UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO INGENIERÍA ELÉCTRICA

DISEÑO DE SISTEMA DE DETECCIÓN Y LOCALIZACIÓN DE DISPAROS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

JAVIER IGNACIO URRUTIA REBELLO

PROFESOR GUÍA:
PABLO ANDRÉS MEDINA COFRE

MIEMBROS DE LA COMISIÓN:
CESAR AUGUSTO AZURDIA MEZA
PABLO GEOVANNY PALACIOS JATIVA

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE: Ingeniero Civil Eléctrico
POR: Javier Ignacio Urrutia Rebello
FECHA: 2021
PROFESOR GUÍA: Pablo Andrés Medina Cofre

DISEÑO DE SISTEMA DE DETECCIÓN Y LOCALIZACIÓN DE DISPAROS

En este trabajo se plantea diseñar un sistema que permita detectar y localizar disparos de armas de fuego en la vía pública, de forma automática y con bajo costo de implementación. Para lograr este objetivo, se decide en primer lugar utilizar un entorno de simulación para sintetizar sonidos que permitan evaluar el sistema en términos de tiempo de respuesta y error de localización. En segundo lugar, se debe implementar un modelo de clasificación que permita reconocer sonidos de disparos. Como tercer punto, se propone crear el diseño utilizando herramientas de *Microsoft Azure* y así aprovechar tecnologías como *IoT* y computación en la nube.

Siguiendo estos objetivos, se comenzó el trabajo creando dos entornos de simulación, uno simple con 4 micrófonos y otro complejo que replica 6 cuadras con 8 micrófonos. Estos fueron utilizados en un programa de simulación que permitió posteriormente sintetizar el audio que captaría cada micrófono para disparos producidos en esos entornos.

Luego se trabajó en la implementación de un modelo de clasificación basado en redes convolucionales. Para ello se utilizó un conjunto de datos con más de 8.000 muestras de sonidos urbanos, incluyendo disparos, para entrenar la red. El resultado fue un modelo capaz de alcanzar una precisión de 100 % y *recall* de 94 % en el conjunto de test utilizado.

El diseño del sistema se basó en utilizar el servicio *IoT Hub* de *Azure* que permite un fácil manejo de múltiples dispositivos *IoT*, que para este proyecto corresponden a los dispositivos en terreno que poseen micrófono. El resto del sistema se diseño para utilizar un servidor virtual en *Azure*, con un esquema basado en módulos. Así, se implementó uno que interactúa con *IoT Hub*, otro que realiza clasificación de sonidos y finalmente uno que realiza la localización. Estos tres se comunican entre sí mediante una base de datos, lo que facilita también la inspección de las detecciones realizadas.

Finalmente se utilizaron los audios simulados en los dos entornos creados para evaluar el sistema, obteniendo tiempos de localización entre 20 y 30 segundos y errores de localización cercanos a 50 cm. Esto significa que este sistema tiene rendimiento similar a otros sistemas comerciales.

A mi querida familia, que siempre me ha apoyado

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Contexto

Los disparos al aire en las calles es un problema existente en nuestro país, llegando incluso a herir o matar a distintas personas. Estos hechos son conocidos coloquialmente como ”balas locas” y no es difícil encontrar noticias al respecto como en [1] donde ya se reportaban 6 victimas a febrero de 2020. Según la Encuesta Nacional Urbana de Seguridad Ciudadana de 2018 [2] cerca de un 10 % de los encuestados afirmaba que siempre suceden balaceras o disparos en sus comunas, lo que da cuenta de la constancia de estas situaciones.

El problema con este tipo de hechos es que normalmente no hay testigos y no se sabe a *priori* de dónde provino el disparo. Esto dificulta la respuesta policial por la falta de información y con ello también aumenta la desconfianza de la comunidad [3]. Como una respuesta a esto, se han creado distintos sistemas de localización de disparos mediante sonido en varias ciudades del mundo.

El origen de estos sistemas se remonta al ámbito militar en la primera guerra mundial, donde se empleó para ubicar artillería[4]. Posteriormente se empezaron a utilizar en ciudades para detectar disparos hechos en las calles y actualmente se han implementado en varias localidades[5] incluyendo Chile[6].

La utilidad de estos sistemas está en que pueden entregar información en tiempo real de cuando y dónde se producen las balaceras, sin necesidad de que exista una denuncia y con ello se aumenta la cantidad de casos detectados. De esta forma el sistema permite una acción más oportuna y eficaz de la policía [3], [7].

Además, al registrar los disparos con su localización generan información adicional que no se tenía previamente. Esto permite la detección de *hotspots* donde se producen con más frecuencia estos casos y pueden permitir la generación de mejores políticas para hacerles frente[8](citado en [3]). De esta forma se puede aumentar la sensación de seguridad en una comunidad y se puede disminuir la cantidad de casos en el tiempo.

En principio, estos sistemas son capaces de localizar donde se produjo un disparo al disponer de una red de micrófonos en las intersecciones de las calles o cornisas de edificios. Éstos escuchan la detonación del arma y permiten triangular su posición a partir de la dife-

rencia de tiempo en que cada sensor recibió el sonido. Para realizar la detección, un servidor debe recibir la información de los sensores para definir si corresponde a un disparo y obtener la ubicación.

Respecto al desempeño se han reportado tasas de verdadero positivos entre 84 % a 97 % en un tiempo de 5 a 90 segundos y con errores de localización entre 3.3 a 12.5 metros. Particularmente en Estados Unidos se han visto reducciones de casos de disparos de hasta un 50 % en las pruebas de estos sistemas[3].

Sin embargo, existen dificultades en su uso. Una de ellas es que al estar inmersos en un entorno urbano, las estructuras como casas y edificios bloquean la propagación libre del sonido, dificultando su detección y haciendo necesario la instalación de más sensores para mantener la efectividad. Por otra parte, se han reportado elevadas tasas de falsas detecciones (entre 3 % y 54 %). siendo la mayor causa sonidos como los fuegos artificiales.[3]

Otra de las quejas con estos sistemas son los elevados costos de mantenimiento,que puede llegar a los $230.000 dólares anuales en una ciudad [9](citado en [3]), [7], lo que produce la percepción de tener una baja relación costo/beneficio. En función de lo anterior se propone diseñar un sistema de detección y localización de disparos con un enfoque en lograr un bajo costo de implementación.

Este sistema será evaluado vía simulaciones que repliquen un entorno urbano y permitan localizar los sonidos generados. Otra parte importante es identificar que estos sean disparos por lo que se implementará un modelo de clasificación basado en redes neuronales convolucionales . Este será evaluado en términos de precisión y *recall*.

Para la implementación del software del sistema, se plantea utilizar herramientas de Microsoft Azure. Estas comprenden diversos servicios basados en la computación en la nube, como alojamiento de bases de datos, aplicaciones de *machine learning* e internet de las cosas (*IoT*) [10]. De esta forma se tendrá un mejor manejo de los módulos que estarían dispuestos en terreno y evita la necesidad de contar con un servidor físico, lo que reduce la complejidad del sistema.

1.2. Objetivos

1.2.1. Objetivo general

Diseñar sistema de detección y localización de disparos para entornos urbanos y con un bajo costo de implementación.

1.2.2. Objetivos específicos

- Elaborar un entorno de simulación que permita replicar la propagación del sonido en un entorno urbano para evaluar el sistema propuesto. Se medirá el tiempo de respuesta del sistema junto con el error en localización.
- Implementar un modelo de clasificación de sonidos que permitan identificar disparos.

Este se evaluará en términos de precisión y *recall*.

- Diseñar la solución utilizando herramientas de Microsoft Azure. Esta estará compuesta del *software* que utilizarían los módulos en terreno y los módulos que detectan y localizan los disparos.

Capítulo 2

Marco teórico y estado del arte

En este capitulo se presentan los conceptos necesarios para abordar este proyecto junto con la revisión del estado del arte. Se comienza con una visión general de ls sistemas de detección y localización de disparos, luego se presenta como se puede modelar y simular el sonido. Después se muestran métodos para clasificar y localizar sonidos. Finalmente se exponen las características específicas del sonido de un disparo.

2.1. Sistema de detección y localización de disparos

2.1.1. Historia

El origen de los sistemas de localización de disparos se remonta a la época de la primera guerra mundial, en una técnica denominada *sound ranging*. Esta técnica fue utilizada exitosamente por los Británicos para localizar y contraatacar la artillería enemiga, utilizando una red de micrófonos de baja frecuencia y operadores. Incluso podían determinar el tipo y calibre de la artillería, junto con el lugar de impacto.[4]

Avanzando en el tiempo, desde la década de los 90 se desarrollaron distintos sistemas para detectar disapros de francotirador. Entre éstos se puede destacar a *Hostile Artillery LOcator*(HALO) que utilizaba sensores acústicos para detectar disparos y morteros o el sistema *Boomerang* que va montado en vehiculos militares para detectar la posición relativa de los disparos.[11]

2.1.2. Actualidad

Los sistemas actualmente comercializados para la detección y localización de disparos se basan en el despliegue de una red de sensores acústicos en las ciudades. El objetivo de estos es el monitoreo continuo para identificar de manera automática o semi-automática el sonido producido por el disparo de un arma y localizar su procedencia. De esta forma se pueden generar alertas para que las autoridades tomen las medidas necesarias.

Por lo general, el sonido detectado por la red de sensores es transmitida a un servidor para ser procesado. Se emplean algoritmos de análisis de señales para discriminar el sonido de un disparo por sobre otros que se generan en un ambiente urbano, como puede ser el ruido de vehículos o personas. Además, con la información de varios sensores se calcula el

lugar donde se produjo el disparo y se pueden detectar otras características del sonido, como el tipo de arma utilizada. Una vez analizado el evento, estos sistemas envían la información georeferenciada al usuario. [11]

Respecto de la red de sensores, estos se suelen ubican en lugares altos como postes de luz o cornisas de edificios. Usualmente estos nodos cuentan con un solo micrófono o también pueden utilizar un arreglo de múltiples micrófonos. En sistemas grandes, la red puede cubrir decenas de kilometros cuadrados. [11]

Sobre el desempeño de estos sistemas, se han visto tiempos de respuesta entre 5 y 90 segundos de ocurrido el disparo. En cuanto a la localización, se ve un error entre 3.3 a 12.5 metros de la ubicación. Por el lado de la clasificación, tienen una tasa de verdadero positivo entre 87 % y 98 % pero sufren de una mayor tasa de falsos positivos de 5 % a 54 %.[3]

2.2. Modelo del sonido

El sonido es la perturbación de los átomos de un medio. Este se propaga como una onda, generalmente de forma oscilatoria, y en medios como el aire lo hace de forma longitudinal, es decir, desde la fuente hacia afuera.

Un efecto relevante es que la intensidad del sonido decrece con la distancia. Ésta se define como la potencia dividida en el área que cubre la onda, $I = P/A$ y si la onda se transmite como una esfera, el área aumenta de forma cuadrática. Luego, la intensidad disminuye de forma proporcional al cuadrado de la distancia.[12]

2.2.1. Efectos ambientales

La propagación del sonido en un espacio abierto está sujeta a diversos fenómenos ambientales. Entre estos se incluyen gradientes de temperatura, el efecto del viento o la absorción ambiental del sonido. Los gradientes de temperatura producen diferencias en la velocidad del sonido, y sumado al viento puede hacer que el nivel del sonido captado difiera de lo que se pueda predecir con modelos que solo consideren la propagación geométrica y la absorción atmosférica. También las precipitaciones cambian el nivel de humedad del aire, la que atenúa el sonido en forma creciente con la frecuencia. La relevancia de éstos efectos aumenta con la distancia al sonidos y deben ser consideradas si el sensor acústico está a más de 100 metros.[13]

La velocidad del sonido en un gas ideal está dada por la relación $v = \sqrt{\gamma \cdot R \cdot T/M}$ [12]. La constante γ corresponde al coeficiente de dilatación adiabática, R es la constante de los gases, T es la temperatura en grados Kelvin y M es la masa molecular del gas. Utilizando las constantes asociadas al aire seco, se obtiene la velocidad del sonido como:

$$v = 20.05\sqrt{T} \tag{2.1}$$

Además se produce un efecto de reverberación en la propagación de la onda de detonación en el entorno urbano. Esto se da por las múltiples reflexiones del sonido en el entorno, como

en el suelo y en edificios u otras superficies. También la presencia de estructuras limita la propagación del sonido y generan situaciones donde no siempre se tiene un camino directo entre el origen del sonido y el micrófono que lo capta. El efecto de estos fenómenos resulta en la distorsión de la onda original y ,llevado al caso de estudio, puede afectar la precisión del sistema para detectar y localizar el sonido de un arma.[11]

2.2.2. Sistema lineal

Un medio de propagación como el aire se puede aproximar como un sistema lineal invariante en el tiempo(*LTI*). Esto implica que el sonido se propaga de una forma repetible e independiente del tiempo, si es que las condiciones del medio no cambian. También, se puede aplicar el principio de superposición para obtener la respuesta de múltiples fuentes de sonido.[14]

Luego, la respuesta al impulso(*IR*) del medio permite modelar las perturbaciones que experimenta la señal al pasar por este. Así, la señal captada en un receptor ($g(t)$) dada la respuesta al impulso ($h(t)$) entre este y el emisor ($s(t)$) se puede calcular como la siguiente ecuación.

$$g(t) = h(t) * s(t) \tag{2.2}$$

Por ejemplo, si hay un receptor a 5 metros de una señal, por ejemplo un aplauso, se recibe el sonido en 0.015 segundos si se asume una velocidad de $343[m/s^2]$ del sonido. También, obviando pérdidas en el medio, la intensidad de la señal se reduce en $1/25$ y por lo tanto la amplitud en $1/5$. Esto se puede representar como la respuesta al impulso en la siguiente imagen(figura 2.1) que al aplicar la convolución con la señal original se obtiene la señal que captaría el receptor.

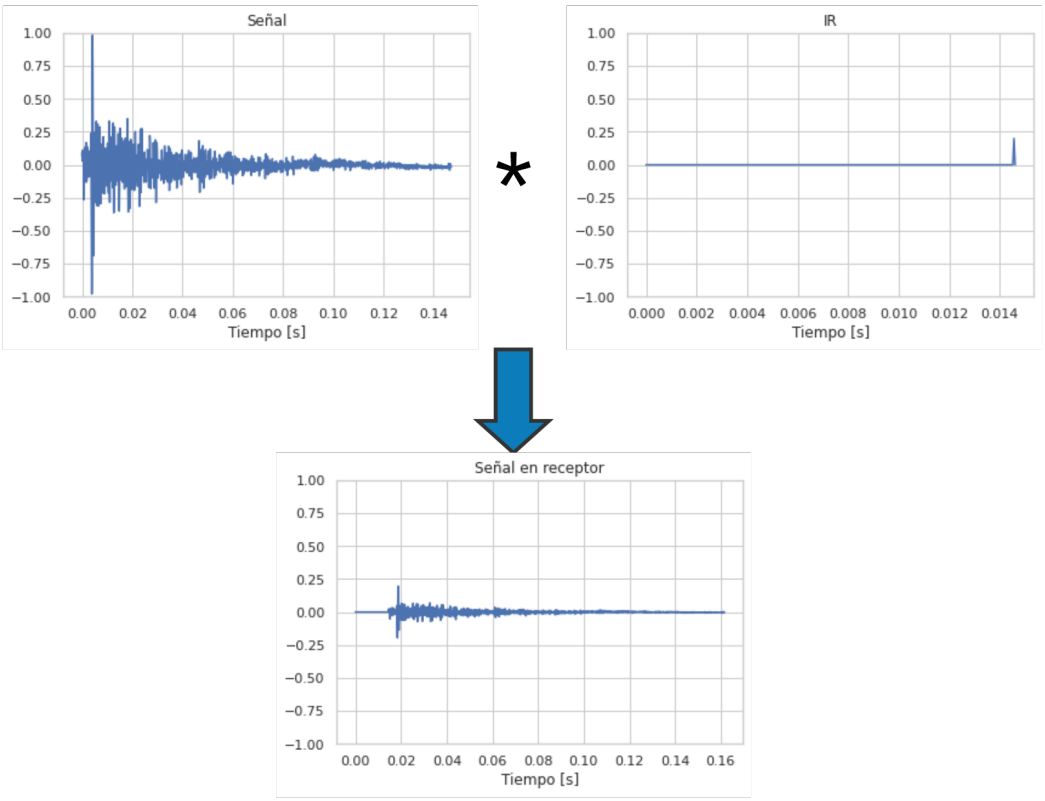


Figura 2.1: Ejemplo de convolución de señal de audio con respuesta al impulso.

