDog system. These insights are drawn from a comparative analysis of loss curves during training and a detailed assessment of each model's performance after training.

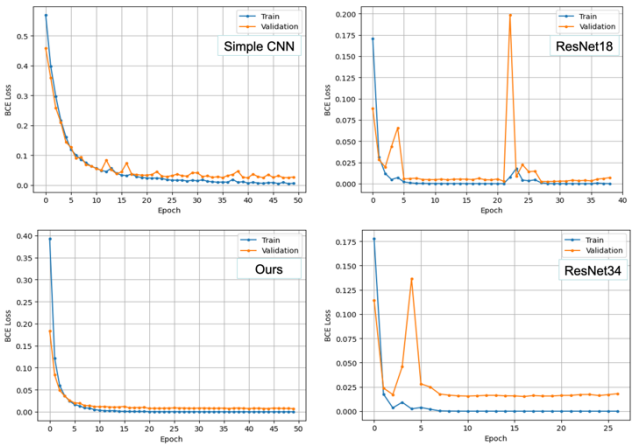


Figure 1: Training and validation loss while training for 50 epochs (or until loss divergence)

a.) Training and Validation Loss

As depicted in Figure 1, the training and validation loss curves across epochs for the four models—Simple CNN, ResNet18, ResNet34, and our custom network—illustrate the different learning dynamics of each. Our custom network shows a promising trajectory, characterized by rapid convergence and exceptional stability throughout the training process. Unlike the ResNet models, which display higher volatility and some divergence between training and validation loss, our network maintains a tight coupling between these metrics, indicating effective generalization without overfitting.

b.) Comparative Performance

The performance of the networks post-training, summarized in the table below, underscores the suitability of our custom network for the WatchDog system:

Table 1: Results of Network Architectures

Network	# Params	Test Accuracy	Test Loss (BCE)	F1-Score	Inference Rate*
Simple CNN	203K	98.901%	3.39e-2	0.9878	580.56 IPS
ResNet18	11M	99.773%	4.66e-3	0.9974	44.94 IPS
ResNet34	21M	99.774%	9.90e-3	0.9975	28.26 IPS
Our Network	**280K**	**99.899%**	**3.02e-3**	**0.9988**	**97.42 IPS**

**Average from 10,000 forward passes with gradient calculation disabled. Conducted on Apple M2 Ultra – 24 core CPU*

Despite having a fraction of the parameters compared to the deeper ResNet models, our custom network achieves the highest test accuracy and F1-score, alongside the lowest binary cross-entropy (BCE) loss. Its inference rate of 97.42 FPS, while lower than that of the Simple CNN, is considerably faster than that of the ResNet models and is well-suited for real-time applications where rapid response is critical.

c.) Inference Efficiency

The inference rate—measured in inferences per second (IPS)—is a critical metric for real-time systems like WatchDog. Our custom network offers a balanced profile of high accuracy and rapid inference, making it particularly effective for deployment in scenarios requiring immediate detection and response. This efficiency is achieved through the network's architectural design, which incorporates dilated convolutions to enhance receptive field size without the computational overhead typically associated with deeper networks.

The results from these tests confirm that our custom network is the best-suited model for the WatchDog system. It provides a superior balance between accuracy, computational efficiency, and real-time performance, crucial for the system's deployment in semi-open environments where quick and accurate detection of gunshots can significantly enhance public safety and emergency response.

B. Deployment

Deployment for the system was centered around the ability of the front-end to display the interface and receive any POST requests. Since the front-end was built with an Angular framework, an additional server program in JavaScript is created to connect the framework to access POST and GET requests.

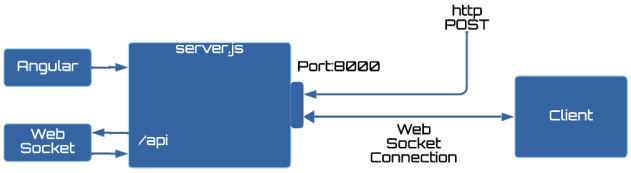


Figure 5:Front-end deployment additions.

