PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

Diseño de un modelo algorítmico para la discriminación de patrones acústicos entre voces y pisadas humanas

Tesis para optar por el Título de Ingeniero Informático que presenta el bachiller:

Cecilia Viera Barthelmes

ASESOR: Dr. César Armando Beltrán Castañón

LIMA - 2019

Dedicatoria

A mi abuela Otilia, por todas sus oraciones en toda mi etapa de estudios, pidiéndole a Dios que siempre me vaya bien en los proyectos que presento.

A mi madre, por todos los ánimos que siempre me llenan de ganas para seguir avanzando, y por la confianza que tiene en mí y que me llena de seguridad. Gracias mamá por estar ahí todos estos años.

A mi padre, por haberme permitido la oportunidad de estudiar la carrera que me gusta en la mejor universidad del país. Has sido un ejemplo de responsabilidad y perserverancia.

Agradecimientos

Al Dr. César Beltrán, por haberme asesorado durante todo el tiempo que me tomó realizar este proyecto y brindado las herramientas y recursos necesarios para su elaboración.

A todos mis profesores, en especial al profesor Fernando Alva, por la paciencia y ayuda brindada siempre que lo necesitaba, y ayudarme a definir mis ideas y objetivos.

Resumen

Actualmente existe una gran demanda de soluciones innovadoras e informáticas que permitan generar sistemas de vigilancia o que ayuden en esta labor. Es así como se han generado diversos proyectos que buscan satisfacer las necesidades de sistemas de este tipo. Mayormente, se ha utilizado la tecnología de imágenes y utilizando drones o algún tipo de cámara, donde una persona está monitoreando estas imágenes captadas en tiempo real para verificar la presencia de un objeto o un ser no deseado. Sin embargo, estas soluciones han presentado una gran complejidad tanto en procesamiento como infraestructura, conllevando así también a un precio elevado de su implantación.

Es por esto que este proyecto de investigación se enfoca en presentar una solución a este problema utilizando recursos más simples, basándose en un reconocimiento de patrones en señales acústicas. Esta es un área de la especialidad de informática que en los últimos años ha tenido un gran desarrollo y estudio debido a las diversas aplicaciones que puede tener en el mundo contemporáneo. Cada vez se han ido perfeccionando los algoritmos de extracción de características y de aprendizaje de máquina, por lo cual en este trabajo se utilizarán y compararán dos métodos de caracterización estudiados en investigaciones de reconocimiento de voz. Además, se desarrollará un módulo de recorte de la señal que permita identificar a las regiones de interés. Finalmente, se usarán redes neuronales como el clasificador del algoritmo.

Índice de tablas

Índice de figuras

Índice General

ET LUX IN TENEBRIS LUCET
MCMXVII

1. Generalidades

1.1. Problemática

La seguridad y vigilancia es un tema cada vez más alarmante en el mundo actual. Dentro de los muchos problemas generados por la inseguridad y falta de vigilancia, se encuentra la vulneración de áreas naturales protegidas por el gobierno. Estas áreas albergan distintas especies animales, muchas veces en peligro de extinción o amenazadas por el ser humano, debido a la destrucción de sus ecosistemas o a su caza furtiva. Por lo general, son áreas sumamente extensas, que cuentan con guardabosques o algún tipo de personal designado a vigilar sus límites, con el fin de verificar que personas no autorizadas se mantengan fuera de estas áreas. Sin embargo, debido a su gran extensión y a los pocos recursos destinados a la seguridad en estas áreas, la vigilancia es una tarea difícil para las personas encargadas de esta labor. Específicamente para la protección de la fauna presente, las amenazas son cada vez más poderosas. Si vemos las estadísticas, tendremos un mejor panorama de esta creciente amenaza: entre los años de 2010 y 2012, se aproxima que más de 100 000 elefantes fueron ilegalmente cazados en las sábanas de África [SKINNER 2014].

Actualmente, hay algunos proyectos y tecnologías orientados a resolver los problemas que surgen por la deficiencia de los mecanismos de seguridad en áreas naturales. Durante el 2014 se inició un proyecto que incluía el uso de drones para el resguardo de los animales en África. Este proyecto consistía en el monitoreo de estos dispositivos, los cuales presentaban su posición y ruta seguida en un mapa y mostraban una imagen en alta calidad de la zona que estaban visualizando en tiempo real. De este modo, la persona controlando el dron podía dar aviso si avistaba a un cazador en una zona resguardada. [WALL 2014]

El Perú no es un país ajeno a este problema. Teniendo gran parte de la biodiversidad del planeta, dentro de nuestro territorio hay numerosas áreas protegidas pertenecientes a la sierra y selva. Cabe resaltar que su seguridad debe ser vista como una necesidad, pues la diversidad biológica sólo de áreas naturales aportan a la economía nacional anualmente 1 000 millones de dólares [INEI 2013]. Sin embargo, los recursos asignados al cuidado de estas áreas y su fauna son extremadamente pequeños en proporción al territorio y los peligros que representa su cuidado. Por esto, sus mecanismos de seguridad solo se basan en un grupo mínimo de personas encargadas del cuidado y protección de las áreas. Estas personas cuentan con medios de transporte para vigilar las áreas, y con medios que les permiten comunicarse entre ellos en caso encuentren algún tipo de actividad ilegal, pero al tener a cargo áreas de gran extensión como lo son las reservas naturales, no se dan abasto para monitorearlas entera y constantemente, e incluso ellos mismos se exponen ante peligros, como lo representan las armas de fuego que portan los

cazadores. Además, al ser un país en desarrollo y con menos recursos económicos de los que necesitamos la inversión en proyectos de equipos complejos y de alto costo como lo son el uso de drones, está fuera del alcance para el gobierno.

Entonces, el problema enfrentado es la falta de herramientas de apoyo de costo bajo en la detección inmediata de presencia humana en áreas de espacios abiertos. Actualmente, la mayoría de proyectos orientados a dar solución a este problema, utilizan medios visuales para la captura de datos del entorno [METCALFE 2017], lo que les sirve para identificar a una persona basándose en distintas características, como la figura del ser humano, o la imagen térmica generada por el mismo, dependiendo de qué tipo de dispositivo visual se esté utilizando. Sin embargo, existen otros medios para su reconocimiento más económicos y menos complejos, tales como los medios auditivos y de señales sísmicas. Cabe mencionar que los proyectos que se enfocan en medios visuales tienen varias limitantes, como lo es la necesidad de alto procesamiento para los archivos de imágenes, que el dispositivo tenga una alta resolución o que los datos tarden en llegar hasta el centro de monitoreo debido al peso de la información transmitida.

Continuando con este enfoque, este trabajo está orientado a desarrollar un algoritmo que analice señales acústicas para poder detectar patrones de audio en ella, tales como patrones de pisadas y voz, lo que permitiría identificar la presencia humana en una zona determinada. Cabe resaltar, que los dispositivos de captura de sonidos suelen ser menos complejos y requieren de menor infraestructura a diferencia de dispositivos de imágenes, además de necesitar una menor capacidad de procesamiento de datos. Estas características de la solución planteada, responderían de manera adecuada a las restricciones de costos y recursos, logrando obtener una herramienta que podría orientarse en un futuro a la detección específica de cazadores.

1.1.1. Objetivo General

Implementar un algoritmo de discriminación de patrones de audio, basado en el análisis de señales acústicas, orientado al reconocimiento de voces y pisadas humanas.

1.1.2. Objetivos Específicos

- Definir el modelo de caracterización de patrones de voces y pisadas previamente definidos en la señal acústica.
- Desarrollar el modelo de clasificación de patrones de audios en las señales acústicas de acuerdo a las clases de patrones pre-definidos.
- Implementar una interfaz que muestre los resultados del algoritmo propuesto referente a un set de datos ingresado y que nos permita comprobar los resultados obtenidos.
- Estimar la precisión y eficiencia de las etapas de entrenamiento y clasificación del algoritmo.

1.1.3. Resultados esperados

- Modelo de pre-procesamiento del audio para el recorte y obtención de las señales acústicas significativas.
- Modelo algorítmico para la extracción de características de patrones predefinidos en la señal acústica con las características más relevantes y significativas para la clasificación.
- Modelo de clasificación de patrones de voces y pisadas en señales acústicas basado en algoritmos de aprendizaje supervisado
- Prototipo de la interfaz de prueba y resultados del algoritmo referente a datos capturados en ambientes reales y no controlados
- Análisis estadístico y comparativo de la medición de la precisión y eficiencia del algoritmo entre los resultados de datos experimentales y los capturados en escenarios del mundo real.