2.3. Simulación de la propagación del sonido

Existen múltiples modelos para simular la propagación del sonido. Algunos se basan en las ecuaciones de ondas, como el método de diferencias finitas, y otros en la geometría del entorno, como *ray tracing*, entre otros. La elección del modelo depende del problema a resolver y las restricciones computacionales.[14]

En este trabajo se utilizará un método que realiza el rastreo de partículas de sonido(*Particle-Tracing*) y es similar a *ray tracing*. Este permite simular la propagación en un espacio abierto, incluyendo distintos fenómenos acústicos como la absorción, reflexión, efectos meteorológicos, etc.[15]

En el modelo, una fuente de sonido emite partículas que contienen una cantidad de energía por banda de frecuencia. Estas se distribuyen según la directividad del emisor, por ejemplo, si la fuente es omnidireccional, las partículas se distribuyen de forma uniforme en una esfera. Luego estas viajan por el medio hasta que colisionan con un objeto, donde pueden ser reflejadas, absorbidas, transmitidas, difuminadas y dispersadas según las características de ese objeto.[15]

También, se simula el efecto de la absorción atmosférica. En [15] se definen dos métodos para ello, uno que reduce la energía que llevan las partículas y el otro se modela la absorción como la probabilidad de que se absorban o eliminen algunas partículas.

Debido a la naturaleza estocástica del método, el receptor debe ser modelado como un volumen esférico finito como se representa en la figura 2.2. Luego, en cada instante de tiempo este contendrá una fracción de las partículas y con ello se puede determinar la energía total del sonido por banda de frecuencia que capta en ese momento. [15]

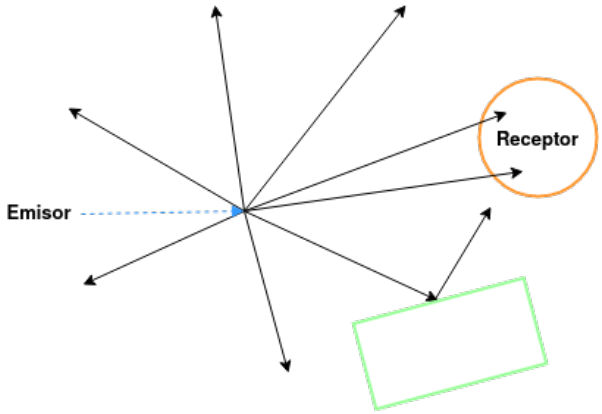


Figura 2.2: Representación de rastreo de partículas de sonido.

El resultado de lo anterior es un histograma del receptor que describe la energía de la señal recibida en cada instante de tiempo. Un ejemplo de este se puede ver en la siguiente figura, donde se recibe directamente una onda y posteriormente su reflexión. Se puede notar además que ambos eventos se ven extendidos en el tiempo producto del tamaño de la esfera que representa al receptor.

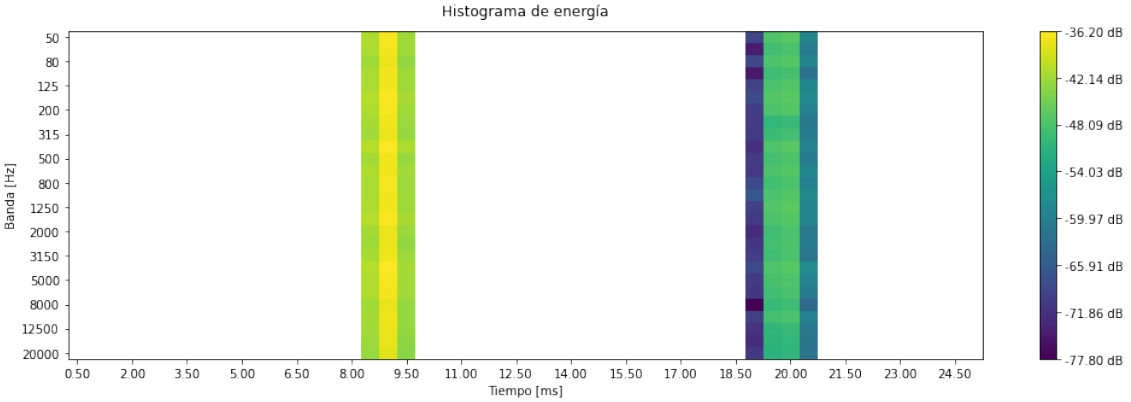


Figura 2.3: Ejemplo de histograma de energía por banda.

Finalmente, a partir de la información contenida en el histograma mostrado anteriormente se puede sintetizar la respuesta al impulso del medio entre el emisor y el receptor[16, pp. 70-72]. Esta representa la forma en que se modifica el sonido al viajar por el medio, y luego permite generar el audio que escucharía el receptor.

2.4. Localización de sonidos

Para localizar la posición del tirador, se utilizan métodos basados en la diferencia temporal de la llegada del sonido a los sensores, o TDOA(*time difference of arrival*). La razón viene por el hecho de que la única información disponible para estimar la posición es la ubicación de cada receptor o micrófono y que la señal llega en momentos distintos a cada uno. En la siguiente imagen se muestra la situación descrita.

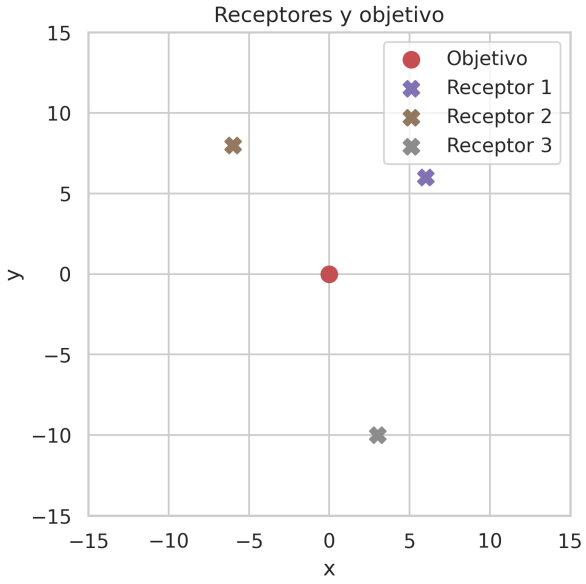


Figura 2.4: Receptores y objetivo a localizar.

La señal que se recibe en cada receptor se puede ver de la siguiente forma, donde existe un retardo dt entre la señal captada por uno y el otro. Esto está sujeto también a otras perturbaciones, como por ejemplo, que se reciban con distinta intensidad que es lo representado por el factor α [17].

$$r_i(t) = s(t - dt) \tag{2.3}$$
$$r_j(t) = \alpha s(t) \tag{2.4}$$

El término dt es lo que se denomina $TDOA$. Existen múltiples formas de estimar ese retardo descritas en la literatura y que se basadan principalmente en encontrar la correlación cruzada entre las muestras de audio que se están analizando[18]. Por otro lado, de conocer el momento t en que la señal de interés es captada por cada receptor, entonces se puede obtener simplemente como sigue, considerando un receptor a y otro b:

$$tdoa_{a,b} = t_a - t_b \tag{2.5}$$

Volviendo a la geometría del problema, y considerando la velocidad de la señal, esta diferencia temporal tiene relación con la distancia entre un receptor y la posición donde se origina la señal. Esto es pues si se conociera esta ubicación, el TDOA se podría calcular de la siguiente forma:

$$tdoa_{a,b} = \frac{\|\mathbf{r_a} - \mathbf{s}\| - \|\mathbf{r_b} - \mathbf{s}\|}{c} \tag{2.6}$$

En esta expresión se utiliza la posición de dos receptores a y b, se calcula la distancia de cada uno al origen de la señal y la resta resulta en el TDOA al dividir por la cvelocidad de la señal. Tomando como referencia el receptor 1 de la figura 2.4 y calculando el TDOA con los otros dos receptores se obtiene la siguiente situación.

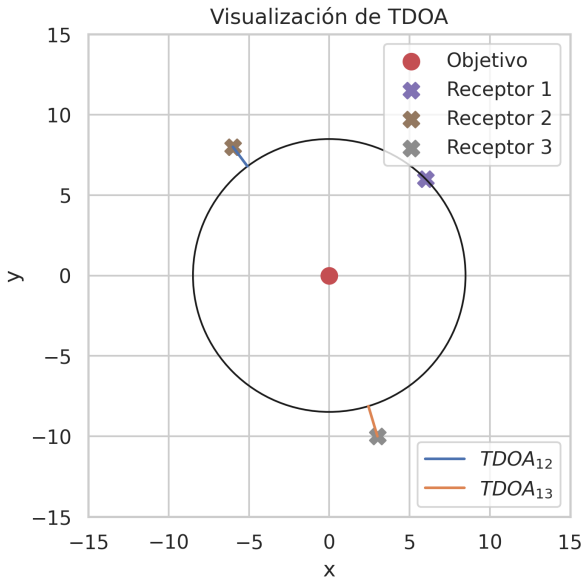


Figura 2.5: Visualización geométrica de TDOA.

Luego, se puede ver que si se poseen múltiples micrófonos, se puede construir un sistema de ecuaciones a partir de la ecuación 2.6 para encontrar la posición del objetivo. En forma geométrica, el conjunto solución de 2.6 para cada par de micrófonos resulta en la mitad de una hipérbola y la intersección de múltiples de estas resulta en la posición buscada. Esto se puede ver en el siguiente ejemplo.

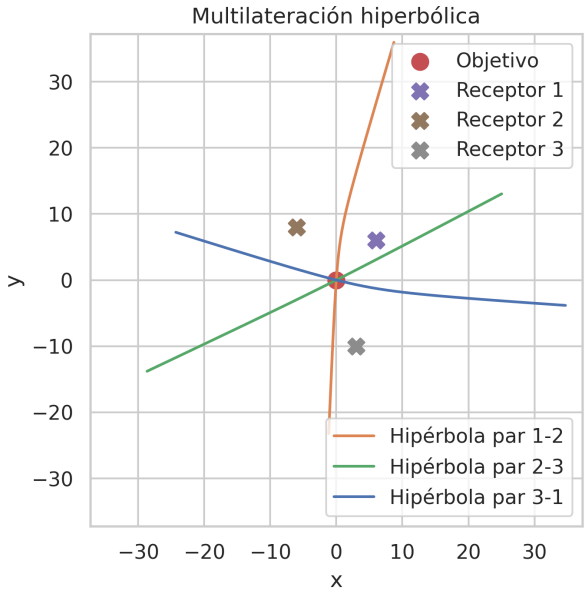


Figura 2.6: Multilateración hiperbólica.

Este método es lo que se conoce como multilateración hiperbólica [17][11]. En la practica, para resolver este problema y estimar la posición, se realiza una optimización basada generalmente en mínimos cuadrados o máxima verosimilitud[11].

2.5. Detección de ruido impulsivo

La detección de ruido impulsivo es una tarea que permite identificar candidatos a disparos en el sistema que se diseña. En [19] se describen una serie de métodos de detección basados en la utilización de umbrales para señales discretas. Estos parten de la base de construir una señal de detección, g_n, derivada de la señal original y donde es más fácil identificar *peaks* que corresponden a ruidos impulsivos.

La señal de detección que se utilizará en este trabajo corresponde al valor absoluto de la cuarta derivada de la señal de audio. La razón de su utilidad cae en que un ruido impulsivo corresponde a un cambio rápido en la señal. También, este enfoque permite analizar cada instante de tiempo de la señal discreta. Luego, esta se calcula de la siguiente forma:

$$g_n = \frac{|s_{n-2} - 4s_{n-1} + 6s_n - 4s_{n+1} + s_{n+2}|}{\Delta t} \tag{2.7}$$

Para el umbra utilizado en la detección, se puede emplear un esquema adaptativo. Este se basa en calcular el promedio de la señal de detección en una ventana de tiempo y este se amplifica por una ganancia. El calculo del umbra para cada instante de tiempo queda de la siguiente forma:

$$\nu_n = \frac{k}{2i+1} \sum_{m=n-i}^{n+i} g_m \tag{2.8}$$

Se puede notar de las ecuaciones 2.7 y 2.8 que es necesario agregar un retardo al detector si se desea emplear con una señal captada a tiempo real. Para el calculo de 2.7 se debe emplear un retardo de 2 muestras mientras que para el umbral adaptativo se requieren i muestras de retardo.

2.6. Identificación del sonido de un disparo

Para identificar el sonido de un disparo por sobre otros que pueden estar presentes en el entorno existen distintas técnicas. En principio, se puede determinar observando la amplitud de la señal captada por un micrófono. Si esta sobrepasa un umbral determinado por sobre el ruido ambiental, se puede clasificar como un posible disparo. Una vez que se detecta, se debe determinar si corresponde a un disparo u otra clase de sonido.[11]

La clasificación de sonidos se trata en tareas más generales como *Environmental Sound Classification*(ESC) o *Sound Event Recognition*(SER). En [20] se revisan distintas técnicas de clasificación sobre la base de datos *ESC-50*[21], donde se obtienen mejores resultados utilizando redes neuronales profundas, específicamente redes convolucionales o CNN(*Convolutional neural network*) con una exactitud de 79 %. En [22] se utiliza la base de datos UrbanSound8K[23] y se obtiene una precisión de 94 % sobre la clase disparo.

2.6.1. Redes neuronales convolucionales

Las redes neuronales convolucionales o CNN fueron diseñadas para procesar datos de entrada que presentan una distribución espacial, como en imágenes donde tienen mayor uso. Estas tienen tres componentes principales, las capas de comvolución, de *pooling* y otras capas asociadas. Una de las ventajas de este tipo de red es que requiere poco preprocesamiento de la entrada, pues las capas convolcionales actúan como filtros de características que se aprenden con los datos.[20]

En principio la operación de convolución se produce entre una entrada bidimensional con un filtro o *kernel* como se ve en la siguiente imagen. Se realiza la multiplicación punto a punto del *kernel* con una parte de la entrada del mismo tamaño, sumando los valores y obteniendo un valor de salida. Aplicando el mismo filtro y desplazándolo por todo el espacio de entrada se obtiene una salida también bidimensional.

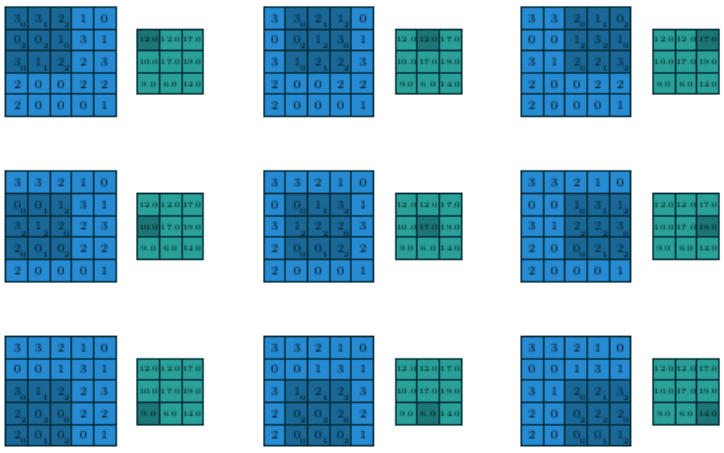


Figura 2.7: Capa de convolución.
Fuente:[24]

La forma de aplicar el *kernel* no está limitada a como se muestra en la figura anterior. Una primera modificación es permitir el *padding* con lo que se puede obtener una salida de las mismas dimensiones que la entrada (*half padding*) o mayor (*full padding*). También se puede definir el desplazamiento del *kernel* que es el *stride*. Un *stride* de 1 es desplazar el *kernel* de a un espacio cada vez, mientras que un *stride* mayor da saltos sobre la entrada, resultando en una salida de menor tamaño.