The additional server.js program shown in the figure above will provide the routing for POST requests from the detection

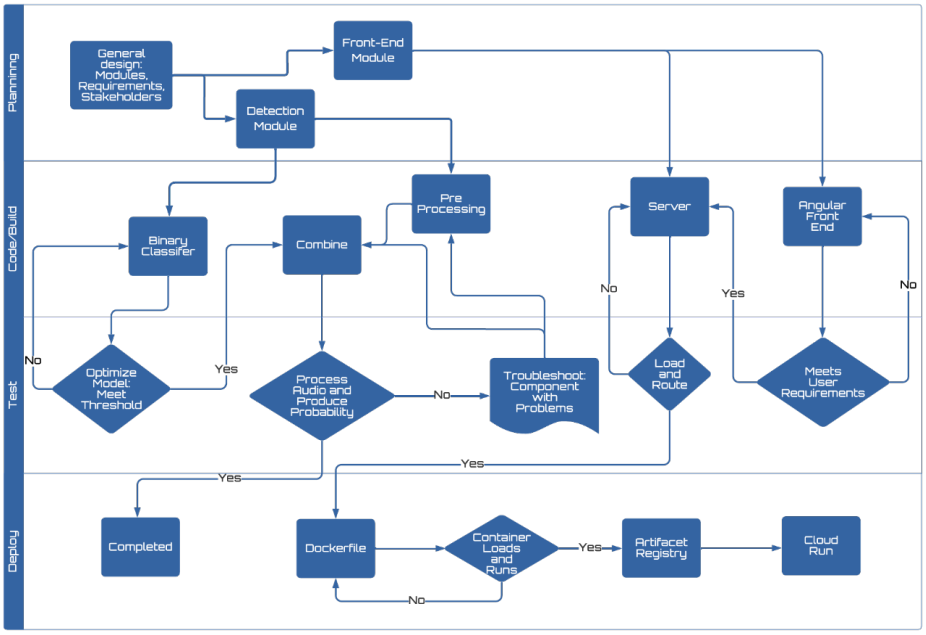


Figure 6: Deployment cycle in this project.

module. Additionally, it will accept the Web Socket request from the client after the Angular framework has compiled in their browser. This web socket connection with all clients will allow the front-end server to make visual changes as POST requests are sent by the detection module and removes the need for the client to consistently request an update.

Deployment was completed by creating a Docker file with all the necessary commands and dependencies to create the image of the front-end. Using Google Cloud's command line interface, the image was pushed into the artifact registry of Google Cloud. With the image pushed into a Google repository, the image can be deployed with Cloud Run, that produces a URL accessible with any browser.

a) Testing Plan and Analysis

The testing for the deployment of the front-end consists of addressing the client view of the deployment and assessing any acknowledgement of POST requests sent. The localized version of the deployment was successful and processed any probability changes visually. However, the deployed version demonstrated the constraints of the Cloud Run resource. Unfortunately, an unsecured web socket connection between client and server could not be completed through Google resources and any visual updates cannot be seen with a browser. However, the server can still process any available POST request sent only through the URL and route and can be seen with the Cloud Run logs. It was discovered that including the container port will prevent the container from receiving and processing any POST requests.

C. Development Operation Cycle

The development operation cycle of this project consists of four major sections: planning, coding, and building, testing, and deployment. The planning sections consisted of sections II and III where we identified the main modules we wished to focus on for the proof of concept. Section IV consists of the building and design for the two modules created while section V explains the testing and evaluation procedures used for the modules and the deployment process.

Each component was divided into two sections in the building stage, where each section was troubleshooted for errors until it was ready to merge for the modules, that then proceeded with further troubleshooting. After the intense troubleshooting process, the deployment considered the readiness of the front-end and the process of producing the Cloud Run URL.

VI. CONCLUSION

The completed proof of concept pipeline demonstrated how a streamlined audio input can be processed and received by the front-end server. The detection module was optimized for real-time information updates with only about 400 ms of delay. This analysis also explored key components of the Angular application, underscoring their roles in enhancing functionality and user experience, particularly in managing critical incidents. Components like buttons, card lists, navbars, and video modals provide the foundation for rapid responses and effective communication with first responders, illustrating modern web development practices and strategies for creating dynamic and interactive web applications.