1.2. Herramientas, métodos y metodologías

1.2.1. Introducción

La sección presentada tiene como propósito establecer los métodos y herramientas a utilizarse por cada etapa del proyecto de modo que se pueda alcanzar el objetivo general previamente definido.

A continuación, se presenta una tabla con cada uno de los resultados esperados con las respectivas herramientas y métodos a utilizarse para conseguir cada uno de ellos.

Resultado Esperado	Herramientas y métodos
RE1: Modelo de pre-procesamiento de la señal para el recorte e identificación de sonidos significativos	• Matlab • Método de extracción de envolvente de la señal
RE2: Modelo algorítmico para la caracterización de patrones predefinidos en la señal acústica.	• Transformada de ondículas • Filtros Wiener • Transformada de Fourier • Coeficientes cepstrales de la frecuencia de Mel • MATLAB • Visual Studio • Octave
RE3: Modelo de clasificación de patrones de audios en señales acústicas	• Máquina de vectores de soporte • Modelo escondido de Markov • Visual Studio
RE4: Prototipo de la interfaz de prueba y resultados del algoritmo	• Visual Studio
RE5: Reporte estadístico de medición de la precisión y eficiencia de las diferenciación de pisadas de acuerdo a los diferentes escenarios a ser propuestos en el plan de pruebas.	• Método de validación cruzada

Tabla 1.1 Herramientas y métodos utilizados por resultado esperado

1.2.2. Herramientas

1.2.2.1. Matlab

Matlab es una plataforma optimizada para resolver problemas de ingeniería y ciencias. Cuenta con herramientas de gráficos para que sea más fácil visualizar la data y sus resultados, así como una amplia librería de funciones utilitarias que permiten una mejor investigación y experimentación de un tema específico. [MATHWORKS 2016]

Matlab es la combinación de laboratorio de matrices (en inglés, Matrix Laboratory) y es un software que fue construido principalmente sobre el uso de vectores y matrices. Esto lo hace particularmente útil para álgebra lineal, pero aparte también es una excelente herramienta para resolver ecuaciones diferenciales y algebraicas, e integración numérica. También tiene su propio lenguaje de programación, el cual es uno de los más fáciles para escribir programas matemáticos. Además, cuenta con paquetes de funciones utilitarias para procesamiento de señales, de imágenes, optimización, etc. [SANDBERG 2003]

Debido a la facilidad que provee MATLAB para utilizar funciones matemáticas complejas sobre un determinado set de datos de entrada, se decidió usarla para la etapa de pre-procesamiento y extracción de características de las señales acústicas.

1.2.2.2. Octave

Octave es un lenguaje de alto nivel, principalmente orientado a resolver cálculos numéricos. Cuenta con una interfaz gráfica y su lenguaje es compatible con Matlab en su mayor parte. Además también cuenta con distintas herramientas para resolver problemas específicos de ecuaciones diferenciales, integración de funciones, manejo de polinomios y matrices, etc. [OCTAVE 2017]

Una de las principales razones por las cuales se decidió usar Octave es que la distribución del software es gratuita, lo que facilita el trabajo de experimentación. Además al ser compatible con Matlab nos permite usar el mismo código en ambos programas. Por un lado, Matlab fue necesario para realizar el pre-procesamiento de la señal usando señales envolventes e implementar uno de los métodos de caracterización, mientras que Octave se utilizó para la implementación del segundo método de extracción y generación del vector de características.

1.2.2.3. Visual Studio

Visual Studio es un juego de herramientas para crear software, desde la fase de planeamiento y a través del diseño, codificación, prueba, análisis de la calidad del código y desempeño. Estas herramientas fueron diseñadas para trabajar unidas y son expuestas mediante el entorno de desarrollo integrado de Visual Studio. Se pueden crear distintos tipos de aplicaciones, como juegos, sitios Web, servicios Web, aplicaciones de gráficos, etc. Además, Visual Studio provee de soporte para una gran cantidad de lenguajes de programación, entre los que se encuentran: C#, C, C++, Javascript y Visual Basic. [MICROSOFT 2015]

MATLAB cuenta con una aplicación que permite llamar a las funciones y programas creados en otros lenguajes de programación, tales como C, C++ y Fortran. Debido a que Visual Studio cuenta con soporte para distintos lenguajes de programación, y cuenta con un entorno amigable y fácil de utilizar, se decidió utilizar esta herramienta para el desarrollo, implementación y prueba del algoritmo.

1.2.3. Métodos

1.2.3.1. Envolvente de la señal

Debido a que se pretende poder identificar en un audio los tiempos entre los cuales se ubica uno de los sonidos buscados (pisadas o voces), se necesita de una función que nos permita identificar las variaciones de la señal, identificando los picos y las regiones que representan a un sonido relevante. Para esto, se utilizarán dos funciones que generan una señal envolvente pertenecientes al conjunto de herramientas de procesamiento de señales provistos por Matlab. En el primer caso, es una señal que recorre todos los picos en el dominio de la frecuencia, mientras que para el segundo caso, es una señal que se basa en la raíz cuadrada promedio del muestreo especificado.

1.2.3.2. Transformada de ondículas

Las transformadas de ondículas son descomposiciones de múltiples resoluciones que pueden ser usadas para analizar tanto señales de audio como visuales. Describen una señal por la energía en cada escala y posición. [XU et al. 1994] Estas han sido usadas en las investigaciones de los últimos años como un método para la reducción del ruido en toda clase de señales. Los métodos suelen estar clasificados en dos categorías: reducción de ruido en el dominio original de la señal, y en el dominio transformado. Es en esta última categoría donde se encuentra la transformada de ondículas (Wavelet Transform), la cual es apropiada para casos donde se analiza una señal no estacionaria y se desea obtener características en el dominio de tiempo-frecuencia. [PATIL 2015]

Las transformaciones de ondículas descomponen una señal en una colección de “bandas de frecuencia” (referidas como escalas) al proyectar la señal en un elemento de un set de funciones básicas. Aunque las escalas no pertenecen a un dominio de frecuencia, la proyección de la señal en diferentes escalas es equivalente a pasarlas por un filtro pasabanda con un banco de filtros de constante-Q. Las funciones básicas son llamadas ondículas. Las ondículas son similares entre sí, variando solo en la dilación y traslación. [BARFORD Y OTROS 2012]

1.2.3.3. Filtros Wiener

El filtrado Wiener es una forma general de encontrar la mejor reconstrucción de una señal ruidosa. Aplica en cualquier función básica ortogonal, y de acuerdo a las funciones base devolverá distintos resultados. Este filtrado brinda una forma óptima de disminuir los componentes que son fuente de ruido, de modo que se pueda obtener la mejor reconstrucción de la señal original. Puede ser aplicado en funciones de base espacial, en base Fourier, o de ondículas. Primero debe

encontrarse cuál es la base más adecuada para el problema en particular, para sobre esta poder aplicar los filtros. [PRESS 2008]

Sin embargo, el uso de estos filtros podría causar efectos negativos en la señal si esta contiene voz, debido a que degrada la calidad o la hace ininteligible. A pesar de que el filtro logra disminuir el ruido en una cantidad considerable, esta cantidad es proporcional a la degradación de la voz. Dependiendo de la naturaleza de la aplicación, algunos sistemas de reducción de ruido podrían requerir de la señal de voz en alta calidad y soportar alta cantidad de ruido, mientras que otros podrían querer una mayor limpieza de la señal antes que poca distorsión. [CHEN 2006] Para el trabajo propuesto, se decidió usar este tipo de filtros pues la limpieza de datos tiene una mayor prioridad que mantener señales de voz claras.

1.2.3.4. Transformada de Fourier

La transformada de Fourier es una herramienta analítica principal para el análisis de señales y sistemas continuos y discretos en tiempo. La transformada es usada para estudiar una señal determinística y un proceso estacionario estocástico. Expresa la señal como una suma de las ondas del coseno y seno de un número infinito o finito de frecuencias. Sin embargo, al terminar todas las señales de frecuencias, la transformada de Fourier de una señal es considerada una transformada sin resolución de tiempo, lo que significa que no es posible determinar en qué momento una cierta componente de frecuencia está presente en la señal. Cada componente de la transformada de Fourier depende del comportamiento global de la señal. [GRIGORYAN 2005]

Como herramienta computacional encontramos la transformada rápida de Fourier, que permite calcular eficientemente la transformada discreta de Fourier de distintas muestras de data en el tiempo. Por eso, se define como una herramienta computacional que facilita el análisis de señales, como el análisis del espectro de energía y la simulación de filtros, mediante el uso de computadoras digitales. [COCHRAN 1967] Gracias a las distintas propiedades de la señal que la transformada de Fourier permite obtener, se utilizará este método para la extracción de características más relevantes en el análisis de los patrones de audio, sobretodo patrones como los ocasionados por las pisadas.