A la salida de la capa de convolución se aplica una función no lineal. Un ejemplo de estás son la función sigmoidea, tangente hiperbólica o ReLu. Esta última tiene la ventaja de ser lineal para números positivos, por lo que calcular su derivada durante el entrenamiento tiene un costo computacional menor.

Las capas convolucionales se suelen combinar con capas de *pooling* a la salida para otorgarle a la red cierta invarianza al desplazamiento de las características de la entrada. En esta capa se realiza un submuestreo de la entrada con funciones comunes de *pooling* son el *max pooling* o el *mean pooling*, que extraen el máximo o el promedio respectivamente de zonas no superpuestas de la matriz de entrada.[24]

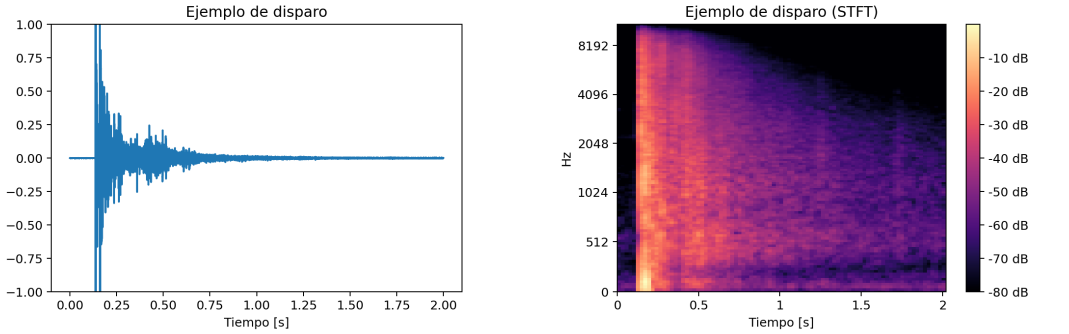
2.6.2. Preprocesamiento del sonido

Una muestra de audio es una secuencia discreta de magnitudes, pero se debe transformar para ser usada en una red convolucional. Una de las transformaciones utilizadas es el espectrograma con escala de frecuencias Mel y escala logarítmica en magnitud, usada por ejemplo en [22] y [25].

Para obtener el espectrograma de una muestra de audio, se calcula la transformada rápida de Fourier sobre ventanas o segmentos consecutivos de esta, que pueden estar solapados. Luego se aplican bancos de filtros triangulares que siguen la escala de Mel. Esta escala busca imitar el comportamiento del oído humano, que discriminan mejor en bajas frecuencias y menos en las altas.

En la figura 2.8 se puede ver una muestra de audio de disparo extraída de la base de datos

UrbanSound8K[23] transformada a espectrograma de Mel en escala logarítmica.



(a) Sonido de disparo

(b) Espectrograma Mel en escala logarítmica

Figura 2.8: Transformación de muestra de sonido a espectrograma en escala de Mel.

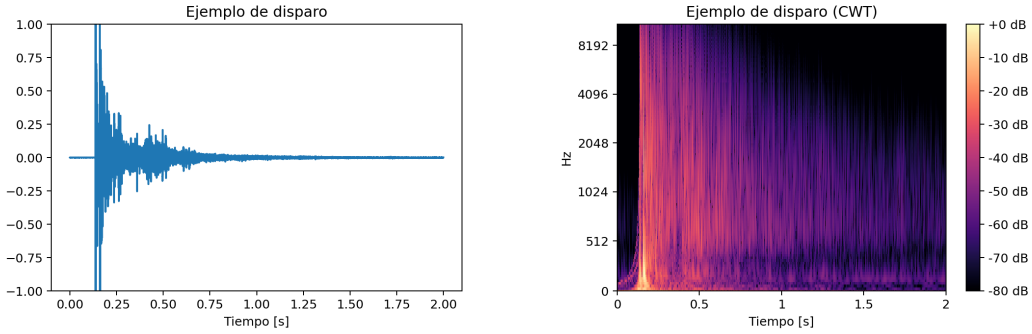
Una alternativa a utilizar la transformada de Fourier en una ventana de tamaño fija es la transformada continua de *wavelet*(*CWT*). Esta se caracteriza por permitir un análisis similar de la señal pero a una resolución variable según la banda de frecuencia [26]. La ecuación de esta tranformada es la siguiente:

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^* \left(\frac{t-\tau}{s} \right) dt \tag{2.9}$$

En esta, ψ se denomina como la función *wavelet* madre y es oscilatoria, de largo finito y existen distintas variantes. Esta función se multiplica con la señal x y actúa como una ventana de análisis. Luego, con el parámetro τ que representan la traslación se desplaza la ventana por la señal.

Por otro lado el parámetro s cambia la escala del análisis, donde se modifica el soporte de la función *wavelet* y su frecuencia central. Este se define como el periodo de la banda a analizar, o $\frac{1}{f}$. Luego, con este parámetro se pueden obtener coeficientes para distintas bandas de frecuencias.

En al siguiente imagen se muestra un ejemplo de un espectrograma obtenido a partir de la transformada de *wavelet* continua. En comparación a la figura (2.8) se ve una mayor resolución temporal en la zona de alta frecuencia. Las escalas utilizadas siguen la escala de frecuencias Mel y se utiliza el *wavelet* de tipo Morlet complejo.



(a) Sonido de disparo

(b) Espectrograma *CWT* en escala logarítmica

Figura 2.9: Transformación de muestra de sonido a espectrograma utilizando transformación de *wavelet* continua.

2.7. Caracterización de un disparo

Un arma convencional utiliza una carga explosiva para propulsar la bala que se dispara[13]. Dadas estas condiciones, el sonido de un arma pequeña se compone de las siguientes tres partes[3]:

- **Onda de detonación**: Onda producida por la detonación de la munición y la expulsión de gases a alta temperatura y presión del cañón.
- **Onda de choque balística**: Es producida por la trayectoria de la bala si su velocidad es supersónica.
- **Silbido**: Es producto de las turbulencias que deja el proyectil en su trayectoria.

Los sistemas actuales de detección de disparos solo detectan el primer componente, pues es la que se produce con mayor intensidad, alcanzando entre 140dB y 170dB en su cercanía. Además, la segunda componente no se produce en el tipo de arma usada usualmente en las ciudades, como son revólveres o escopetas, pues esos proyectiles no alcanzan velocidades supersónicas.

En la siguiente imagen (figura 2.10) se muestra como ejemplo la grabación de un disparo de revólver. En este se puede ver claramente la onda de detonación y que no presenta la onda de choque balística por el tipo de arma y munición. También se puede ver el reflejo de la onda de detonación.

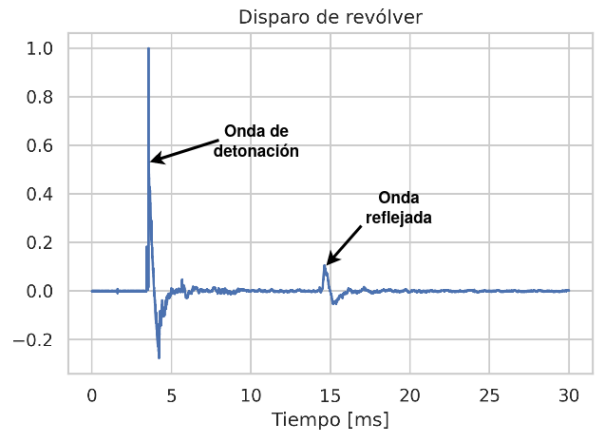


Figura 2.10: Ejemplo de disparo de revólver.

La onda de detonación también muestra comportamientos direccionales. Las componentes de baja frecuencia se propagan hacia el frente del arma, mientras que las de alta frecuencia lo hacen de forma omnidireccional[11].

Otro aspecto que resulta relevante para este trabajo es la duración del *peak* de la onda de detonación. En [27] se estudió la naturaleza del *peak* para distintos tipos de armas militares y en el caso de las pistolas se encontró una duración promedio de 0.2 ms. Esto implica que las componentes en frecuencia estarían principalmente bajo los 2500Hz.

Capítulo 3

Metodología

En este capítulo se presenta y explica la metodología empleada para cumplir cada objetivo específico. De esta forma, se comienza con el trabajo realizado para simular sonidos, siguiendo con el desarrollo del clasificador de sonidos y finalmente la arquitectura de software usada para el sistema completo.

3.1. Simulación

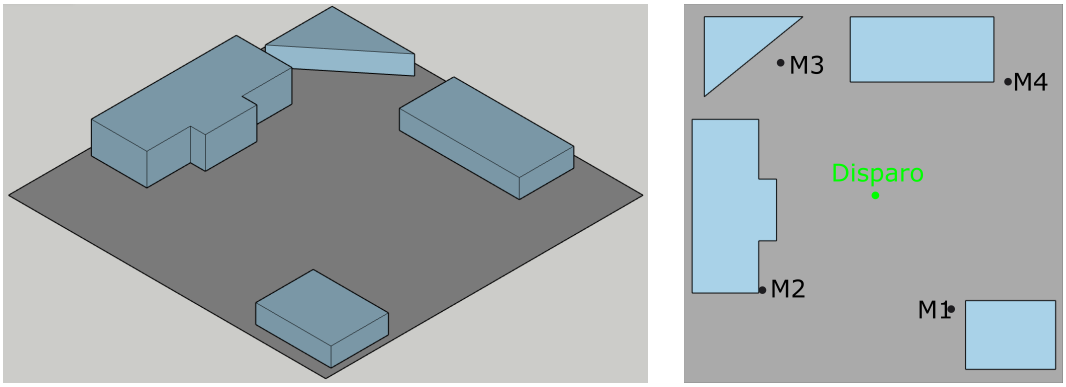
Se planteó en los objetivos específicos realizar simulaciones de disparos en ambientes urbanos para evaluar el rendimiento del sistema propuesto. Para ello se utiliza el modelo de *Particle Tracing* explicado en la sección 2.3 del marco teórico.

El programa de simulación empleado para esta tarea es *I-Simpa* [28] que permite modelar la propagación del sonido en entornos 3D. A continuación se muestran los entornos creados.

3.1.1. Entornos de simulación

3.1.1.1. Entorno simple

El entorno simple se diseño como un caso donde todos los micrófonos tengan linea de visión directa a la fuente de sonido y donde existan pocas estructuras donde se generen reflexiones. Como se ve en la figura 3.1, este cuenta con cuatro estructuras cerca de los bordes, dejando despejado el espacio del centro. El tamaño de este entorno es de 100x100 metros.



(a) Modelo 3D

(b) Ubicación de micrófonos y fuente de sonido

Figura 3.1: Entorno simple.

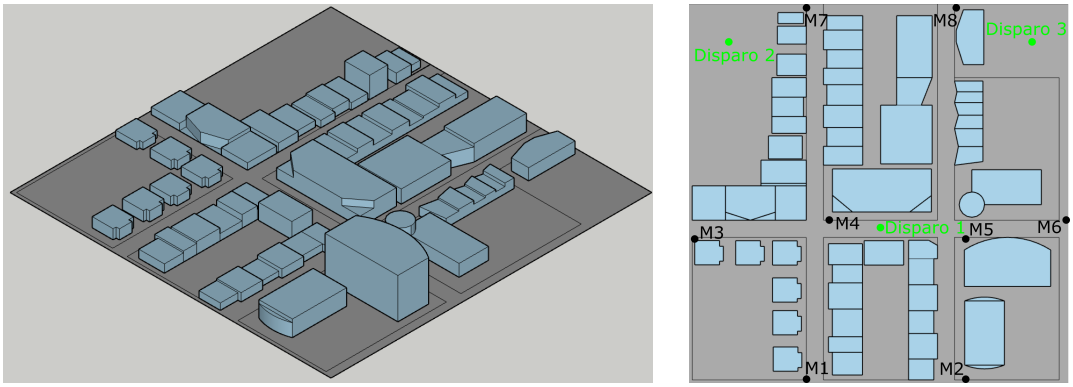
Los micrófonos fueron ubicados cerca de las estructuras y la fuente de sonido está en el centro. Específicamente, los micrófonos M1,M3 y M4 fueron ubicados a una altura de 4 metros simulando la instalación en un poste mientras que el micrófono M2 simula estar en la cornisa de un edificio. Los parámetros de la ubicación se pueden consultar en la siguiente tabla, donde el origen se encuentra en la esquina inferior izquierda de la imagen 3.1.b.

Tabla 3.1: Posición de elementos en entorno simple

Elemento	Coordenada X [m]	Coordenada Y [m]	Coordenada Z [m]
M1	70	20	4
M2	20.2	25	10
M3	25	85	4
M4	85	80	4
Disparo	50	50	1.5

3.1.1.2. Entorno complejo

En el entorno complejo se intenta modelar parte de una ciudad, con un terreno de 200x200 metros. Como se ve en la figura 3.2, se representan seis cuadras con estructuras de distintas formas y alturas. El fin de ésto fue fomentar la generación de ondas reflejadas.



(a) Modelo 3D

(b) Ubicación de micrófonos y fuentes de sonido

Figura 3.2: Entorno complejo.

Los micrófonos fueron ubicados en las esquinas y al final de las calles, todos a una altura de 4 metros simulando la instalación en un poste. También se simularon 3 ubicaciones para la fuente de sonido, una dentro de la calle del medio y dos más en las esquinas del entorno. En la siguiente tabla se pueden consultar los parámetros utilizados, , donde el origen se encuentra en la esquina inferior izquierda de la imagen 3.2.b.

Tabla 3.2: Posición de elementos en entorno complejo

Elemento	Coordenada X [m]	Coordenada Y [m]	Coordenada Z [m]
M1	61	2	5
M2	145	2	5
M3	2	76	5
M4	73	86	5
M5	145	76	5
M6	198	86	5
M7	61	198	5
M8	140	198	5
Disparo 1	100	82	1.5
Disparo 2	20	180	1.5
Disparo 3	180	180	1.5

3.1.2. Parámetros de simulación

Para llevar a cabo la simulación se deben especificar una serie de parámetros. Existen unos referentes al modelo 3D donde se realiza la simulación y otros son específicos del modelo *Particle Tracing*. A continuación se especifican los más relevantes.

Respecto del modelo 3D, el parámetro más influyente es el coeficiente de absorción de energía de las superficies. En primer lugar, para simular un entorno abierto, se crean superficie extra de cielo en el modelo con coeficiente 1.0 para que no reflejen sonido. Para el suelo y las estructuras se utilizó como referencia los parámetros vistos en un trabajo similar [29],

empleando los valores mostrados en la siguiente tabla e interpolando para otras bandas de frecuencia.

Tabla 3.3: Coeficientes de absorción según superficie y banda de frecuencia utilizados en simulación

Superficie	Coeficiente de absorción por banda [Hz]					
	125	250	500	1000	2000	4000
Pavimento	0.02	0.03	0.03	0.03	0.03	0.02
Fachada	0.10	0.05	0.06	0.07	0.09	0.08

Para el modelo *Particle tracing* se especifica en primer lugar las frecuencias que se simulan, que el programa utilizado están limitadas a una escala de tercios de octava en el rango de 50 HZ a 20.000 HZ. En el caso del entorno simple se utilizó el rango completo. Por otro lado, en la simulación del entorno complejo se acotó al rango 63Hz-16.000Hz para disminuir el tiempo de simulación.