There are additional components and modules needing planning and development to accomplish all the objectives for WatchDog. The detection module can be expanded to include a multi-classification module, trained with a personally funded data set for various gun types to produce accurate type predictions. Creating a more diverse and extensive data set provides a more reliable testing basis and gives more opportunity for optimization. There can also be additional front-end pages dedicated to first-responder protocols and limited views when they are sent out to the public.

The tracking module is needed to collect real-time information and maintain a record of the incident as well as

information such as the path traveled by the perpetrator so first-responders can detain them. The video management module can also be used to provide live video footage of the incident and provide information for tracking and logging.

Areas of improvement for the WatchDog system include better time-based accuracy with an ideal time frame for notifications and how consistently changes must be reported. It is also crucial for a greater understanding of what information is necessary for real-time reporting and general logging. Moreover, what visualization aspects make the UI/UX more user-friendly for all stakeholders. Finally, who would benefit from making the final decision when the system has identified a gunshot and what type of overriding privileges are critical. These concerns can only be addressed with the involvement of the stakeholders, considering their opinions and advice on making this system reliable and simple for emergency procedures.

ACKNOWLEDGMENT

The authors of this paper extend their gratitude to Dr. Nigel John due to his consistent advice throughout the development of WatchDog and assistance with any troubleshooting necessary.

REFERENCES

[1] "Mass Shooting Factsheet | Rockefeller Institute of Government," Rockefeller Institute of Government, Jun. 07, 2018. https://rockinst.org/gun-violence/mass-shooting-factsheet/ (accessed Apr. 16, 2024).

[2] "Public Mass Shootings: Database Amasses Details of a Half Century of U.S. Mass Shootings with Firearms, Generating Psychosocial Histories," National Institute of Justice, 2024. https://nij.ojp.gov/topics/articles/public-mass-shootings-database-amasses-details-half-century-us-mass-shootings (accessed Apr. 16, 2024).

[3] A. Morehead, L. Ogden, G. Magee, R. Hosler, B. White, and G. Mohler, "Low Cost Gunshot Detection using Deep Learning on the Raspberry Pi," IUScholarWorks (Indiana University), Dec. 2019, doi: https://doi.org/10.1109/bigdata47090.2019.9006456.

[4] Tuncer, T., Dogan, S., Akbal, E., Aydemir, E. (2021). An Automated Gunshot Audio Classification Method Based On Finger Pattern Feature Generator And Iterative Relieff Feature Selector, *Journal of Engineering Science of Adıyaman University*

[5] http://cadreforensics.com/audio/ (NO LONGER AVAILABLE)

[6] Kabealo, Ruksana et al. "A multi-firearm, multi-orientation audio dataset of gunshots." *Data in brief* vol. 48 109091. 25 Mar. 2023, doi:10.1016/j.dib.2023.109091

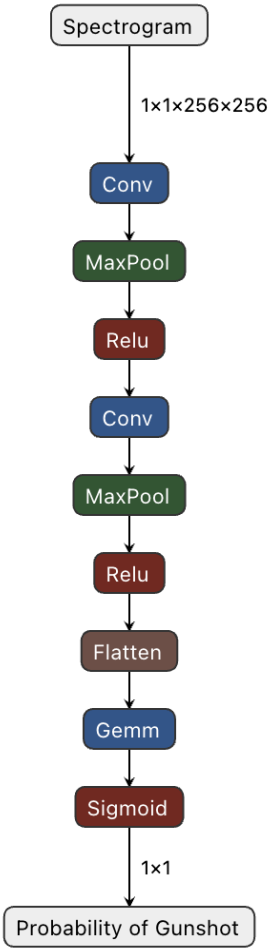
[7] J. Salamon, C. Jacoby and J. P. Bello, ″A Dataset and Taxonomy for Urban Sound Research*″, 22nd ACM International Conference on Multimedia,* Orlando USA, Nov. 2014.

[8] K. J. Piczak. "ESC: Dataset for Environmental Sound Classification." *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia, 2015.