1.2.3.5. Coeficientes cepstrales de frecuencia de Mel

En el procesamiento de señales de audio, el cepstrum de la frecuencia de Mel es una representación de un espectro de corto plazo en una señal de audio, basado en una transformada lineal del coseno de un espectro de baja energía en una escala de Mel no-lineal de frecuencia. Los coeficientes extraídos son una de las características más utilizadas en reconocimiento de voz. [YUAN 2014]

Este método está basado en la variación del rango auditivo humano respecto a la frecuencia en su punto más crítico. Tiene dos tipos de filtros que están separados linealmente de la siguiente manera: por debajo de 1000 Hz, en una

baja frecuencia y por encima de los 1000 Hz, en un espaciamiento logarítmico. [MUDA 2010] A continuación se presenta un esquema de los pasos a seguir para realizar este método:

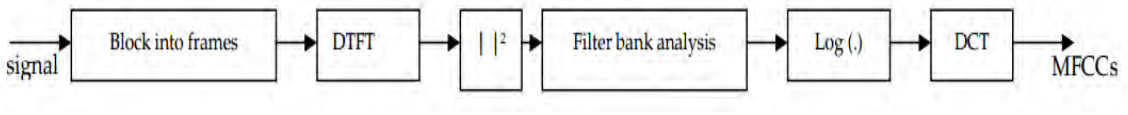


Figura 1.1.1 Etapas del método de los Coeficientes Cepstrales de frecuencia de Mel(MFCC) para la extracción de características. Imagen extraída de ABDALLA, Mahmoud y ALI, Hanaa, Wavelet-Based Mel-Frequency Cepstral Coefficients for Speaker Identification using Hidden Markov Models*, 2010, p.18.*

De acuerdo a este esquema, el primer paso es dividir la señal de audio en diferentes bloques cuidando que no haya problemas de solapamiento. El segundo paso es aplicar la Transformada Discreta de Tiempo de Fourier al segmento de la señal. El tercer paso es calcular el cuadrado del resultado del paso anterior. Las salidas de este paso son las energías del banco de filtros en escala Mel. El siguiente paso es calcular su logaritmo. Y por último, para calcular las constantes se aplica la transformada discreta del coseno a los logaritmos de las energías del banco de filtros. [ABDALLA y ALI 2010] Este método plantea implementarse para la identificación de patrones de voz durante el presente trabajo.

1.2.3.6. Máquina de vectores de soporte

Una máquina de vectores de soporte es un algoritmo computacional que aprende mediante ejemplos a asignar etiquetas a determinados objetos que recibe como entradas. Tiene una gran cantidad de aplicaciones en distintas áreas de ciencias. En esencia, es un algoritmo para maximizar una función matemática en particular con respecto a una colección de datos dada. Hay cuatro conceptos principales para entender este algoritmo: el hiperplano separador, el hiperplano de margen máximo, el margen indefinido, y la función núcleo. [NOBLE 2006]

Estos modelos son llamados no-paramétricos, lo que no significa que no tengan parámetros del todo. Por el contrario, su “aprendizaje” (selección, identificación, estimación, entrenamiento) es un tema crucial. Sin embargo, a diferencia de la inferencia clásica estadística, los parámetros no son predefinidos y su número depende de la data de entrenamiento usada. En otras palabras, los parámetros que definen la capacidad del modelo son dependientes de los datos, de tal forma que hacen equivalente la capacidad del modelo a la complejidad de la data. [WANG 2005]

Para este trabajo se decidió usar como parte de la etapa de clasificación máquina de vectores de soporte debido a la generalidad que esta tiene, ya que puede usarse para diversos tipos de aplicaciones como detección de rostros, clasificación de imágenes, reconocimiento de escritura, bioinformática, etc. [GUODONG 2014] Además, al ser un algoritmo de entrenamiento supervisado, los resultados de clasificación podrían mejorar de acuerdo a la cantidad y calidad de data que le sea proporcionado.

1.2.3.7. Método de Validación Cruzada

El método de validaciones cruzadas es uno de los más usados comúnmente para evaluar el desempeño predictivo de un modelo, que es dado a priori o desarrollado bajo un procedimiento de modelado. Básicamente, basado en la separación de data, parte de esta es usada para encajar en cada modelo competidor, y el resto de data es usada para medir el desempeño predictivo de cada modelo mediante la validación de errores, y el modelo con el mejor desempeño en general es seleccionado. Un problema fundamental en la aplicación de las validaciones cruzadas para la selección del modelo es la elección del ratio de separación de la data, o el tamaño de la data de validación. La elección óptima para la proporción de separación de la data depende si la data se encuentra bajo un marco paramétrico o no-paramétrico. Para el caso de un marco paramétrico, se quiere decir que el modelo de selección buscado se encuentra definido dentro del conjunto de modelos candidatos, mientras que en el segundo caso, el modelo buscado no es parte del conjunto de modelos planteados por lo que se debe aplicar una función de regresión que permita encontrarlo. [ZHANG y YANG 2015]

1.2.4. Metodología y plan de trabajo

De acuerdo al trabajo, se ha planteado el siguiente esquema para el desarrollo del algoritmo, con diferentes etapas en donde se utilizarán las herramientas y métodos previamente mencionados.

Figura 1.2 Representación de las etapas de la metodología a seguir

1. Generación de la base de datos de señales de audio
   Esta es la etapa preliminar del proyecto, en donde se generará una base de datos la cual contendrá los datos que se utilizarán como entrada en el posterior proceso de entrenamiento y validación del algoritmo clasificador. Durante esta etapa se definirá el formato en el que se guardarán las señales, así como etiquetarlas de acuerdo a su contenido.
2. Pre-procesamiento
   A esta etapa llega como entrada un audio de longitud variable donde se encuentran mezclados sonidos tanto de voces como de pisadas. Durante el pre-procesamiento se debe identificar las regiones de interés en esta señal de modo que se pueda realizar un recorte obteniendo audios de tiempo corto (menor a un segundo) donde se encuentre un sonido a clasificar.
3. Extracción de características
   Durante esta etapa se debe obtener un vector de características a partir de las señales preprocesadas provenientes de la etapa anterior. Estas características se definirán experimentalmente, de acuerdo a distintas

combinaciones para poder observar los resultados de cada una y determinar la mejor.

4. Entrenamiento
   Durante esta etapa se debe entrenar al algoritmo, para ello se utilizará un conjunto de datos destinado a ello. Este conjunto es el más grande dentro de la base de datos ya que pretende proveer al modelo de clasificación con la mayor cantidad de información posible, de modo que pueda obtener buenos resultados en la etapa siguiente, la cual es la etapa de clasificación.
5. Clasificación
   Esta es la última etapa del desarrollo del algoritmo, y recibe como entrada el vector con las características más relevantes de la señal. Luego, pasa a través de los distintos clasificadores propuestos, para finalmente entregar como resultado a qué clase pertenece el sonido. Durante esta fase, que se realiza inmediatamente después de la etapa de entrenamiento, se procede a probar el clasificador, obteniendo un porcentaje de aciertos y fallos para determinar respecto a esto cuál fue el mejor.

1.3. Delimitación

1.3.1. Alcance

El proyecto presentado en este trabajo se encuentra dentro del área de Ciencias de la Computación, y está orientado a la investigación aplicada. Por ello, implica el desarrollo de un algoritmo capaz de clasificar una señal de audio de acuerdo a la presencia de patrones de audio previamente definidos. Para ello se considerarán la identificación de dos tipos de patrones: pisadas y voces humanas.

El trabajo constará de tres módulos principales, los cuales se centran en el pre-procesamiento de la señal, la extracción de características principales para el análisis y correcta clasificación de la señal, y por último la clasificación de la misma. Del primer módulo, se busca obtener una señal libre de ruido, que esté segmentada de tal manera que el sonido del audio buscado, ya sea pisadas o voces, sea el más claro sobre otro tipo de sonidos. Del segundo módulo, se obtendrán cuáles son las características más representativas de la señal que permiten reconocer qué tipo de patrón se está analizando. Del último módulo, se busca obtener el mejor modelo clasificador de aprendizaje supervisado para la discriminación de las señales de entrada. De acuerdo a todo lo planteado, el presente trabajo busca brindar un algoritmo que pueda ser empleado en un futuro como la herramienta principal para la elaboración de un sistema completo de vigilancia, de modo que se pueda identificar qué tipo de actividades están ocurriendo en un determinado momento basado en señales de audio recibidas.