Otro parámetro es el tiempo entre cada paso de la simulación, que fue fijado en 1 ms para ambos casos. También está el radio de la esfera que representa cada micrófono, fijado en 0.2 m para ambos casos. Éstos influye en la resolución temporal del histograma de energía que se obtiene de la simulación.

Finalmente se debe fijar el número de partículas. Éstas tienen que ser suficientes para que, una vez sean emitidas, logren cubrir el entorno y que llegue una cantidad adecuada a cada micrófono. Se tomó como referencia el modelo enunciado en [30] que define una relación entre el tamaño del entorno, el volumen que cubre la representación de un micrófono y la distancia entre este y la fuente de sonido.

Para aplicar la relación, se consideró la distancia más larga entre los micrófonos y las fuentes de sonido de cada entorno. De esta forma, para el entorno simple se estiman necesario 250 millones de partículas, mientras que el en complejo se requerirían cerca de 1.200 millones. Por limitaciones de velocidad de computo, se disminuyó este último a 1.000 millones.

En la siguiente tabla se resumen los parámetros utilizados en cada caso:

Tabla 3.4: Parámetros de simulación

Parámetro	Entorno simple	Entorno complejo
Rango de frecuencias [Hz]	50-20000	63-16000
Tiempo de paso [ms]	1	1
Radio esfera de receptor [m]	0.2	0.2
Número de partículas	$2.5 \cdot 10^8$	10^9

3.1.3. Obtención de respuesta al impulso

El resultado de la simulación por *Particle tracing* es un histograma de la energía recibida a lo largo del tiempo por cada micrófono. Este tiene un comportamiento similar a un espectrograma de Fourier, sin embargo, este contiene solo magnitud y no la fase de la señal. Por lo tanto, no es posible extraer directamente la respuesta al impulso.

Como alternativa, se puede sintetizar una respuesta al impulso que tenga un comportamiento similar en el tiempo. Un proceso como tal se describe en [16] para obtener la respuesta al impulso de una habitación.

En este proceso se modelan las reflexiones del sonido como eventos aleatorios, donde la cantidad de ocurrencias en un intervalo de tiempo sigue una distribución de Poisson. A partir de esto, se deriva la siguiente expresión para definir el delta de tiempo que ocurre entre reflexiones a partir de un numero aleatorio.

$$\Delta t_A(z,t) = \frac{1}{\mu(t)} ln\left(\frac{1}{z}\right) \quad 0 < z \leq 1 \tag{3.1}$$

En esta, z es un número aleatorio, mientras que μ es el promedio de la ocurrencia de reflexiones y esta dado por:

$$\mu(t) = \frac{4\pi c^3 t^2}{V} \tag{3.2}$$

Donde V es el volumen del entorno de simulación, c es la velocidad de la señal y t es el tiempo actual. Luego, ambas expresiones se utilizan para generar una señal aleatoria de eventos de reflexión.

Para ello, se representan las reflexiones como deltas de dirac de magnitud constante, empezando en $t_0 = 0.0014\sqrt[3]{V}$. Luego, se avanza en el tiempo en pasos $\Delta t_A(z,t)$, hasta llegar a la duración deseada. El signo de las deltas de dirac se intercambia en cada paso para adaptar la frecuencia de muestreo de la respuesta al impulso que se obtiene después.

Para sintetizar la respuesta al impulso, se crea un banco de filtros a partir de las bandas de frecuencia presentes en histograma de energía que entrega la simulación. Este se aplica sobre la señal aleatoria de deltas de dirac y como resultado se obtiene una señal filtrada por cada banda.

Como paso final, las muestras de las señales filtradas son ajustadas según la magnitud que muestra el histograma de energía en cada instante de tiempo. El resultado final se obtiene de sumar todas estas señales y se genera así una respuesta al impulso con un comportamiento similar al que tendría la verdadera.

3.1.4. Sintetización de sonidos de disparos

Una vez obtenida la respuesta al impulso para cada micrófono con el procedimiento explicado anteriormente, se deben sintetizar audios con disparos. Para ello se necesita una muestra de un disparo anecoico, o en otras palabras, que no presente reflexiones. En [31] se

realizaron grabaciones de múltiples armas de fuego en el desierto con condiciones prácticamente anecoicas.

Para las simulaciones a realizar en este trabajo, se tomó la siguiente muestra de disparo anecoico. Este corresponde al disparo de un revólver Ruger SP101.

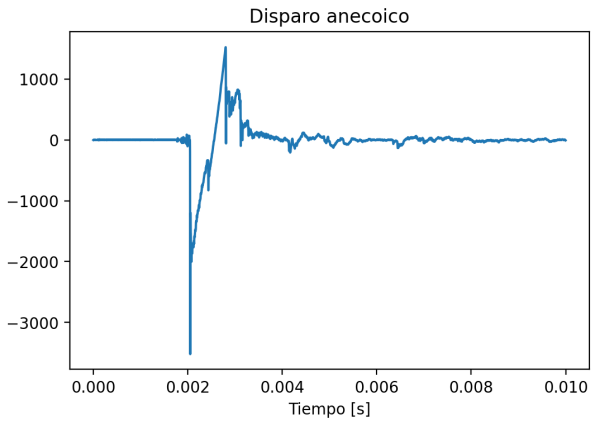


Figura 3.3: Disparo anecoico de revólver Ruger SP101.

3.2. Clasificación de sonidos

Para este proyecto se entrenó un modelo de clasificación de sonidos basado en redes neuronales convolucionales. Este tiene la labor de diferenciar entre ruidos y disparos dentro del sistema diseñado. A continuación se abordarán temas como los datos utilizados, su preprocesameinto y las métricas para evaluar el rendimiento.

3.2.1. Datos de entrenamiento

El conjunto de datos utilizado para entrenar el modelo de clasificación es *UrbanSound8k*[23]. Este posee 8732 muestras de sonido donde están representadas 10 clases de ruidos o sonidos urbanos. Las clases presentes junto con la cantidad de ejemplos de cada una se muestran en la siguiente tabla.

Tabla 3.5: Número de ejemplos por clase en *UrbanSound8k*

Clase	N° muestras
Aire acondicionado	1000
Bocina	429
Disparo	374
Ladrido	1000
Martillo neumático	1000
Motor	1000
Música callejera	1000
Niños jugando	1000
Perforación	1000
Sirena	929

La clase de interés para este trabajo son los disparos, que con sólo 374 muestras resulta ser la menos representada. Producto de esto se toman medidas explicadas en las siguientes secciones para reducir este problema durante el entrenamiento.

Finalmente, el conjunto de datos fue dividió en tres subconjuntos. El primero de *test* correspondiente a un décimo del total para evaluar el rendimiento final del clasificador. El segundo es de evaluación o desarrollo que también posee un décimo del total y se utiliza para detener el entrenamiento de la red convolucional antes de que se sobreajuste. El último es el utilizado para entrenar el modelo.

3.2.2. Aumentación de datos

Para mejorar el rendimiento del clasificador entrenado, se aplicó una serie de transformaciones a las muestras de audio para aumentar la cantidad de muestras disponibles. Siguiendo los resultados expuestos en [22] se aplicaron las transformaciones que daban mejores resultados para clasificar disparos. Estas son variaciones leves de *pitch* de un semitono, aplicar filtros comunes de compresión de rango dinámico y agregar ruidos de fondo.

La variación de *pitch* se realizó subiendo o bajando en un semitono cada muestra del conjunto de entrenamiento, manteniendo la duración. Se eligió ese rango pues cambios mayores pueden cambiar completamente como suena el audio y la intención es sólo aumentar la variación en los ejemplos.

La compresión de rango dinámico altera el volumen de los sonidos fuertes y suaves de una señal de audio para reducir el rango dinámico que posee. Este proceso se suele utilizar en la grabación, reproducción y/o transmisión del sonido. Tal como en [22] se utilizaron cuatro configuraciones pensadas para música, películas, habla y radio.

Para agregar ruido de fondo se utilizaron cuatro muestras de sonidos que contienen ruido de obras de construcción, ruido de tráfico, gente en la calle y ruido de un parque. Estos fueron combinados realizando una suma ponderada entre cada muestra de ruido y del conjunto de entrenamiento. El factor de ponderación para el ruido fue aleatorio en cada caso en el rango [0.2 ,0.4].

Adicionalmente, se agregó como transformación adicional un desplazamiento del inicio de cada muestra de audio. Esto tiene el fin de mejorar el rendimiento del clasificador cuando el disparo ocurre en distintos instantes dentro del audio que se clasifica.

Producto de lo anterior, se logró aumentar el conjunto de entrenamiento a 84.948 ejemplos, En teoría, esto permitirá que la red generalice mejor y también que el rendimiento no sufra por el hecho de que los disparos son la clase menos representada de todo el conjunto.

3.2.3. Pre-procesamiento de datos

Como se mencionó en la sección 2.6.2, es necesario procesar el sonido para que sea clasificado en la red convolucional.

En primer lugar, cada muestra de audio fue ajustada a una duración de tres segundos. Esto se realizó recortando o agregando silencio cuando el audio era más largo o corto respectivamente. Se eligió esta cantidad pues más del 90 % de los ejemplos de disparos tienen una duración de al menos esa cantidad.

Luego, es necesario transformar el audio a una representación bidimensional de tiempo y frecuencia. Para ello se utilizó la transformada continua de wavelet(**CWT**), con la wavelet de tipo Morlet. Puesto que le frecuencia de muestreo es 22.050 Hz, se aplica el análisis a distintas escalas para cubrir un rango de [11.025 Hz, 0.5Hz] en escala logarítmica, quedando 97 bandas de frecuencia.

Este espectrograma es de tipo complejo inicialmente, por lo que se toma el valor absoluto en cada componente para pasarlo a un espectrograma real. Adicionalmente se cambia a una escala logarítmica en decibel, tomando como referencia el valor máximo presente en este. Luego, cada espectrograma queda originalmente en el rango [0 dB, -80 dB] donde se elige -80 dB como punto de corte.

Adicionalmente, debido a que al utilizar **CWT** se obtiene un espectrograma con la misma resolución temporal que el audio original, es necesario realizar un proceso de submuestreo para reducir el tamaño. Esto se consigue filtrando el eje temporal con un filtro pasa baja y luego seleccionar muestras a una distancia regular. Con este proceso se consigue un espectrograma de tamaño 97x166.

Finalmente, se realiza un proceso de normalización para llevar los espectrogramas al rango de valores [0,1]. Esto es necesario pues al realizar la etapa de filtrado y submuestreo se altera el rango de valores en decibel. Además, reducir el rango permite un mejor entrenamiento de la red neuronal.

3.2.4. Métricas de evaluación

En este trabajo, la clasificación del sonido se abordará como un problema de clasificación binario, detectando si existe o no un disparo. En este formato se define una clase de interés como la clase positiva, y otra como la clase negativa. Al evaluar el clasificador sobre un conjunto de datos, se pueden tabular los resultados en una matriz de confusión que registra cuatro cantidades.[32]

En primer lugar se identifican los verdaderos positivos(*TP*) que son aquellos elementos correctamente clasificados como la clase positiva, de forma contraria, los falsos positivos(*FP*) son erróneamente detectados como positivos. En segundo lugar, los verdadero negativos(*TN*) son elementos de la clase negativa correctamente clasificados como tal, mientras que los falsos negativo(*FN*) son elementos de la clase positiva detectados como negativos.

A partir de esta matriz, se pueden construir distintas métricas de evaluación. Una de las que se utilizará en este trabajo es el *recall*, también conocido como tasa de verdaderos positivos. Esta mide la proporción de elementos de la clase positiva que fueron detectados y clasificados como tal por el modelo. Se calcula con la siguiente fórmula:

$$recall = \frac{TP}{TP + FN} \tag{3.3}$$

Otra métrica es la precisión, que ve la tasa de elementos clasificados como positivos que efectivamente lo son. Esta se calcula como se muestra en la siguiente ecuación:

$$prec = \frac{TP}{TP + FP} \tag{3.4}$$

El modelo de clasififcación implementado entregará un puntaje de clasificación que indica en cierta medida la confianza de que un ejemplo pertenezca a la clase positiva. Por lo tanto, se debe fijar un umbral para etiquetar la clase. Esto implica que la matriz de confusión y en particular el *recall* y la precisión dependen del umbral seleccionado.

Luego, resulta útil emplear también una métrica independiente del umbral. La que se seleccionó es el área bajo la curva precisión-*recall*(PR-AUC), curva que se genera al variar el valor del umbral entre 0 y 1, y calcular precisión y *recall* en cada caso. Esta métrica resultará necesaria más adelante para comparar el rendimiento del clasificador frente a distintos hiperparámetros.

3.3. Diseño del sistema

El diseño del software del sistema completo se realizó bajo el concepto de separar cada función y aprovechar los servicios de computación en la nube disponibles. Específicamente, en este trabajo se utilizará Microsoft Azure para construir el sistema. Producto de lo anterior es la arquitectura mostrada en la figura 3.4.

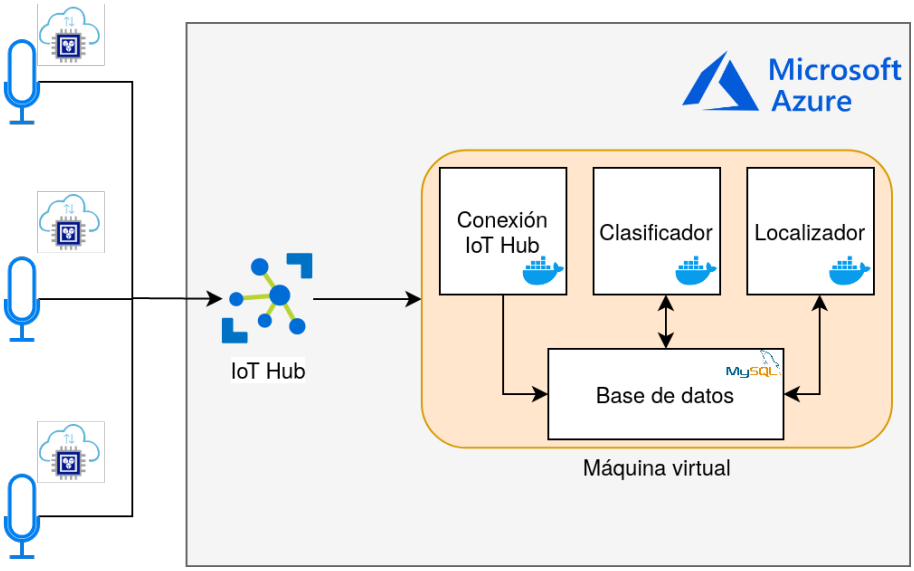


Figura 3.4: Arquitectura de software del sistema diseñado.

Como se ve en la imagen, la arquitectura esta diseñada en múltiples módulos. Esto permite abordar cada parte de forma semi-independiente para facilitar la programación. Además se pueden reconocer dos partes, una física donde están los micrófonos desplegados, y otra completamente en la nube utilizando una máquina virtual.

El servicio que une la parte física con la maquina virtual es *IoT Hub*. Este se encarga de administrar la conexión de dispositivos IoT a Azure, además de enrutar el traspaso de información entre ambos lados. Así, este servicio resulta una elección natural para el escenario que se quiere resolver, pues es fundamental permitir la conexión de múltiples micrófonos para que el sistema sea capaz de cubrir una zona amplia y localizar los disparos.

Por otro lado, el utilizar una máquina virtual en vez de un servidor físico posee algunas ventajas únicas. Por un lado se paga sólo por el uso de la máquina y se puede elegir la capacidad según los requerimientos específicos del sistema. Por otro lado, se evitan costos y complicaciones asociadas a mantener funcionando un servidor local.