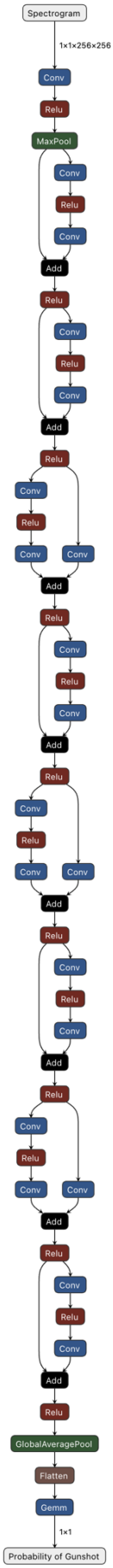
[9] Moez Krichen, "Convolutional Neural Networks: A Survey," Computers, vol. 12, no. 8, pp. 151–151, Jul. 2023, doi: https://doi.org/10.3390/computers12080151.

A. Network Diagrams

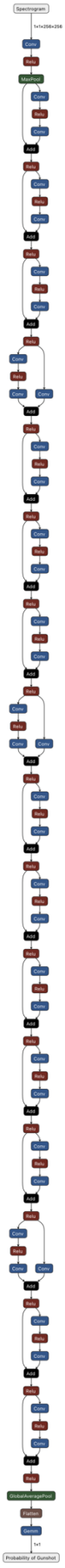
A1. Simple CNN



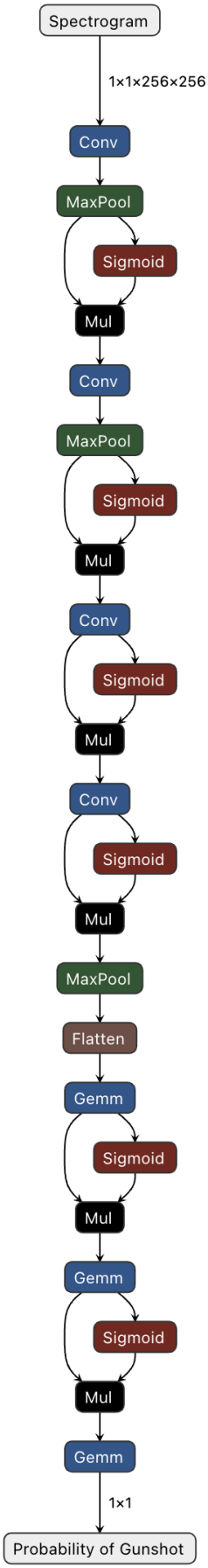
A2. ResNet18



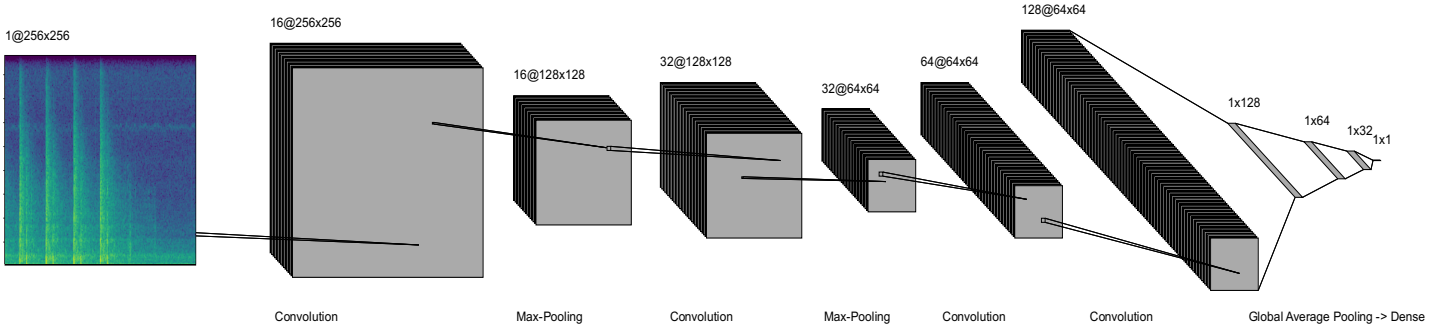
A3. ResNet34



A4. Enhanced CNN



A5. Enhanced CNN – 3D View



B. Example Spectrogram