El presente trabajo estará orientado a lugares de espacios abiertos con las siguientes características: deben tener terrenos llanos de poca o nula vegetación y pocas fuentes de ruido, tales como ruido proveniente de animales, tránsito de vehículos terrestres o aéreos, y elevada actividad humana.

1.3.2. Limitaciones

Respecto a las limitaciones del proyecto, en primer lugar, las señales utilizadas para la etapa de entrenamiento y análisis serán capturadas en dos tipos de ambientes, cerrados y abiertos, asegurando que tengan una cantidad de ruido moderada de manera que la etapa de pre-procesamiento no sea muy compleja y no tome un tiempo mayor del planificado. Debido a que el objetivo de este trabajo no se centra en el pre-procesamiento realizado, se están simplificando las variables que podrían aparecer a raíz del ruido, las cuales incluyen funciones más elaboradas de limpieza, filtro y segmentación de la señal auditiva. Se han definido dos tipos de ambientes: dentro de una oficina cerrada y los exteriores del pabellón de Ingeniería Electrónica, Informática y de Telecomunicaciones de la Pontificia Universidad Católica del Perú. Para este último caso, se ha definido tomar las muestras en un horario que tenga la menor cantidad de actividad sonora posible.

En segundo lugar, debemos mencionar que las señales que serán analizadas a modo de prueba serán capturadas en ambientes de espacios abiertos, similar al ambiente propuesto en la etapa de entrenamiento debido a la dificultad de poder probar el algoritmo con audios pertenecientes a un área natural.

En tercer lugar, respecto a las pisadas, se ha definido que se reconocerán las pisadas de una persona por vez y los pesos de las personas que transiten deben ser similares entre sí, con una variación de 10 kilos como máximo. Además, todas deberán ir al mismo ritmo de caminata, siendo de una rapidez entre lenta y moderada, considerándose en un rango de 30 m/min a 40 m/min. Respecto a las voces, se ha delimitado que también sea la voz de una persona adulta femenina o masculina por vez, debiendo tener un volumen alto o por lo menos audible respecto a la distancia donde se encuentra el micrófono.

1.3.3. Riesgos

Los riesgos identificados que podrían afectar la continuidad del proyecto se describen a continuación, junto a sus respectivas medidas de mitigación.

Riesgo Identificado	Impacto	Medidas correctivas
Retraso en la obtención de muestras a utilizar	Alto	Buscar fuentes en internet con señales de audio similares que puedan ser utilizadas en el proyecto
Alta presencia de ruido en las señales de audio.	Medio	Establecer una cantidad de muestras a obtener mayor de la necesaria para poder contar con una cifra considerable en caso algunas muestras deban ser eliminadas.
Dificultad en la integración de las funciones de Matlab y otro lenguaje de programación	Medio	Buscar la forma de guardar los resultados de Matlab de modo que puedan ser usados por otro programa, por ejemplo, mediante archivos XML.

Tabla 1.2 Riesgos identificados y sus medidas correctivas

1.4. Justificación y Viabilidad

1.4.1. Justificación

El proyecto presentado tiene como finalidad implementar un algoritmo que, en base a las señales acústicas captadas mediante un dispositivo de audio, sea capaz de procesarlas y clasificarlas adecuadamente de acuerdo a la presencia de sonidos de pisadas o voces. Esto está orientado a que en un futuro pueda desarrollarse un sistema de vigilancia de áreas en espacios abiertos para poder identificar a personas no autorizadas.

Uno de los principales beneficios de realizar este trabajo está en encontrar una herramienta alternativa para el futuro desarrollo de sistemas de vigilancia que necesite menor capacidad de procesamiento y sea más económica en cuanto a infraestructura (dispositivos de captura) y uso de recursos tecnológicos. Esto tiene como principal fin que sea un sistema más accesible a personas, empresas o países, manteniendo el margen de eficiencia y efectividad en cuanto a resultados de modo que pueda ser confiable y se pueda automatizar la tarea o por lo menos reducir la cantidad de personal necesaria para realizarla. Otro de los beneficios del presente trabajo reside en el aporte de nuevas ideas y resultados al área de reconocimiento de patrones en sonidos. Esta es un área que aún está siendo desarrollada actualmente debido a la gran concentración en el uso de medios visuales, por lo que aportaría material para un futuro estudio más profundo y a detalle.

Respecto a otras aplicaciones que podrían darse en el mundo real, podemos mencionar en primer lugar, que se podría usar en cualquier tipo de sistema de vigilancia orientado a detectar personas, realizando los correspondientes ajustes en cuanto al entorno donde se quiere aplicar. Esto podría tener uso en áreas como la seguridad de empresas, centros comerciales, hogares, seguridad militar o incluso espionaje. Y en segundo lugar, podría usarse como un mecanismo de control y obtener estadísticas de él, por ejemplo, para conocer cuántas personas entran o salen de un lugar.

1.4.2. Viabilidad

En esta sección se presentan y explican los distintos ámbitos de la viabilidad del proyecto, a fin de determinar si el trabajo propuesto podrá ser desarrollado.

1.4.2.1. Viabilidad Técnica

Respecto a la viabilidad técnica, podemos mencionar que tanto las etapas del proyecto como los algoritmos de soporte a usarse en cada una cuentan con diversas investigaciones científicas que ya han validado su uso. Además, las herramientas elegidas para el desarrollo del proyecto contienen toda la funcionalidad necesaria para la implementación del mismo de acuerdo a la documentación encontrada.

1.4.2.2. Viabilidad Temporal

La viabilidad del proyecto se definió en dos etapas correspondientes a dos semestres académicos, y según esto se delimitó el tema así como el alcance del proyecto de modo que pudiera adecuarse al rango de tiempo definido. Según esto, para la primera etapa el enfoque está en el planteamiento del problema, los objetivos y los elementos necesarios para el desarrollo del proyecto. Para la segunda etapa, se realizará el análisis de los datos, la implementación del algoritmo y evaluar y documentar los resultados obtenidos. Para el pre-procesamiento de los datos, se considera que ocupará un 25% del tiempo total, para el análisis y clasificación se estima el 50% debido a que estos son los módulos principales del tema planteado. Para el resto de actividades, se asigna el 25% de tiempo restante. También se ha considerado tener una etapa intermedia durante la cual se armará la base de datos para las futuras pruebas y entrenamiento del algoritmo.

1.4.2.3. Viabilidad Económica

Sobre la viabilidad económica, podemos mencionar que no habría mayores problemas pues los programas elegidos a usar como herramientas, tienen licencias que son provistas a los alumnos de la Pontificia Universidad Católica del Perú.

2. Marco conceptual

2.1. Conceptualización

2.1.1. Transductor

Un transductor es un dispositivo que recibe la potencia de un sistema mecánico o acústico y la transmite a otro, en forma distinta. Es decir, tiene una entrada de información o algún tipo de señal (que suele ser una manifestación de energía), la transforma internamente y la muestra mediante una salida con un formato determinado [PAPAVASSILIOU 2008].

Uno de los transductores más conocidos y relevantes para este trabajo es el micrófono, el cual es considerado un transductor del tipo electroacústico, ya que convierte las vibraciones sonoras en energía eléctrica.

2.1.2. Sensor

Un sensor es un dispositivo que responde ante estímulos físicos, químicos o biológicos. Detecta estas variaciones en una magnitud física y las convierte, por lo general, en señales eléctricas, útiles para un sistema de medida o control. La principal diferencia entre los sensores y los transductores es que los primeros miden constantemente las variaciones en la señal o energía específica gracias a alguna propiedad que posee el sensor en sí [PAPAVASSILIOU 2008]. Existen diversos tipos de sensores, pero este trabajo se enfocará principalmente en dos de ellos.

2.1.2.1. Sensores acústicos de onda

Estos son dispositivos electrónicos capaces de medir niveles de sonido. Su mecanismo de detección es mediante una onda mecánica y generalmente el material usado para generar las ondas acústicas es piezoeléctrico. Esto quiere decir que la señal eléctrica generada se obtiene mediante cambios en la presión en el material piezoeléctrico.

2.1.2.2. Sensores sísmicos

Son dispositivos usados para medir vibraciones sísmicas mediante la conversión del movimiento del suelo en una señal electrónica medible. La señal obtenida naturalmente es analógica, por eso los sensores sísmicos deben estar relacionados con una unidad de adquisición de data para poder convertir la señal en una salida con formato digital que pueda ser leída y procesada por computadoras. Un sensor sísmico frecuentemente es el geófono.