Ahora se describirán los módulos programados en las siguientes secciones.

3.3.1. Módulo en terreno

El módulo en terreno corresponde al dispositivo que está desplegado y que capta el audio. Este fue diseñado teniendo en mente una unidad de computo equivalente a una Raspberry Pi[33] más un micrófono en forma esencial. Adicionalmente, este debe contener una conexión de datos en red celular.

La función que realiza es fundamentalmente detectar posibles disparos. Para ello, debe captar el audio y realizar una detección de ruido impulsivo. Una vez realizada esta, la debe comunicar al servicio de *IoT Hub* para un procesamiento posterior.

Este módulo se programó como un dispositivo de tipo *IoT Edge*[34]. Esto se utiliza cuando el dispositivo es capaz de realizar computo de forma local, antes de enviar información captada

por los sensores. Al darse este caso, su funcionalidad se debe programar en forma de módulos y se administrar desde *IoT Hub*.

Por esta razón, el software del módulo en terreno es el mostrado en la figura 3.5. Este está compuesto de dos submódulos descritos a continuación

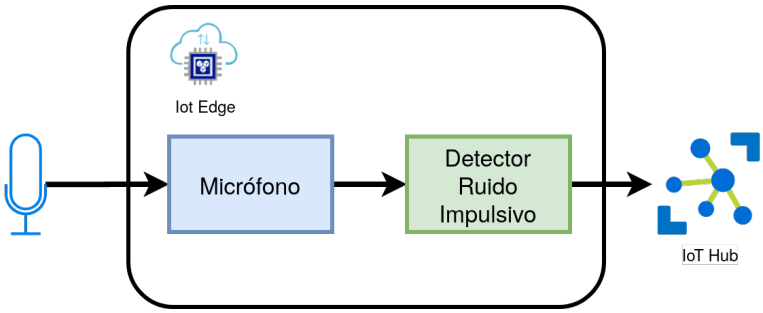


Figura 3.5: Módulo en terreno.

3.3.1.1. Súbmodulo de micrófono

El súbmodulo de micrófono es la interfaz con el micrófono conectado al dispositivo. Este capta audio en formato de 16 bits y a una tasa de muestreo de 22050 Hz. Luego, este pasa al siguiente submódulo en forma de bloques cortos de audio, a una tasa de 15 bloques por segundo junto con una estampilla temporal.

3.3.1.2. Súbmodulo detector de ruido impulsivo

El súbmodulo detector de ruido impulsivo tiene la labor de detectar los posibles disparos. Esto se logra implementando un detector a tiempo real basado en el valor absoluto de la cuarta derivada de la señal de audio, junto con un umbral adaptativo, como se explicó en 2.5. Sumado a lo anterior, también debe transmitir la información a *IoT Hub*.

La información que debe entregar para posteriormente lograr la clasificación y localización de un disparo son principalmente dos componentes. La primera es una estampilla temporal del momento en que se realizó la detección. La segunda es una muestra de audio de tres segundos de duración que será clasificada por el módulo de clasificación para confirmar la detección.

Para facilitar la implementación, este submódulo fue diseñado en base a una máquina de estados. Esto surge pues según este esperando una detección o si ya lo hizo debe cambiar su comportamiento. En el siguiente esquema se muestran los estados empleados.

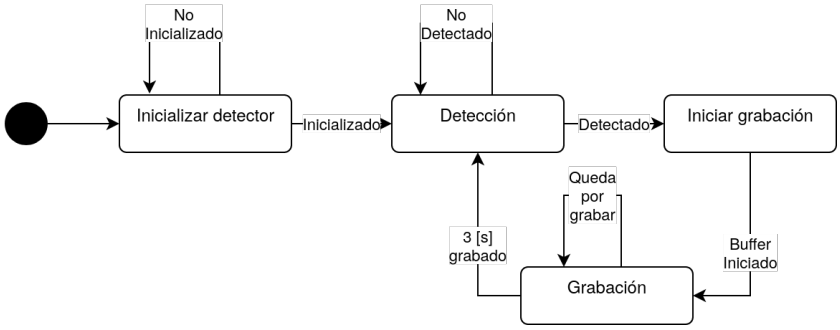


Figura 3.6: Máquina de estados del detector de ruido impulsivo.

Puesto que el umbral adaptativo impone un retraso en el detector, este implementa un *buffer* de audio que se debe inicializar para un correcto funcionamiento. Luego, el estado inicial del submódulo es esperar un tiempo para este proceso. Terminada esta etapa, se entra al estado normal de detección.

De ocurrir una detección en este segundo estado, se debe comenzar el proceso de grabar una muestra de audio para enviar al sistema. Aquí se pasa por un estado transitorio que permite iniciar un *buffer* de grabación donde se incorpora audio pasado equivalente al retardo del detector. Luego, se pasa al estado de grabación donde guarda cada nuevo bloque de audio entrante hasta completar tres segundos.

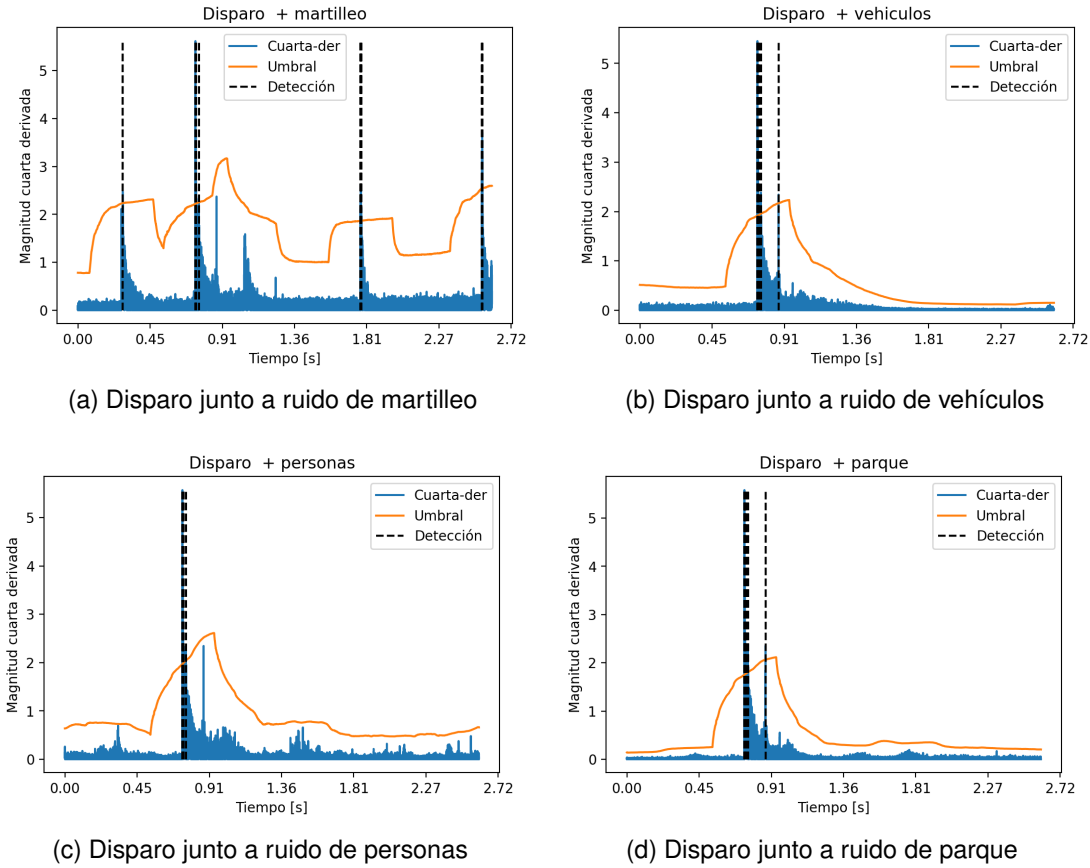
Terminada la grabación, un hilo separado realiza la tarea de enviar la información a IoT Hub, mientras que el submódulo vuelve al estado de detección para continuar su labor. De esta forma se asegura que siempre se puedan realizar detecciones, incluso si está subiendo información a la nube.

El detector de ruido impulsivo posee dos parámetros fundamentales. El primero es el ancho de la ventana utilizada para calcular el umbra adaptativo. El segundo es la ganancia del umbral. Los parámetros utilizados se muestran en la siguiente tabla.

Tabla 3.6: Parámetros del detector de ruido impulsivo

Parámetro	valor
Ancho ventana del umbral	$\sim$0.2 ms
Ganancia del umbral	15

A continuación se presentan cuatro ejemplos de como se comporta el detector de ruido impulsivo, utilizando un sonido de disparo mezclado con distintos ruidos a una SNR de 5dB. En cada uno se muestra el valor absoluto de la cuarta derivada del sonido, junto con el valor del umbral adaptativo, en cada instante y se marcan las detecciones. Se puede comprobar que logra detectar el disparo en cada caso, mientras que en el ejemplo 3.7.a también detecta cada martilleo.



(a) Disparo junto a ruido de martilleo

(b) Disparo junto a ruido de vehículos

(c) Disparo junto a ruido de personas

(d) Disparo junto a ruido de parque

Figura 3.7: Ejemplo de detección de ruido impulsivo.

Finalmente, cabe recalcar que este detector sólo tiene la intención de evitar el envío constante de información al servidor central del sistema. Por lo tanto, no se realiza un estudio exhaustivo de sus parámetros y es esperable que deje pasar múltiples ruidos que no corresponden a disparos y que serán distinguidos posteriormente por el clasificador.

3.3.1.3. Simulación de micrófono

Como se planteó en los objetivos específicos, el sistema diseñado será evaluado mediante simulaciones. Por esta razón es necesario modificar este módulo para permitir esta tarea. Gracias a la forma en que fue diseñado el sistema, es directo ver que basta con reemplazar el submódulo de micrófono por uno que lo simule,y sin necesidad de modificar el resto del sistema, como se aprecia en la figura 3.8.

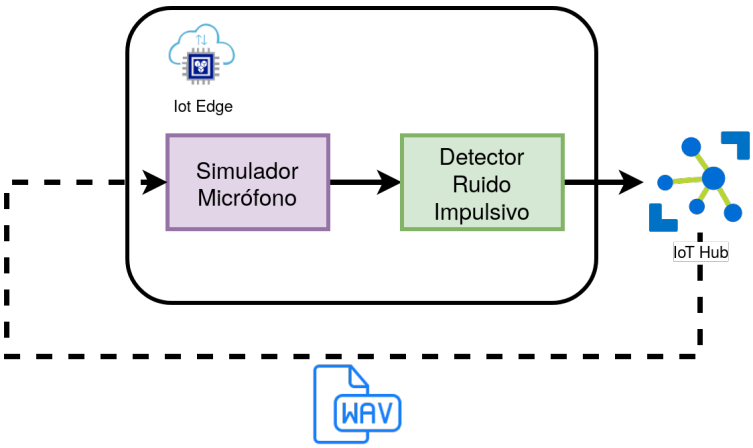


Figura 3.8: Módulo en terreno simulado.

El submódulo creado para la simulación simplemente recibe un archivo de audio desde *IoT Hub* y lo reproduce tal como si fuera audio captado desde el micrófono. Como requisito, el audio recibido debe tener las mismas especificaciones utilizado para captar el audio, que es una tasa de muestreo de 22.050 Hz y formato de 16 bit.

Una ventaja importante de poseer este submódulo es que todo el resto del sistema, y en especial la parte que se ejecuta en la nube, es prácticamente igual al que se utilizaría si se contara con dispositivos reales. Esto permite asegurar que todas las conexiones y módulos se puedan comunicar de forma correcta. También permite, salvo cierto margen de error, se podrá medir el tiempo de respuesta efectivo del sistema.

3.3.2. Base de datos

En la base de datos se maneja la información necesaria para cada etapa del sistema. Además, esta permite el traspaso de información de manera simplificada entre los módulos. La base de datos está dividida en tres tablas que se detallan a continuación.

En la primera tabla se almacena información de cada dispositivo desplegado. Esta es esencialmente el nombre con el cual se identifica en *IoT Hub* y las coordenadas de su ubicación. Esta es relevante para el proceso de localización de sonidos posteriormente.

En la segunda tabla se almacena los datos de cada detección realizada por los dispositivos en terreno. Esta se compone inicialmente del nombre del dispositivo que realizó la detección, una estampilla temporal y la muestra de audio de 3 segundos de duración. A esto se suma un par de variables para indicar el resultado de la clasificación y posteriormente si fue localizado.

En la tercera tabla se almacenan los disparos que lograron ser localizados. Como resultado se guarda la estampilla temporal de cuando ocurrió y las coordenadas. A futuro, se podría acceder a esta información mediante una interfaz de usuario.

3.3.3. Módulo conector IoT Hub

El módulo conector IoT Hub, tal como indica su nombre, se encarga de manejar la conexión entre *IoT Hub* y la base de datos presente en la máquina virtual. Este recibe cada detección nueva que envían los módulos en terreno y los ingresa a esta con el formato correcto.

3.3.4. Módulo clasificador

El módulo clasificador implementa el modelo convolucional entrenado para distinguir los sonidos de disparos. Este monitorea constantemente la presencia de nuevas entradas en la base de datos y las clasifica.

Cada entrada de detección tiene un campo para marcar la etiqueta clase y que toma un valor por defecto nulo. Así, este módulo verifica cada 10 segundos si existen entradas sin clase y, de existir, clasifica todas las que encuentra. La etiqueta marcada es un caractér que toma el valor 1 si es disparo y 0 si no.

3.3.5. Módulo localizador

El módulo localizador realiza la tarea final del sistema que es encontrar el punto donde probablemente se originó un disparo. Para ello monitorea la base de datos por entradas clasificadas como disparos pero que todavía no han sido localizadas. Luego ejecuta el algoritmo de multilateración hiperbólica para encontrar el origen del disparo.

Originalmente cada entrada de detección en la base de datos es independiente. Por lo tanto se deben implementar heurísticas para determinar que existe un conjunto relacionado. Para esta versión del sistema se implementó una basada en el tiempo.

En primer lugar, se buscan entradas que hayan ocurrido en los últimos 30 minutos, para asegurar que sean recientes, y que estén clasificadas como disparo. Luego, para encontrar aquellas que estén relacionadas, se toma va tomando una como referencia y se buscan si existen al menos otras dos que hayan ocurrido en una vecindad de 200 ms. De encontrar un grupo que cumple esta regla, se procede con la multilateración hiperbólica.

Pensando en una implementación más robusta, la búsqueda por el criterio del tiempo no es suficiente y se deberían agregar otras. Por ejemplo, como se tiene la información de donde está el micrófono que capta cada entrada, se puede agregar un criterio por distancia para no relacionar aquellas que ocurren de forma lejana. También se podría hacer una comparación con un criterio en base a la correlación de las muestras de audio para verificar que sean similares.

3.3.5.1. Multilateración hiperbólica

Una vez que se tiene un grupo de entradas relacionadas, se puede estimar la ubicación del origen del disparo resolviendo la multilateración hiperbólica. A continuación se detallan los pasos a seguir.

En primer lugar, se deben estimar los **TDOA** entre los pares de micrófonos asociados a

cada entrada de detección. Estos se pueden calcular a partir de las estampillas temporales presentes en cada una, que fue determinada al momento de realizar la detección de ruido impulsivo. La relación un micrófono a y b está dada por la siguiente ecuación, donde t es la estampilla temporal en segundos.

$$\hat{tdoa}_{a,b} = t_a - t_b \tag{3.5}$$

Por otro lado sabemos que si se conociera la ubicación de la fuente de sonido, el **TDOA** se podría calcular de la siguiente manera dada la geometría del problema. Aquí, los términos $\mathbf{r}$ hacen referencia al punto bidimensional donde se ubica un micrófono y $\mathbf{s}$ es la ubicación del origen del sonido.

$$tdoa_{a,b} = \frac{\|\mathbf{r_a} - \mathbf{s}\| - \|\mathbf{r_b} - \mathbf{s}\|}{c} \tag{3.6}$$

Luego, definiendo residuos como la resta entre ambas expresiones, se puede plantear el problema como la minimización del los errores cuadráticos. En su forma clásica se puede expresar de la siguiente manera, donde p es un indice asociado a cada par de micrófonos presente en el análisis.

$$\min_{\hat{\mathbf{s}}} \sum_{p} \left(\hat{tdoa}_{p_a,p_b} - \frac{\|\mathbf{r}_{p_a} - \hat{\mathbf{s}}\| - \|\mathbf{r}_{p_b} - \hat{\mathbf{s}}\|}{c} \right)^2 \tag{3.7}$$

El módulo implementado resuelve este problema mediante un *solver* numérico. Una vez que se obtiene una estimación de la ubicación, se guarda el resultado en la base de datos junto con el tiempo

Capítulo 4

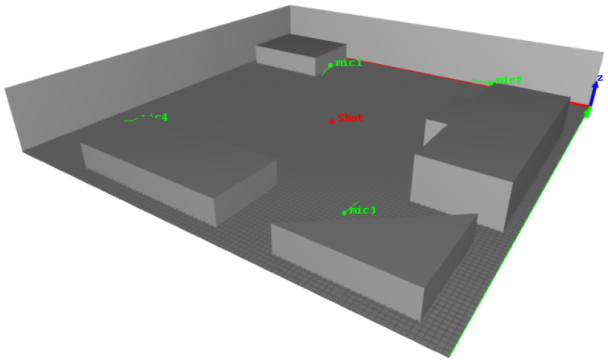
Resultados y discusión

En este capitulo se exponen los resultados obtenidos de cada objetivo específico. En primer lugar se muestran los disparos simulados y luego el modelo de clasificación implementado, junto con las métricas obtenidas. Finalmente, se muestran experimentos realizados sobre los módulos del sistema y se analiza el rendimiento del sistema completo.

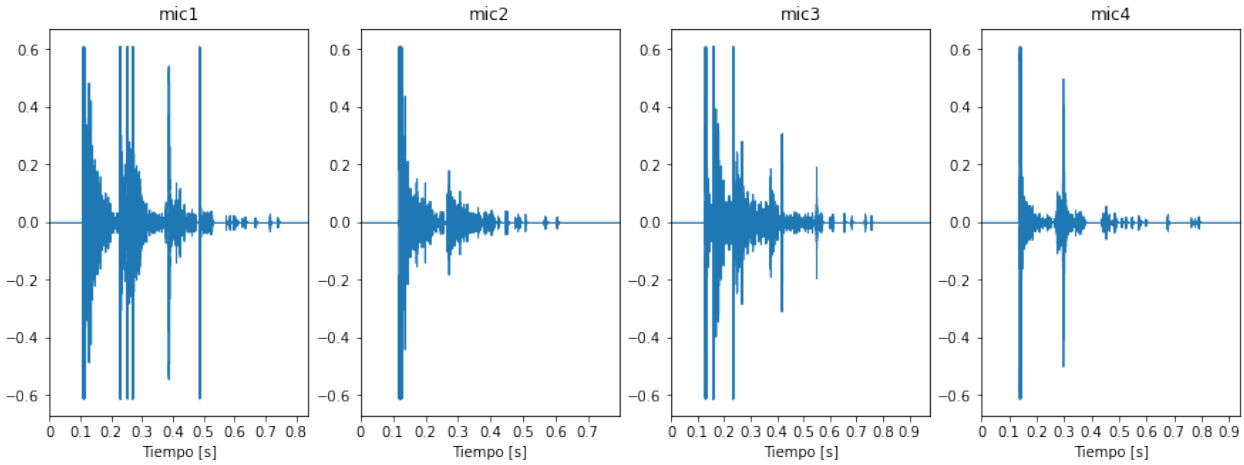
4.1. Simulación de disparos

La simulación de disparos se realizó como se indicó en 3.1.3 y resultó exitosa para los dos entornos propuestos. En total se sintetizaron 4 audios para el entorno simple y 24 para el entorno complejo. Esto marca el cumplimiento del primer objetivo específico de este trabajo.

Como ejemplo, en la figura 4.1 se puede ver el audio sintetizado para cada micrófono simulado del entorno simple. En especial se puede notar que el tiempo donde inicia el sonido es ligeramente distinto para cada uno y que se ve un número de reflexiones diferente dada la posición. En el caso del entorno complejo se ve la misma situación, solo que algunos micrófonos no captan la señal según donde se origine el disparo.



(a) Modelo 3D en simulador



(b) Audio sintetizado de cada micrófono

Figura 4.1: Audios sintetizados modelo simple.

Un punto a notar es que si bien los disparos sintetizados cumplen con las características deseadas, principalmente un retardo proporcional a las distancias que permiten la localización, estos no son de alta calidad. Esto se debe principalmente al paso de tiempo utilizado durante la simulación, que es de un milisegundo. Esto se podría mejorar utilizando un paso menor, a costa de mayor tiempo de simulación, pero como se verá mas adelante, no implica un inconveniente demasiado alto.

4.2. Modelo de clasificación

En primera instancia, se entrenó un modelo convolucional con una arquitectura basada en [22], que se tomó como modelo *baseline*. Este cuenta con 3 capas convolucionales, mas dos lineales, todas con función de activación ReLu, para entregar el puntaje de clasificación. El entrenamiento se realizó por múltiples épocas hasta que el set de desarrollo dejó de mostrar una mejoría en el puntaje PR-AUC y se fijó el umbral según los resultados en ese set. El rendimiento alcanzado en el set de *test* se resume en la siguiente tabla.

Tabla 4.1: Métricas modelo baseline

Métrica	Puntaje
Precisión	0.968
Recall	0.938
PR-AUC	0.941

Como se ve del puntaje, el rendimiento del modelo es bueno y comparable al de [22]. Sin embargo, dado que utiliza capas lineales en la salida, existe el riesgo de que se sobreajuste a que el disparo ocurra en determinado momento dentro de la ventana de 3 segundos que se analiza. Por lo tanto, se realizó un experimento para investigar si ocurre esto, donde se utilizan los disparos del set de desarrollo y se le aplican distintos desplazamientos temporales antes de clasificar. El resultado se resume en el siguiente gráfico.

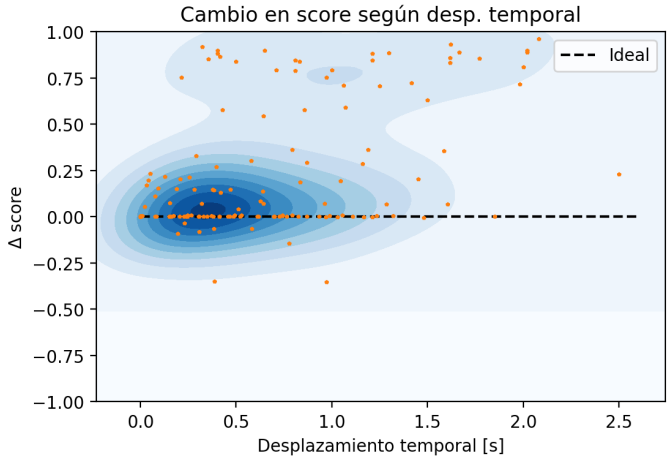


Figura 4.2: Cambio de puntaje frente a desplzamientos en modelo *baseline*.

De la figura 4.2 se observa que efectivamente existen múltiples muestras del dataset donde al introducir un desplazamiento varía considerablemente su puntaje, lo que genera un cambio en la etiqueta dada por el modelo. Por lo tanto, resultó necesario buscar una forma de reducir o eliminar este fenómeno.

En [35] se propuso que las capas lineales completamente conectadas que se suelen utilizar en redes convolucionales para tareas de clasificación son propensas a sobreajsutarse y hacen que el clasificador sea menos robusto a traslaciones espaciales, en el caso de imágenes. Como alternativa se propone reemplazarlas por una capa convolucional donde se obtenga un mapa de características por clase y luego se aplique un promedio global en cada uno para obtener el puntaje de clasificación. De la misma manera también se puede obtener el máximo de cada mapa para utilizar como el puntaje.

Este enfoque tiene varias ventajas, partiendo por mejorar el rendimiento frente a traslaciones de las características, que es lo que ocurre cuando un disparo ocurre en un momento distinto. Además, el último mapa de características de la red es más interpretable pues resalta la zona que aporta más al puntaje de clasificación. La operación de obtener el promedio global

se le denomina *Global Average Pooling(GAP)* en la literatura, mientras que aplicar el máximo se conoce como *Global Max Pooling(GMP)*

Luego, se procedió a modificar la estructura de la red utilizada, reemplazando las dos capas lineales del final por una convolucional con kernel de tamaño 1x1. Primero se entrenó un modelo utilizando *GAP* donde hubo una mejora en el fenómeno analizado, pero no tan significativa. Luego se entrenó otro con *GMP*, que obtuvo mejores resultados. A continuación se muestran las métricas obtenidas por este último modelo, junto con la arquitectura utilizada en la figura 4.3.

Tabla 4.2: Métricas modelo GMP

Métrica	Puntaje
Precisión	1
Recall	0.938
PR-AUC	0.955

Figura 4.3: Arquitectura modelo *GMP*.

En primer lugar, se puede ver que mejoró la precisión y el puntaje PR-AUC respecto del primer modelo. Luego, se repitió el experimento de evaluar disparos sometidos a distintos desplazmaientos temporales, obteniendo el gráfico de la figura 4.4. Se puede notar una reducción notoria de la dispersión.

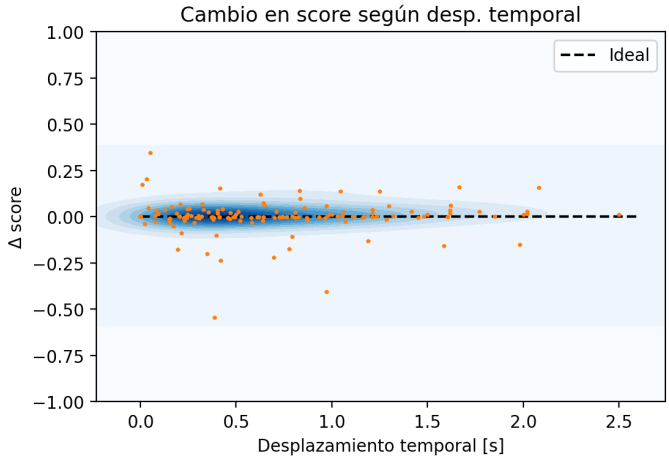


Figura 4.4: Cambio de puntaje frente a desplzamientos en modelo *GMP*.

Posteriormente, se intentó mejorar el rendimiento probando arquitecturas más modernas de redes convolucionales. Tal es el caso de la expuesta en [36], donde fundamentalmente proponen la convolución separable en profundidad. Usualmente un kernel convolucional para

una entrada bidimensional tiene tres dimensiones, el alto, el ancho y el número de mapas o profundidad de la entrada. Al separar la convolución en una exclusiva de profundidad unida a otra que solo abarca el alto y ancho, afirman que se puede lograr el mismo rendimiento pero con un menor número de parámetros. Esto permite, por ejemplo, aumentar la profundidad de las redes sin aumentar en gran medida la cantidad de parámetros a entrenar.

Creando una red basada en convoluciones separables y manteniendo el operador *GMP*, se llegó a una arquitectura que tiene nueve capas convolucionales de profundidad, frente a las cuatro de los otros modelos (ver figura 4.5). También contiene 8.265 parámetros que es menor a los 154.817 que poseía el modelo propuesto con *GMP*. Las métricas obtenidas para esta red se muestran a continuación, refiriéndose a ella como modelo *GMP-Xception*.

Tabla 4.3: Métricas modelo GMP-Xception

Métrica	Puntaje
Precisión	1
Recall	0.938
PR-AUC	0.993

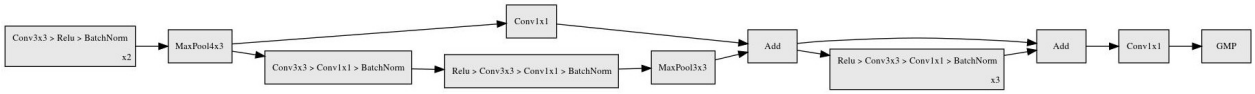


Figura 4.5: Arquitectura modelo *GMP-Xception*.

Como se observa de estos resultados, se obtuvo un aumento en la métrica de PR-AUC. Esto sumando al menor numero de parámetros permite elegir este modelo para utilizar posteriormente en el módulo diseñado para clasificar los sonidos entrantes al sistema. También se repite el experimento sobre el efecto del desplazamiento de los disparos del set de desarrollo, como se ve en la siguiente figura, y se verifica un comportamiento similar al modelo anterior con sólo *GMP*.

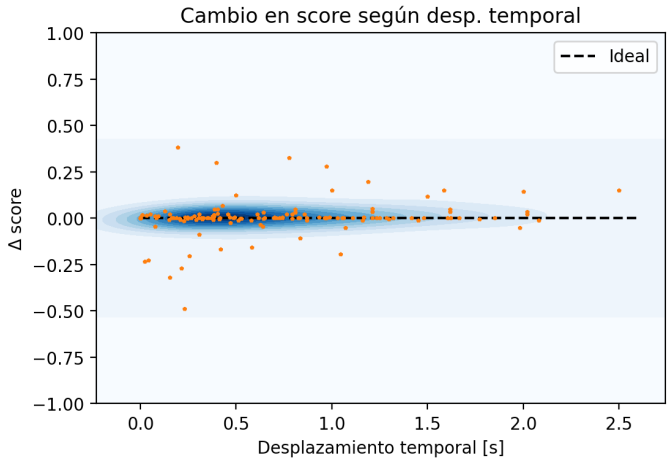
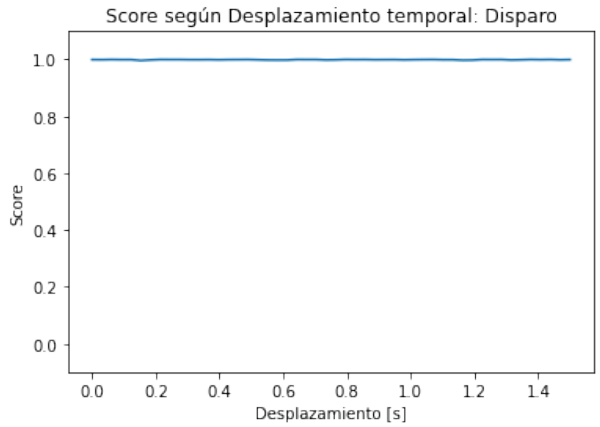
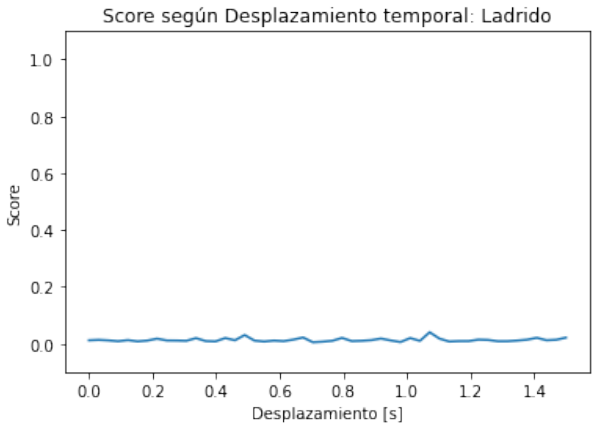


Figura 4.6: Cambio de puntaje frente a desplzamientos en modelo *GMP-Xception*.