2.1.3. Ruido

Este es un término usado para hacer referencia a cualquier señal no deseada que interfiere con la medida y procesamiento de la señal deseada. Sin embargo, esta es una señal bastante amplia pues existen distintos tipos de ruido, para señales de comunicación, señales de voz, señales de video, en

imágenes, etc. Específicamente, para las señales de audio, se puede dividir el ruido en cuatro categorías: ruido aditivo (proveniente de fuentes de sonido en el ambiente), interferencia, reverberación (causada por propagación en diferentes caminos), y eco. La contaminación en la señal causada por el ruido puede cambiar drásticamente las características de la señal de audio y degradar su calidad. [BENETSY 2009]

2.1.4. Reducción de ruido

Por el hecho de vivir en un ambiente natural donde el ruido es inevitable, las señales de voz y audio son generadas inmersas en un ambiente acústico lleno de ruido. Por esto, es esencial para el procesamiento de voz y sistemas de comunicación aplicar efectivamente técnicas para la reducción de ruido de modo que se pueda extraer la señal deseada. Estas técnicas tienen un rango amplio de aplicaciones, desde ayudas para escuchar por teléfonos celulares, sistemas controlados por voz, y sistemas de reconocimiento de voz automáticos. La decisión de usar o no usar una técnica de reducción puede tener un impacto significativo en el funcionamiento de los mismos. Este es un problema bastante complejo y retador debido a diferentes razones. Primero, la naturaleza y características del ruido cambian significativamente de acuerdo a la aplicación, y además, varían con el tiempo. Y en segundo, el objetivo de reducción del ruido es extremadamente dependiente del contexto específico y la aplicación. [CHEN 2006]

2.1.5. Características biométricas

Las características biométricas hacen referencia a características físicas o del comportamiento que son únicas para cada ser humano. Por lo mismo, estas pueden fácilmente asociarse con el reconocimiento de la identidad de una persona. Ejemplos de ellas son las huellas dactilares, el iris, patrones faciales, la firma y los pasos o forma de caminar [JAIN 2004].

2.1.6. Reconocimiento de patrones

Para definir lo que es el reconocimiento de patrones, primero debemos definir un patrón. En simples términos, un patrón es la descripción de un objeto. Dependiendo de la naturaleza del patrón, el reconocimiento puede estar dividido en dos tipos: reconocimiento de objetos concretos y reconocimiento de objetos abstractos. La palabra reconocimiento hace referencia al proceso de conocer una entidad o ganar conocimiento sobre ella. De acuerdo al trabajo presentado, podemos definir el reconocimiento de patrones como un proceso en el que se reconoce un patrón específico usando una computadora mediante algoritmos que extraigan y analicen las características del objeto estudiado. [KHODASKAR y LADHAKE 2014]

2.2. Estado del Arte

2.2.1. Método usado en la revisión del estado del arte

La revisión sistemática fue el método utilizado en la revisión del estado del arte. Durante este proceso se utilizaron las siguientes bases de datos de publicaciones académicas.

- ResearchGate
- IEEE Explore
- Scholar Google
- Science Direct

Formulación de la pregunta

La pregunta de investigación formulada fue: ¿Cuáles son los tipos de sistemas actuales de vigilancia? Los términos usados para resolver esta pregunta fueron: “surveillance”, “system”, “audio”, “pattern”, “footstep”, “speech”, “recognition”,”detection”.

2.2.2. Selección de fuentes

Resultados

Cadena	Research Gate	IEEE Explore	Scholar Google
Surveillance system	10000	652	5590
Audio pattern recognition	100	788	107000
Footstep detection	50	55	16200
Speech recognition	10000	14422	856000

Tabla 2.1 Cantidad de trabajos encontrados por cadena de búsqueda

Cabe aclarar que para la búsqueda de los artículos relevantes se tuvo que identificar un rango de fechas de publicación, que se estableció entre el 2007 hasta el año 2017, y aparte la definición del área. Para ello, se buscó artículos que estuvieran clasificados dentro de cualquiera de las siguientes áreas: computer science, algorithms, artificial intelligence, engineering, acoustics, patterns, feature extraction, video and signal processing.

2.2.3. Investigaciones destacadas en el tema

A continuación se presentarán las investigaciones más destacadas. Esta sección estará dividida en tres áreas: Sistemas de vigilancia basados en audio y video, sistemas de vigilancia basados en audio, y reconocimiento de patrones de audio. Se ha seguido este esquema para poder tener una visión global y una específica sobre la situación actual del tratamiento de señales acústicas con fines de vigilancia o seguridad. Los trabajos fueron seleccionados de acuerdo a la similitud del título y contenido con cada área propuesta, y también de acuerdo a los resultados obtenidos en cada investigación.

2.2.3.1. Sistemas de vigilancia basados en audio y sonido

- **CASSANDRA: fusión de sensores de audio y video para la detección de comportamiento humano agresivo**
  Este trabajo presentó un sistema de vigilancia inteligente de nombre CASSANDRA, que estaba orientado a detectar comportamiento humano agresivo en ambientes públicos. Un aspecto significativo de este sistema, fue el uso complementario de señales de audio y video para evitar ambigüedades en la clasificación de las escenas reales captadas.
  En el primer nivel, se extraen las señales de audio y video de los sensores, las cuales son procesadas para producir un nivel intermedio

de datos donde se resumen las características de cada señal. Luego, estas entran al proceso de clasificación, que consiste en una red Bayesiana dinámica que combina audio y video e incorpora cualquier conocimiento del contexto específico para producir un indicador global de agresión.

Para la unidad de audio, se trabajó en el dominio de tiempo-frecuencia con un enfoque común en el análisis auditivo de entornos. La transformación del dominio de tiempo-señal al de tiempo-frecuencia se realizó mediante un modelo de transmisión lineal basado en el oído humano. Los resultados se obtuvieron en forma de un espectro de energía llamado cocleograma.

Para la unidad de video, se trabajaron con elipses y puntos para detectar y representar las figuras humanas y las partes más importantes de ella. Luego está la unidad de fusión, la cual recibe los datos procesados de las unidades previas y los clasifica de acuerdo a una red Bayesiana. Los resultados obtenidos fueron de aproximadamente 80% de precisión. [ZAJDEL Y OTROS 2007]

- **Reconocimiento audiovisual de eventos en secuencias de video de vigilancia**

  Este trabajo presentó un nuevo método capaz de integrar información proveniente de sensores de audio y video para el análisis del entorno en un escenario común de vigilancia, utilizando solo una cámara y un micrófono monoaural.

  La información visual es analizada por un módulo de modelación estándar sobre el fondo y el primer plano visual, reforzado con una etapa de detección de audio bajo el mismo esquema de modelo. Este proceso permite detectar patrones de audio y video por separado representando eventos inusuales en el entorno analizado. La integración de ambas señales se realiza mediante la sincronía de los eventos, y su asociación se basa en el cómputo de un rasgo característico llamado la matriz de concurrencia de audio-video, permitiendo segmentar los eventos audiovisuales y discriminarlos.

  Para las señales visuales, se utilizó un modelo Gaussiano para identificar el primer plano en una escena, y un histograma de colores para detectar el inicio y fin de eventos. Para las señales auditivas, primero se usó un análisis de frecuencia multibanda para caracterizar la señal y extraer rasgos característicos. Luego se utiliza una mezcla adaptativa de Gaussianos para modelar las características relacionadas a cada banda de frecuencia. Para la experimentación del trabajo, se usaron algoritmos clasificadores y de agrupamiento, como el de los vecinos más cercanos (KNN). [CRISTANI Y OTROS 2007]

2.2.3.2. Sistemas de vigilancia basados en audio

- **Detección y localización de gritos y disparos para sistemas de vigilancia basados en audio**

  El objetivo de este trabajo fue identificar sonidos de gritos y disparos en un área cuadrada de espacio público y localizar su ubicación de modo

que una cámara de video pueda ser apuntada en tal dirección posteriormente. El sistema emplea dos modelos de mezcla Gaussiana para la discriminación de los sonidos. Cada clasificador fue entrenado usando diferentes características de los sonidos, las cuales fueron escogidas de una colección de rasgos de audio que son de diferentes tipos: características temporales, de energía, espectrales, y perceptuales. Además, el trabajo introduce nuevos tipos de características: distribución de espectro, periodicidad, y características basadas en la función de auto-correlación. Para extraerlas se usaron filtros y funciones envolventes de la señal. [VALENZISE 2007]

- **Sistema de vigilancia basado en señales acústicas para la detección de intrusos**

  Este proyecto tuvo el propósito de identificar una persona no deseada en un área específica cerrada mediante las señales recibidas por medio de un micrófono de red distribuida. El sistema proponía interpretar las señales de audio para discriminar los sonidos provenientes del exterior, y los que se generaban dentro de la habitación, los cuales serían considerados falsas alarmas. El sistema se componía de los siguientes bloques esenciales: el localizador de la fuente de sonido, detector de evento acústico, el cual detecta una variación en los sonidos utilizando energía y una función de variación espectral, y el módulo de generación de alarma, el cual recibe información de los otros módulos para tomar la decisión de lanzar o no una alarma.

  Cabe recalcar que el sistema no realiza una clasificación de los sonidos que recibe, solo los discrimina como provenientes del exterior o interior. Los resultados fueron bastante favorables, teniendo un ratio de cero respecto a fallos, y de 0.01 de sonidos del exterior y 0.03 de sonidos del interior respecto a falsas alarmas. [ZIEGER 2009]

- **Detección acústica ligera de explotación forestal en redes de sensores inalámbricos**

  Esta investigación estuvo orientada a detectar el sonido de sierras eléctricas en un área forestal, de modo que se pueda prevenir de la tala indiscriminada de árboles. Se utilizaron sensores acústicos, los cuales enviaban señales que después de ser limpiadas de ruido, pasaban por un algoritmo que extraía sus características más relevantes, para lo cual usaron una función de autocorrelación entre las señales que recibían. Durante el trabajo, se explica que a pesar de que la Transformada de Fourier hubiera sido más adecuada para la extracción de características, este método implicaba una mayor capacidad de procesamiento lo cual, dada la arquitectura planteada, no podía ser implementado.

  Para la clasificación, se utilizaron distintos algoritmos. Primero se compararon algoritmos del tipo árbol de decisión, siendo el mejor de ellos el árbol de decisión alternativo. Posteriormente, este algoritmo se comparó contra los resultados del algoritmo de máquinas de vectores

de soporte. Se obtuvo una precisión de entre 77% y 78% de los dos algoritmos respectivamente [CZUNI 2014].

2.2.3.3. Reconocimiento de patrones en audio

- **Método acústico de huellas para la detección de patrones en señales de audio**

  El trabajo se enfocó en el desarrollo de un algoritmo de reconocimiento de señales de audio con una complejidad computacional limitada. Debido a que el algoritmo iba a usarse en entornos al aire libre donde los ruidos del fondo son altos, se propuso una optimización para los sonidos mecánicos como lo son las bocinas de los autos, el timbrado de teléfonos, etc. El proyecto se divide en dos fases: detección y reconocimiento.

  Para la fase de detección, se basaron en la relación señal a ruido: cada vez que aparecía un pico, se consideraba que se había detectado la presencia de un evento. Para la fase de reconocimiento, se debía obtener el tipo de sonido que había sido detectado. Para esto, se utilizaron la frecuencia, utilizando el método de auto-correlación, y el envolvente espectral. Luego, era comparado con audios de las señales de audio comunes, utilizando la fórmula del promedio lineal de error. La transición entre ambas fases era llevada a cabo mediante un modelo de máquina de estados finitos. [ALIAS 2014]

- **Reconocimiento de eventos acústicos usando redes neuronales profundas**

  Este trabajo propuso el uso de una red neuronal profunda para el reconocimiento de eventos acústicos aislados, como los pasos, el llanto de un bebé, una motocicleta, la lluvia, etc. El trabajo propone un algoritmo generalizado, que puede usarse para distintas situaciones, teniendo que hacer ciertos ajustes para poder personalizarlo y obtener una mayor eficiencia. Las redes neuronales profundas han sido utilizadas previamente para el reconocimiento automático de voz, obteniendo resultados mejores a lo esperado. Por lo mismo, se propone utilizarlas para el reconocimiento de otro tipo de eventos, debido a la similitud en los problemas.

  Para el pre-procesamiento de las señales, se utilizó una función de segmentación de Hamming para normalizar la señal en amplitud y dividirla en marcos de 50 ms. Para la extracción de características, se utilizaron los coeficientes cepstrales de la frecuencia de Mel, obteniendo 40 coeficientes por cada marco. Se probó el método propuesto con una base de datos de 1325 archivos pertenecientes a 61 clases de eventos distintos. Los resultados obtenidos mostraron una precisión de más del 60%, siendo mejor que el método común utilizado de modelo de mezcla Gaussiana basado en modelos escondidos de Markov. [GENCOGLU Y OTROS 2014]

- **Reconocimiento automático de eventos de audio usando patrones dinámicos locales binarios**

Este trabajo se basa en el uso de patrones locales binarios, el cual es muy usado para representar características de imágenes, aplicándolo al reconocimiento de eventos de audio. Para reproducir un reconocimiento efectivo, se propone utilizar un nuevo modelo dinámico de patrones locales binarios, basado en un análisis de espectrograma y el sistema auditivo humano. Para esto se proponen cinco etapas principales.

Primero, la señal de audio es convertida a un espectrograma por la Transformada de corto plazo de Fourier. Segundo, se aplican filtros de imágenes para reforzar el espectrograma. Tercero, este espectrograma se divide en distintas sub-bandas de frecuencias. Cuarto, la característica del patrón dinámico local binario es extraída de cada sub-banda. Y quinto, estas características son concatenadas en un vector que sirve como dato de entrada para el proceso de clasificación, llevado a cabo por una máquina de vectores de soporte. [WANG 2015]

2.2.3.4. Sistemas de reconocimiento de pisadas

- **Análisis de cadencia de patrones temporales de andadura para la discriminación sísmica entre pisadas humanas y cuadrúpedas**

  El objetivo principal de esta investigación fue ver cuán estadísticamente diferentes son los patrones temporales de andaduras entre humanos y cuadrúpedos, y verificar la posibilidad de que el análisis de cadencias de patrones de andaduras sea usado como una característica principal para el reconocimiento de pisadas. Previamente, en otros trabajos se propone utilizar el análisis de cadencias basándose en la frecuencia fundamental de la andadura, sin embargo, si un cuadrúpedo camina de forma lenta podría generar la misma frecuencia de andadura que un humano. Es por eso que el trabajo propone el uso de patrones temporales de la andadura.

  Para la clasificación, utilizan un modelo de mezcla Gaussiano debido a las variables multimodales que se plantean a partir del preprocesamiento y extracción de características. Finalmente, los resultados demuestran que se obtuvo más del 95% de reconocimiento acertado [PARK 2009].

- **Marco para la detección de pisadas en un entorno cerrado**

  En este trabajo se presenta un marco para un sistema técnico y un reporte de análisis experimental para la detección de sonidos de pisadas en ambientes internos o cerrados usando técnicas de detección de inicio de pisadas, reconocimiento de patrones de una sola pisada, y cálculo del ritmo en una base de datos etiquetada, que contiene datos de sonidos de pisadas por una duración de 5 horas.

  Como pre-procesamiento, cada señal debe simular la sensación del sonido de la percepción auditiva humana, y se debe suavizar el impacto del evento en términos de energía. Luego, como el impacto de una pisada puede incluir distintos sub-impacto, se debe agrupar todos estos sub-impacto en uno solo. Este proceso se define como la detección del último punto de la pisada. Las características extraídas por cada señal se basaron en la energía espectral, la cual tenía ciertas variaciones de

acuerdo al tipo de piso. Luego, se obtienen características del ritmo general de la caminata humana, entre las que se encuentran el intervalo entre cada impacto de pisada, la duración de esta, el poder del impacto, etc. Y se utilizó un modelo de probabilidad Gaussiana para su clasificación. [SHE 2004]

- **Análisis rítmico de las señales ortogonales de la caminata humana**
  Esta investigación incluye la extracción de diferentes características de la caminata, es decir se enfoca no solo en pisadas, sino también en movimientos de brazos y piernas por ejemplo, captados a través de distintos tipos de sensores como sísmicos, ultrasónicos y electromagnéticos. Lo interesante de esta investigación es que trabajó con dos tipos de data: una data captada dentro del pasillo de un edificio, y otra captada en un ambiente al aire libre perteneciente al sendero de un bosque. Esto nos permite identificar qué características son similares independientemente del ambiente en donde fueron captadas, y para qué características sí se debe hacer un análisis dedicado dependiendo del tipo de suelo y entorno en el que se encuentra [EKIMOV 2010].