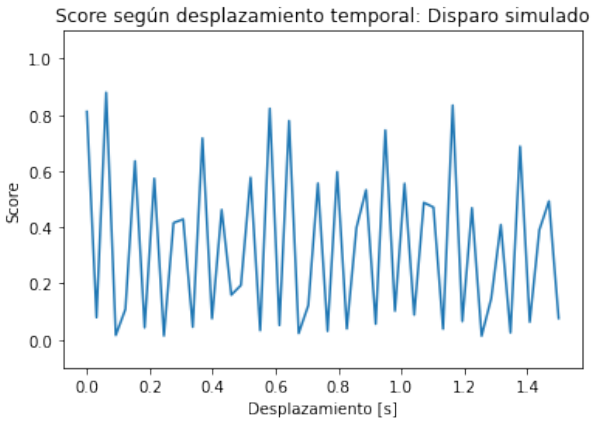
Pese a lo anterior, al avanzar el desarrollo y probar este clasificador con los disparos simulados, se observó que el problema del desplazamiento temporal no estaba del todo solucionado. Como ejemplo, se muestra en la figura 4.7 como se comporta un disparo y el ladrido de un perro del conjunto de desarrollo, junto con un disparo simulado. Se puede notar que los primeros dos muestran un comportamiento estable frente al desplazamiento, pero en el disparo simulado se genera un comportamiento oscilatorio, donde en momentos se clasificaría como disparos y en otros no.



(a) Ejemplo de disparo



(b) Ejemplo de ruido



(c) Ejemplo de disparo simulado

Figura 4.7: Efecto de desplazamiento temporal en *score*.

Como segunda medida para solucionar este problema, se propuso realizar la clasificación de una muestra de audio múltiples veces, desplazando el origen del audio cinco veces a la derecha en el tiempo. Esto genera un arreglo de cinco puntajes, donde se elige el máximo como el puntaje a considerar, de una forma similar a como funciona *GMP*. Bajo este esquema, el comportamiento del disparo simulado queda como en la siguiente figura, donde se disminuye en gran medida la amplitud de la oscilación del puntaje, y consistentemente logra ser clasificado como disparo, considerando un umbral de 0.5.

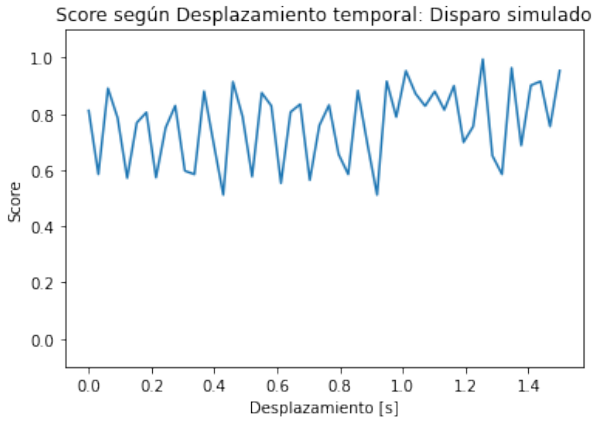


Figura 4.8: Efecto de desplazamiento en disparo simulado con corrección.

También, se probo el rendimiento del modelo frente al ruido ambiente. Para ello, se emplearon cuatro muestras de ruido que son martilleo en la calle, vehículos, calle con personas y parque. Estos se sumaron a los ejemplos del set de *test* a distintos niveles de SNR y midiendo PR-AUC en cada caso, obteniendo el resultado del siguiente gráfico.

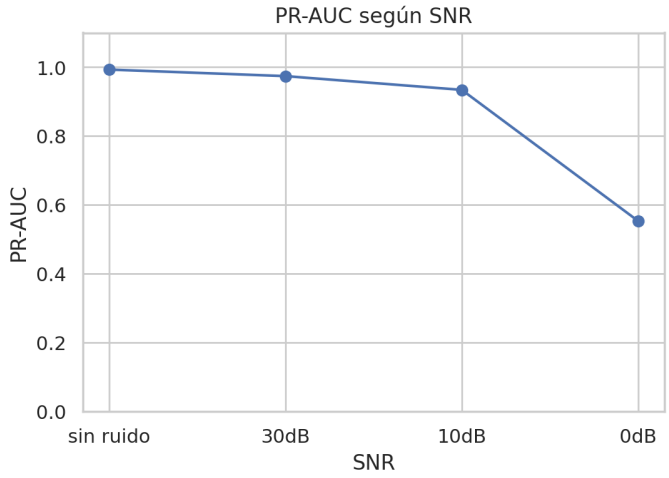


Figura 4.9: Puntaje PR-AUC frente a distintos niveles de SNR en el modelo *GMP-Xception*.

Como era de esperar, el rendimiento del modelo disminuye frente al aumento del ruido ambiente. Sin embargo, la caída no es demasiado pronunciada y se ve que incluso con 0 SNR el modelo todavía es capaz de distinguir disparos.

4.3. Pruebas del sistema

A continuación se muestran los resultados de distintas pruebas y experimentos realizados sobre los módulos desarrollados para el sistema de clasificación y localización de disparos propuesto.

4.3.1. Módulo de localización

Antes de realizar la prueba sobre el sistema completo, se realizaron experimentos propios para el módulo de localización. En específico, se quiso comprobar el efecto de la diferencia entre la altura de los micrófonos y la fuente de sonido. Esto es relevante pues se plantea entregar una estimación en dos dimensiones, pero en la practica los micrófonos estarán a distintas alturas frente al origen del sonido debido a la posición de la instalación o la geometría de la ciudad.

Se desarrollaron múltiples casos donde se varía la diferencia entre la altura de los receptores y la señal a localizar. En todos ellos se resuelve mediante multilateración hiperbólica minimizando el error cuadrático medio de la estimación como se propuso en la metodología.

Como primer caso, se estudió lo que ocurre si todos los receptores del arreglo están a una misma distancia del origen de una señal que ocurre a una altura de 1.5 [m]. En este, se varió la distancia a la que se encuentran de 1 a 100 metros y también la altura de instalación, partiendo por la misma de la señal y luego variando a 4, 8 y 12 metros. Esta situación se muestra en la siguiente figura.

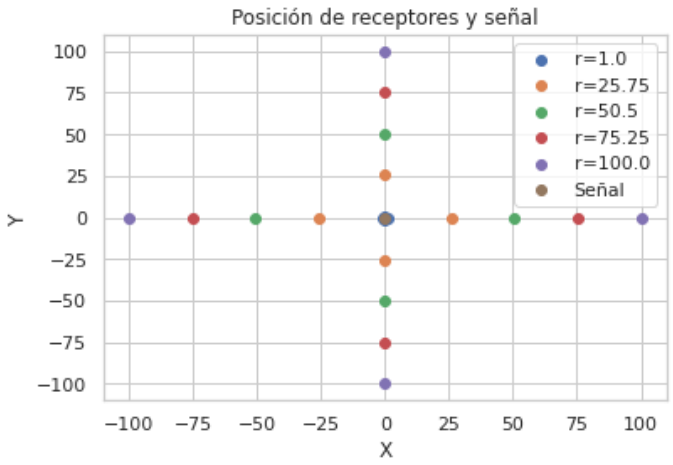


Figura 4.10: Posición receptores y señal de primer caso.

El error de localización para este caso se muestra en la figura 4.11. En esta configuración resulta que el error de localización es prácticamente independiente de la altura a la que se instalen los receptores y también de la distancia a la que se encuentran de la señal. Esto se puede entender por la simetría presente en este caso. Esta ocasiona que el centro del grupo de receptores posea el mínimo error cuadrático al resolver la multilateración hiperbólica.

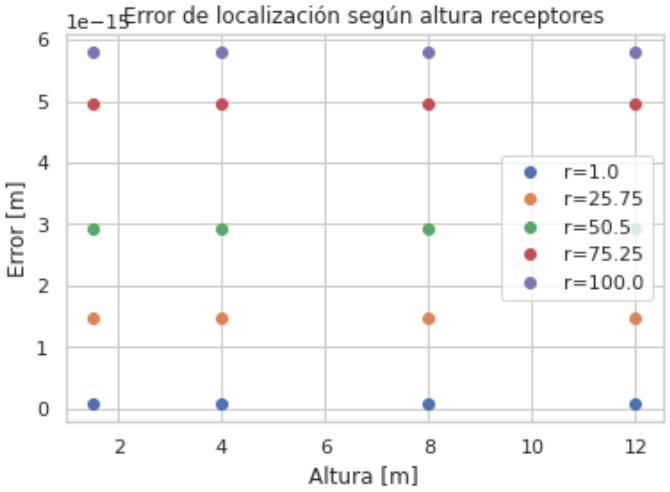
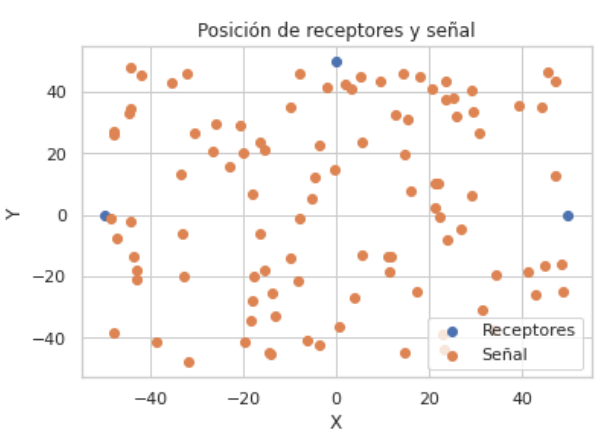
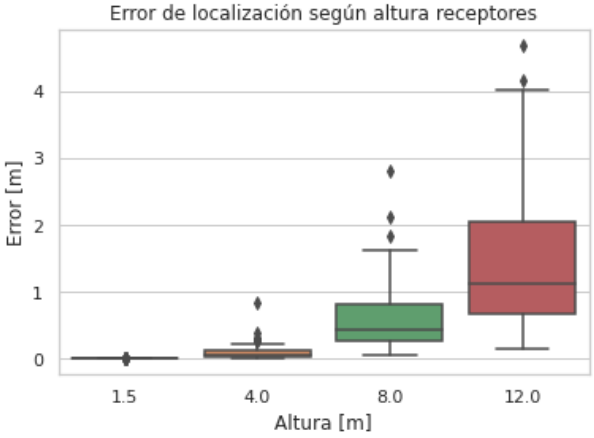


Figura 4.11: Error de localización según altura de receptores de primer caso.

Como segundo caso, se rompió la simetría dejando el arreglo de receptores fijos, en X e Y, y se varía la posición de la fuente de sonido en forma aleatoria. Esto resulta en el escenario mostrado en la figura 4.12.a . La altura de instalación de los receptores varió de forma igual al caso anterior y se obtiene el siguiente resultado.



(a) Posición receptores y señal

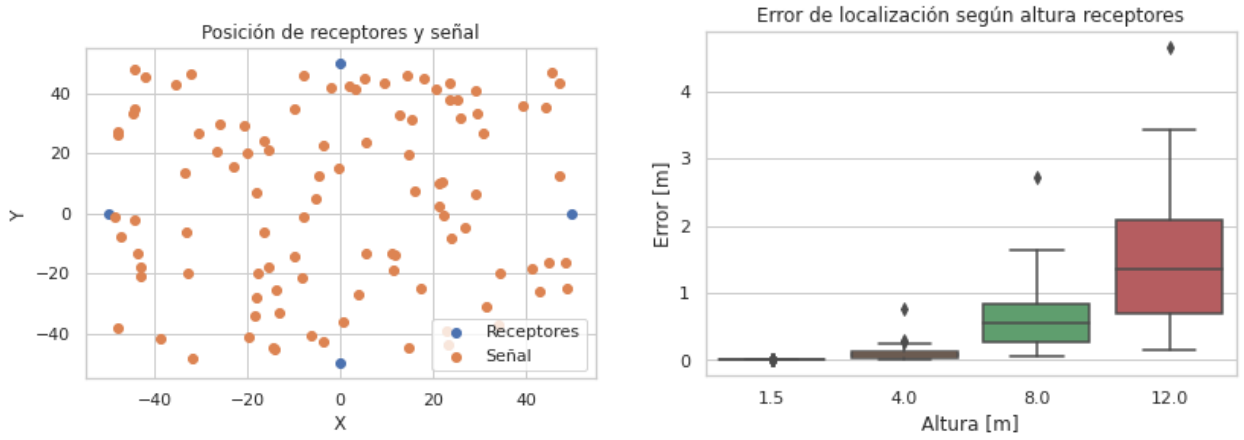


(b) Error de localización según altura de receptores

Figura 4.12: Experimento de localización según altura de receptores, segundo caso.

Viendo el resultado del la figura 4.12.b, se puede inferir que existe una tendencia al aumento del error cuando incrementa la diferencia entre la altura del origen de la señal y los receptores. También se observa que aumenta la varianza del error según aumenta la altura de instalación.

Como tercer caso se quiso comprobar si poseer un micrófono adicional reduce el error obtenido. Se repite entonces el escenario anterior, pero se agrega una cuarto micrófono como se ve en la figura 4.13. La situación no cambia demasiado, pero si se nota una reducción de los peores casos cuando los receptores están a 12 metros.

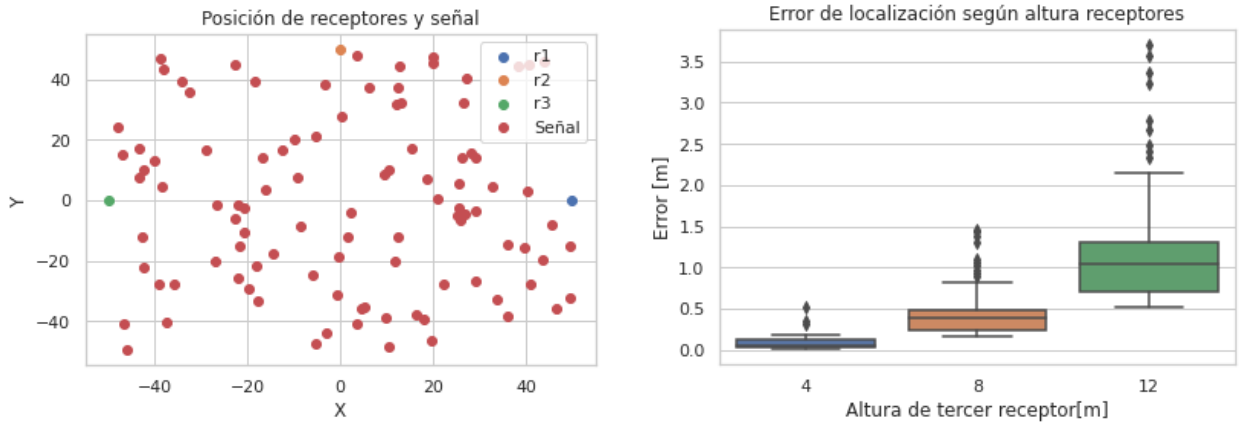


(a) Posición receptores y señal

(b) Error de localización según altura de receptores

Figura 4.13: Experimento de localización según altura de receptores, tercer caso.

Como cuarto caso se vió el efecto de que uno de los micrófonos este a una altura distinta de los demás. Este escenario se podría producir si los receptores están instalados en postes de la calle y uno se instala en la cornisa de un edificio o esta en una calle a distinta altura. Los resultados de este experimento se ven en la figura 4.14.



(a) Posición receptores y señal

(b) Error de localización según altura de receptores

Figura 4.14: Experimento de localización según altura de receptores, cuarto caso.

De este caso se puede ver que el error igual aumenta según la altura del tercer micrófono. Pero como se podría esperar, la mediana del error es menor al caso donde todos los micrófonos se elevan respecto a la fuente de sonido.