2.2.3.5. Sistemas de reconocimiento de voz

- **Característica de modulación de duración media para un reconocimiento robusto de voz**
  Diversos estudios han demostrado que los sistemas de voz automáticos se deterioran en resultados con niveles elevados de ruido y degradaciones en el canal de la señal, comparado a la capacidad humana de reconocimiento de voz. Por esto, en este trabajo se presenta una característica basada en modulación de duración media de la amplitud de voz, que es una característica compuesta al capturar las modulaciones de sub-banda y una modulación resumida del total de la señal.
  Esta característica es similar a la modulación normalizada de coeficientes cepstrales, o los coeficientes de energía promedio de Hilbert, pero la diferencia es que tiene una capa extra de modulación resumida y una un nuevo enfoque para estimar las modulaciones en amplitud de una sub-band limitada por señales de voz. Se busca que esta característica sea resistente al ruido y al canal para obtener una mayor eficiencia en el reconocimiento de voz. [MITRA Y OTROS 2014]

- **Siri**
  Este es un programa de computadora desarrollado por la compañía Apple que trabaja como un asistente personal y navegador interactivo con el usuario manejando el dispositivo en el cual este programa es albergado. Este recibe comandos por medio de la voz, los procesa y responde a ellos mediante acciones o respuestas habladas. [APPLE 2014]

2.2.4. Conclusiones

Podemos concluir, en primer lugar, que existen varios proyectos que involucran las señales de audio para la detección de algún tipo de evento. Sin embargo, estos proyectos están enfocados en temas como vigilancia urbana, doméstica o empresarial. También se ha podido observar que existen modelos básicos para el reconocimiento de patrones por audio, que sin embargo se podrían adaptar a una situación más específica para obtener un mayor porcentaje de precisión en su clasificación.

En segundo lugar, después de la revisión de los artículos e investigaciones científicas relacionadas al tema de reconocimiento de pisadas, podemos concluir que la mayoría de ellos tienen como complemento a las señales acústicas, las señales sísmicas para la detección y análisis de pisadas, debido a que este tipo de señales ofrece un mejor análisis y por ende una clasificación de mayor precisión. También vemos que hay distintos tipos de pre-procesamiento de información, entre los cuales el principal es el de la Transformada de Fourier y las variantes que pueda tener. Además, los clasificadores usados son del tipo supervisado y variando en sus resultados de acuerdo al tipo de característica extraída.

En tercer lugar, se observa que el reconocimiento de voz es un tema bastante avanzado a diferencia de las pisadas, pues ya es aplicado a software de uso comercial e incluso ha llegado a estar a un nivel en el que se puede descifrar lo que la persona está diciendo, incluso en diferentes idiomas.

3. Adquisición y pre-procesamiento de las señales de audio

3.1. Generación de audios propios

Inicialmente se cuenta con una base de datos que tiene archivos de audios que fueron grabados en ambientes controlados para realizar la etapa de entrenamiento y clasificación. Estos audios se caracterizan por tener una cantidad mínima de ruido, además de ser homogéneos en la calidad de la señal y resaltar las características necesarias. Por ejemplo, para los audios de pisadas no se necesita elevar mucho el volumen para escuchar el sonido que producen estas. Sin embargo, se ha generado un pequeño conjunto de audios que provienen de entornos no controlados y ruidosos de modo que se pueda ver la diferencia en la precisión del algoritmo con este tipo de audios. Para esto, se han grabado audios de duraciones variables con un teléfono celular. En ellos se intercalan sonidos tanto de pisadas como de voces. Dos de estos audios se encuentran en espacios cerrados, mientras que los otros dos fueron grabados en espacios abiertos. Las características generales de estos audios son las mismas que los audios del conjunto de entrenamiento, el cual es detallado a continuación.

3.2. Base de datos de audio

3.2.1. Descripción de data propia

Para realizar la comprobación de la eficacia del modelo planteado, se ha generado un conjunto de archivos de audios propios, los cuales constan tanto de voz como de pisadas humanas grabadas en ambientes poco controlados. Estos han sido guardados en formato wav y la duración se ha recortado para que sea de un segundo, al igual que los archivos utilizados para la etapa de entrenamiento del modelo.

3.2.2. Descripción de data de repositorios

Para conformar la base de datos, se han tomado archivos de distintas fuentes. En el caso de audios de voces, estos fueron tomados de una base de datos de libre uso “Census Database AN4” del grupo de reconocimiento de voz de la Universidad Carnegie Mellon. Esta base de datos original contiene, para el directorio de entrenamiento, un total de 74 subdirectorios, uno por cada sujeto hablante. 21 subdirectorios pertenecen a mujeres y 53 a hombres. Para el directorio de validación, se encuentran 10 subdirectorios, 3 de mujeres y 7 de hombres. Los audios originalmente se encuentran en formato .raw o .sph por lo que tuvieron que transformarse a formato .wav utilizando una frecuencia de muestreo de 16 kHz. Además, los audios tenían duraciones de 4 a 6 segundos ya que eran deletreos alfanuméricos al azar. Estos audios fueron recortados de modo que tuvieran la duración de un segundo o menos y contuvieran la voz de una persona deletreando una letra, número o palabra corta. Después de las modificaciones especificadas a los audios originales, se eligieron al azar 95 audios para la etapa de entrenamiento, pertenecientes a 7 hombres y 5 mujeres, y 58 audios para la validación, pertenecientes a 3 hombres y 2 mujeres.

En el caso de los audios de pisadas para la etapa de entrenamiento, estos fueron extraídos de los videojuego Counter Strike, el cual es del tipo de jugador en primera persona, y asegurándonos que se encontrara en la más óptima calidad. Se han adquirido audios de pisadas hasta en 10 terrenos diferentes, teniendo 6 pisadas por cada tipo de terreno. Estos audios fueron encontrados en formato wav por lo que no tuvieron que pasar por alguna transformación previa. Para la etapa de validación, se utilizó un juego de datos perteneciente a otro videojuego, Minecraft, el cual se encontraba en formato OGG. En este caso, se tenía audio por dos canales distintos por lo que se encontraba en modo estéreo. Tuvo que hacerse una transformación a formato wav y reducir el sonido a un solo canal para que cumpliera con el tipo de sonido mono.

Finalmente, se cuentan con 95 audios de voces y 90 audios de pisadas sobre distintos terrenos para la etapa de entrenamiento. Para la etapa de validación, se obtuvieron 58 audios de voces y 60 audios de pisadas.

Características generales:

- Duración no mayor a 1 segundo
- Formato wav
- Tipo de sonido: Mono (un solo canal)
- Frecuencia de muestreo en voces: 16000 Hz
- Frecuencia de muestreo en pisadas: 44100 Hz

3.3. Generación de concatenados de audios

Para la experimentación final de este trabajo donde se incluye en módulo de pre-procesamiento, se han generado 16 audios de duración variable mayor a 1 segundo, los cuales contienen en un orden aleatorio sonidos de voces y pisadas. Estos audios fueron generados a partir del programa Audacity y los sonidos son los mismos utilizados en la etapa de clasificación descrito en el apartado anterior. De los 16 audios generados, 11 audios contienen 4 sonidos diferentes intercalados, y 5 audios contienen 5 sonidos intercalados. Esto resultaría en un total de 69 sonidos que se buscan identificar y clasificar correctamente. A continuación se presenta el esquema de los audios utilizados.

Nombre	Descripción de sonidos	Duración (seg)
Audio 1	Pisada, pisada, pisada, pisada	1.6
Audio 2	Pisada, pisada, pisada, voz	2.6
Audio 3	Pisada, pisada, voz, pisada	3.1
Audio 4	Pisada, pisada, voz, voz	1.7
Audio 5	Pisada, voz, pisada, pisada	3.2
Audio 6	Pisada, voz, pisada, voz	1.4
Audio 7	Pisada, voz, voz, pisada	1.8
Audio 8	Pisada, voz, voz, voz	2.1
Audio 9	Voz, pisada, pisada, pisada	1.6
Audio 10	Voz, pisada, pisada, voz	2.2
Audio 11	Voz, pisada, voz, pisada	2.7
Audio 12	Voz, pisada, voz, voz, pisada	3.2
Audio 13	Voz, voz, pisada, pisada, voz	3.0

Audio 14	Voz, pisada, pisada, voz pisada	2.3
Audio 15	Voz, voz, voz, pisada , pisada	2.1
Audio 16	Voz, voz, voz, voz, pisada	2.7

Tabla 3.1 Descripción de audios con sonidos concatenados

Posteriormente, también cabe resaltar que estos audios tienen las mismas características generales del conjunto de datos anterior, con la única diferencia que en todos estos audios se trabajó con una frecuencia de muestreo de 44100 Hz, por lo que los audios de voces fueron transformados a esta frecuencia más elevada.