4.3.2. Pruebas generales

Finalmente se probó el sistema completo, utilizando los audios simulados a partir del entorno simple y complejo que se propusieron. Para ello, primero se extendieron los audios a una duración de diez segundos y cada prueba se repitió diez veces, dejando el evento de disparo cerca del medio. Se creó una instancia de máquina virtual en *Azure* de cuatro núcleos y 8 Gb

de RAM para ejecutar los módulos del sistema, mientras que los dispositivos en terreno fueron simulados de forma local.

4.3.2.1. Entorno simple

La primera prueba del sistema completo ejecutándose en la máquina virtual se realizó con el entorno simple. Para este fue necesario simular cuatro dispositivos y a cada uno se le envió un audio junto con el instante para iniciar la simulación. En 10 intentos se midió el tiempo de localización, entre que se origina el disparo y se obtiene la posición, y el error de ésta, que se muestran en la siguiente tabla.

Tabla 4.4: Resultados promedio de prueba en entorno simple

Métrica	Resultado
Tiempo de localización	22.7 [s]
Error de localización	0.45 [m]

Se puede destacar que el tiempo promedio en obtener la posición es inferior a 30 segundos, comparable con los sistemas comerciales. Por otro lado el error en la estimación es menor a un metro, también dentro de lo que otorgan los sistemas comerciales y suficiente para ubicar donde se origina el disparo. Aún así, esta prueba tiene ciertas condiciones ideales como una conexión estable entre los dispositivos y el servidor, además de poseer nulo ruido ambiental.

En el siguiente gráfico se puede ver la comparación entre la posición real del disparo y la estimación obtenida en los 10 intentos. Se puede notar que ésta varia entre dos puntos, comportamiento que se da porque en ciertos casos se estima la posición con los cuatro dispositivos, y en otros hay uno cuya detección llega más tarde y no alcanza a ser considerado.

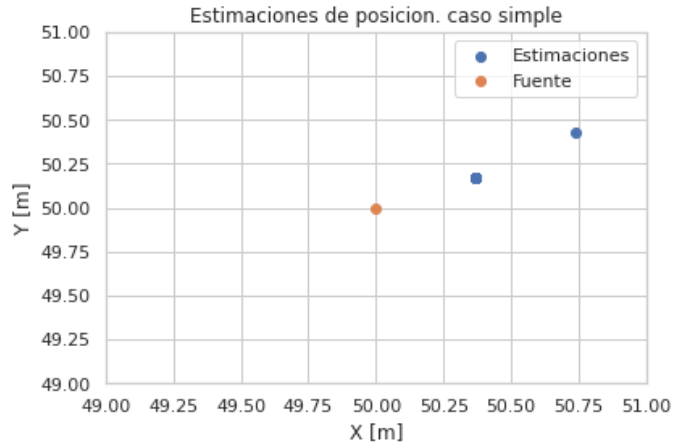


Figura 4.15: Resultado localización de disparo en entorno simple.

Como se vió en los experimentos del módulo de localización, el error depende también de la altura en que están los micrófonos respecto del disparo. Resolviendo la multilateración hiperbólica con las ubicaciones exactas de los elementos para este escenario, resulta una posición estimada en (50.34, 50.24) con un error de 0.42 [m]. Esto es prácticamente igual a lo

obtenido utilizando los audios simulados con el sistema completo, lo que muestra el correcto funcionamiento de este.

4.3.2.2. Entorno complejo

En el entorno complejo se debieron simular 8 dispositivos y tres disparos, que suman un total de 24 audios que fueron enviados. De los tres disparos, sólo uno fue localizado, puesto que los últimos dos ocurrían en ubicaciones donde las estructuras obstruían la línea de visión de los micrófonos y no se alcanzaba el mínimo necesario de tres para realizar la localización. A continuación se muestra el resultado promedio del primer disparo.

Tabla 4.5: Resultados promedio de prueba en entorno complejo, disparo 1

Métrica	Resultado
Tiempo de localización	28.6 [s]
Error de localización	0.52 [m]

Tanto el tiempo promedio como el error de localización es comparable con el caso anterior del entorno simple. Este resultado se da a pesar de la existencia de más reflexiones en comparación al otro caso, y resulta gracias a que se utiliza la detección del ruido impulsivo para determinar el instante donde ocurre el disparo. A continuación se muestra también las posiciones estimadas en los diez intentos.

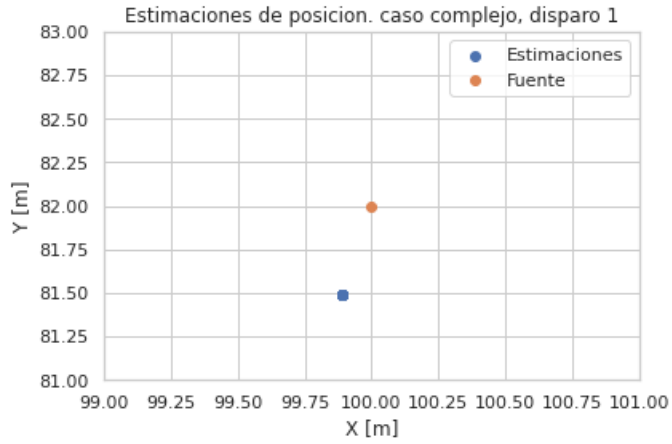


Figura 4.16: Resultado localización de disparo en entorno complejo.

En comparación con el entorno simple, la posición fue más consistente, lo que se puede deber al número y la posición de los micrófonos. Para este caso, el resultado de la localización utilizando la ubicación exacta de los elementos de este escenario y sin considerar obstáculos, se obtiene una estimación en (100.02, 82.01) con un error de 0.02[m]. Luego, el error introducido por el sistema es de 0,5 [m] y se puede deber en parte a que no todos los micrófonos tienen linea de visión directa a la fuente de sonido, por lo que no se pueden utilizar todos al mismo tiempo para localizar.

Capítulo 5

Conclusión

Como resultado de este trabajo se consiguió diseñar un sistema de detección y localización funcional. Este es capaz de trabajar a casi tiempo real y logra distinguir sonidos de disparos mediante una red convolucional. Además cada parte que lo compone fue probada individualmente y luego en conjunto mediante experimentos y simulaciones.

Lo que se puede destacar del sistema es que alcanza un rendimiento con tiempos de respuesta cercanos a 30 segundos y un bajo error de localización, similar a los sistemas comerciales que existen actualmente como ShotSpotter que entrega una alerta en menos de 60 segundos[37]. También se puede destacar el diseño basado en tecnologías actuales de IoT y computación en la nube, otorgando una mayor simplicidad y flexibilidad para su implementación. Al no depender de la instalación de un servidor físico, disminuye el costo de mantenimiento del sistema y se puede escalar según la necesidad.

Esto le permitiría la posibilidad de abarcar amplias zonas para cumplir la labor de notificar prontamente cuando ocurran disparos. También permitiría la fácil recopilación de datos que pueden servir para mejorar las políticas publicas de seguridad urbana. Además, aprovechando tecnologías de conexión inalámbrica para los dispositivos en terreno se pueden eliminar el costo de instalación de cableado y el costo de reparaciones de las lineas que eso conlleva eventualmente.

Por lo citado anteriormente, se considera cumplidos los tres objetivos específicos planteados originalmente. Estos fueron la elaboración de entornos de simulación para evaluar el sistema, la implementación del modelo de clasificación de sonidos y el diseño del sistema utilizando herramientas de Microsoft Azure.

Junto con lo anterior, también se pueden nombrar algunas posibles mejorar para el sistema. Por ejemplo, se podría implementar una eucarística mejor para agrupar las detecciones que llegan de los micrófonos para que si una detección llega más tarde, por retrasos en la comunicación, se pueda utilizar para mejorar la precisión de una localización ya efectuada. También se podría investigar el beneficio de utilizar técnicas de correlación de señales para asegurar que se agrupen detecciones de un mismo evento.

Otra mejora que puede quedar como trabajo futuro es utilizar más datos para entrenar el modelo de clasificación e intentar mejorar más el rendimiento frente a ruido ambiental. Junto con lo anterior, también queda pendiente experimentar con un prototipo físico para evaluar el

rendimiento con el hardware que se utilizaría.

Bibliografía

[1] J. M. Ojeda. (feb. de 2020). *"Balas locas" en la zona sur de la Región Metropolitana: 6 víctimas y ningún detenido*, [En línea]. Disponible: https://www.latercera.com/nacional/noticia/balas-locas-la-zona-sur-6-victimas-ningun-detenido/1001001/.

[2] Instituto Nacional de Estadísticas. (mayo de 2019). *Síntesis de resultados XV Encuesta Nacional Urbana de Seguridad Ciudadana*, [En línea]. Disponible: https://www.ine.cl/docs/default-source/seguridad-ciudadana/publicaciones-y-anuarios/2018/s%C3%ADntesis-de-resultados-xv-enusc-2018.pdf?sfvrsn=11af55b6_2.

[3] J. R. Aguilar, "Sistemas de detección de disparos: ¿son eficaces para controlar la violencia con armas de fuego en América Latina?", *URVIO. Revista Latinoamericana de Estudios de Seguridad*, n.º 23, pp. 128-141, 26 de nov. de 2018.

[4] D. Gendzwill, "Locating cannons by sound ranging in World War I", *The Leading Edge*, vol. 26, n.º 1, pp. 27-29, ene. de 2007.

[5] C. Williams. (mayo de 2017). *How ShotSpotter locates gunfire, helps police catch shooters and works to 'denormalize' gun violence*, [En línea]. Disponible: https://www.washingtonpost.com/news/true-crime/wp/2017/05/10/how-shotspotter-locates-gunfire-helps-police-catch-shooters-and-denormalize-gun-violence/.

[6] K. Hillmann. (ago. de 2017). *Biobío: instalan detectores de disparos*, [En línea]. Disponible: https://www.latercera.com/noticia/biobio-instalan-detectores-disparos/.

[7] J. Carroll. (jun. de 2019). *Pros And Cons Of San Diego's Gunshot Detection System*, [En línea]. Disponible: https://www.kpbs.org/news/2019/jun/19/pros-and-cons-gunshot-detection-system/.

[8] R. Muggah e I. Szabó de Carvalho. (abr. de 2017). *Existe una cura para la epidemia de homicidios en América Latina, y no requiere más policías o cárceles*, [En línea]. Disponible: https://es.weforum.org/agenda/2017/04/existe-una-cura-para-la-epidemia-de-homicidios-en-america-latina-y-no-requiere-mas-policias-o-carceles/.

[9] (Mayo de 2015). *Sistema de detecção de tiros não funciona no Guajuviras*, [En línea]. Disponible: https://jornaltimoneiro.com.br/index.php/2015/05/08/sistema-de-deteccao-de-tiros-nao-funciona-no-guajuviras/.

[10] Microsoft. (2020). *Conozca Azure*, [En línea]. Disponible: https://azure.microsoft.com/es-es/overview/.

[11] J. Aguilar, "Gunshot Detection Systems in Civilian Law Enforcement", *Journal of the Audio Engineering Society*, vol. 63, n.º 4, pp. 280-291, abr. de 2015.

[12] W. Moebs, S. J. Ling y J. Sanny. (sep. de 2016). *Sound*, [En línea]. Disponible: https://openstax.org/books/university-physics-volume-1/pages/17-2-speed-of-sound.

[13] R. C. Maher, "Acoustical Characterization of Gunshots", en *2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, abr. de 2007, pp. 1-5.

[14] M. Vorländer, *Auralization*, ép. RWTHedition. Berlin, Heidelberg: Springer, 2008.

[15] J. Picaut y N. FORTIN, "SPPS, a particle-tracing numerical code for indoor and outdoor sound propagation prediction", abr. de 2012.

[16] D. Schröder, "Physically Based Real-Time Auralization of Interactive Virtual Environments", Tesis doct., RWTH Aachen University, Aachen, 2011.

[17] A. Bensky, "Time of Arrival and Time Difference of Arrival", en *Wireless Positioning Technologies and Applications*, ép. GNSS technology and applications series, Second, Artech House, 2016, pp. 192-196.

[18] F. Silva y W. Alves, "Robust TDOA-Based Sound Source Localization", sep. de 2015.

[19] I. Kauppinen, "Methods for detecting impulsive noise in speech and audio signals", en *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, vol. 2, Santorini, Greece: IEEE, 2002, pp. 967-970.

[20] N. Shreyas, M. Venkatraman, S. Malini y S. Chandrakala, "Chapter 7 - Trends of Sound Event Recognition in Audio Surveillance: A Recent Review and Study", en *The Cognitive Approach in Cloud Computing and Internet of Things Technologies for Surveillance Tracking Systems*, ép. Intelligent Data-Centric Systems, Academic Press, ene. de 2020, pp. 95-106.

[21] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification", en *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia: ACM Press, 13 de oct. de 2015, pp. 1015-1018.

[22] J. Salamon y J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification", *IEEE Signal Processing Lett.*, vol. 24, n.º 3, pp. 279-283, mar. de 2017.

[23] J. Salamon, C. Jacoby y J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research", en *Proceedings of the ACM International Conference on Multimedia - MM '14*, Orlando, Florida, USA: ACM Press, 2014, pp. 1041-1044.

[24] V. Dumoulin y F. Visin, "A guide to convolution arithmetic for deep learning", 2016. [En línea]. Disponible: arXiv: 1603.07285 [stat.ML].

[25] B. McFee, J. Salamon y J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection", 10 de ago. de 2018. [En línea]. Disponible: arXiv: 1804.10070 [cs.SD].

[26] M. Akay, "Wavelets in biomedical engineering", *Ann. of Biomedical Engineering*, vol. 23, n.º 5, pp. 531-542, sep. de 1995.

[27] M. E. Ylikoski, J. O. Pekkarinen, J. P. Starck, R. J. Pääkkönen y J. S. Ylikoski, "Physical characteristics of gunfire impulse noise and its attenuation by hearing protectors", *Scandinavian Audiology*, vol. 24, n.º 1, pp. 3-11, ene. de 1995.

[28] J. Picaut y N. Fortin, "I-Simpa, a graphical user interface devoted to host 3D sound propagation numerical codes", 2012.

[29] J. Lucio Naranjo, F. Castro Pinto, J. C. Torres y R. Tenenbaum, "Acoustic Simulator for Urban Noise Analysis", Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia - 30° CILAMCE, Armação dos Búzios, RJ, Brasil, 5 de nov. de 2009.

[30] Z. Xiangyang, C. Ke'an y S. Jincai, "On the accuracy of the ray-tracing algorithms based on various sound receiver models", *Applied Acoustics*, vol. 64, n.º 4, pp. 433-441, abr. de 2003.

[31] T. K. Routh y R. C. Maher, "Recording anechoic gunshot waveforms of several firearms at 500 kilohertz sampling rate", Salt Lake City, Utah, 2016.

[32] A. Tharwat, "Classification assessment methods", *Applied Computing and Informatics*, vol. ahead-of-print, ahead-of-print 1 de ene. de 2020.

[33] *Raspberry pi 4 model b specifications*, Raspberry Pi, [En línea]. Disponible: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/.

[34] *IoT edge | microsoft azure*, [En línea]. Disponible: https://azure.microsoft.com/en-us/services/iot-edge/.

[35] M. Lin, Q. Chen y S. Yan, "Network In Network", *arXiv:1312.4400 [cs]*, 4 de mar. de 2014. [En línea]. Disponible: arXiv: 1312.4400.

[36] F. Chollet, "Xception: Deep learning with depthwise separable convolutions", *arXiv:1610.02357 [cs]*, 4 de abr. de 2017. [En línea]. Disponible: arXiv: 1610.02357.

[37] *ShotSpotter technology*, ShotSpotter, [En línea]. Disponible: https://www.shotspotter.com/technology/.