3.4. Pre-procesamiento de la señal

Como se mencionó anteriormente, existe un conjunto de datos los cuales consisten en una mezcla de sonidos a identificar. Para estos audios se consideró tener un módulo de pre-procesamiento el cual se encargue de identificar las regiones de interés en la señal, recortarlos y obtener como resultado distintas señales a partir de una sola que representen los sonidos encontrados en ella. Para lograrlo se utilizó una función envolvente de la señal que permitiera identificar los picos a lo largo de la señal en una cantidad determinada de muestras.

Inicialmente se tuvieron tres tipos de envolventes basados en una característica de la señal para calcularlo. Estas eran: envolventes analíticas, que utilizan filtros de Hilbert; envolventes basadas en la raíz cuadrada promedio para lo cual se utiliza el método de *windowing* (se le llamará de aquí en adelante RMS por sus siglas en inglés *root-mean-square*); y envolventes basadas en los picos de la frecuencia determinadas por interpolación de spline. Para determinar el mejor tipo a utilizar en la experimentación, se probó con los tres tipos de envolvente con un audio aleatorio del conjunto de audios y con el mismo parámetro de cantidad de muestreo en 1000. Debido a que los sonidos tienen una frecuencia de muestreo de 44100 y lo audios no tienen una duración mayor a los 3.5 segundos, se consideró adecuado que el muestreo se tome cada 1000 muestras para tomar la señal envolvente. Se obtuvieron los resultados mostrados en las Figura 3.1., Figura 3.2. y Figura 3.3.

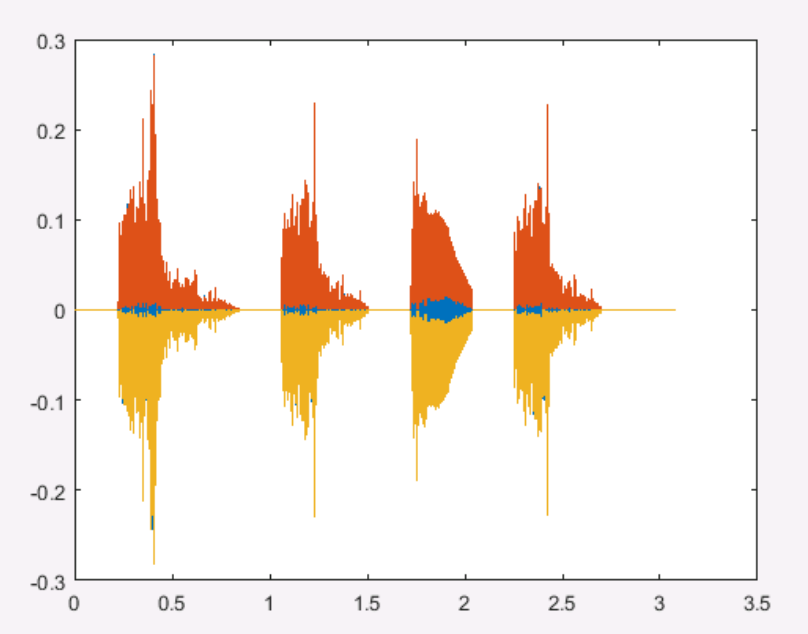


Figura 3.1 Señal con envolvente analítica con un muestreo tomado cada 1000 muestras. En azul se observa la señal original, en naranja la señal que representa los altos, y en amarillo la señal que representa los bajos según el resultado de la envolvente analítica.

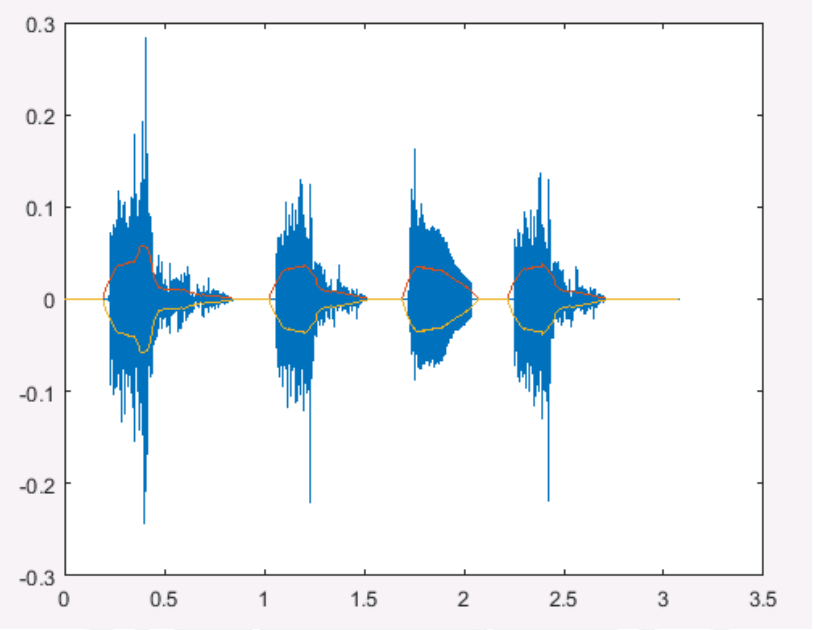


Figura 3.2 Señal con envolvente RMS con un muestreo tomado cada 1000 muestras. En azul se observa la señal original, en naranja la señal que representa los altos, y en amarillo la señal que representa los bajos según el resultado de la envolvente RMS.

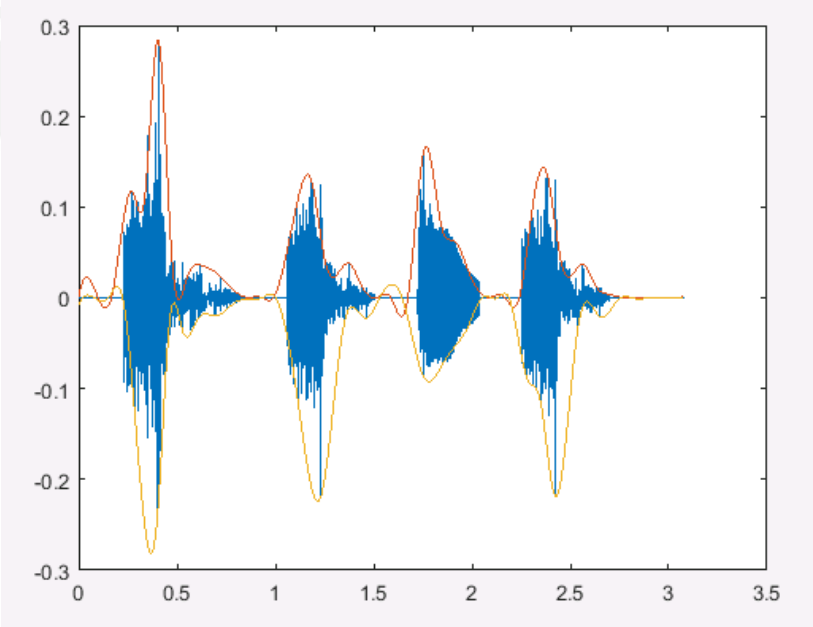


Figura 3.3 Señal con envolvente de picos con un muestreo tomado cada 1000 muestras. En azul se observa la señal original, en naranja la señal que representa los altos, y en amarillo la señal que representa los bajos según el resultado de la envolvente de picos.

De acuerdo a los resultados obtenidos, se observó que tanto la señal envolvente RMS como la señal basada en picos tenían resultados con los cuales se podría trabajar en el módulo de pre-procesamiento. Debido a esto, la experimentación más adelante planteada compara ambas funciones para evaluar los resultados respecto

a todo el conjunto de datos y al resultado final de la clasificación. Además, también se determina que se trabajará con la señal que representa los altos y para determinar si se empieza con un sonido significativo, se tomará un valor como mínimo en el dominio de la frecuencia. Para poder establecer el valor mínimo de frecuencia que marca el inicio de un sonido a clasificar, se analizó las señales ya que en todas se aprecia que las frecuencias mayores a una amplitud de 0.05 son parte de una pisada o una voz. La cantidad de muestras a tener en consideración para la función envolvente como base será de 1000, lo que equivale a una frecuencia de muestreo de 44.1 muestras/seg. Esto representaría una toma de muestra cada 0.02 segundos aproximadamente. Esto será presentado en la sección de experimentación.

4. Caracterización de audios

4.1. Método MFCC

4.1.1. Introducción

Para extraer un vector de características más significativas de los audios, uno de los métodos que se ha decidido utilizar es el de coeficientes cepstrales de frecuencia de Mel (Mel Frequency Cepstral Coefficients) que de aquí en adelante abreviaremos como MFCC. Para dar una breve explicación del procedimiento de esta función, se utilizará como base el capítulo 9.3 del libro de procesamiento de voz y lenguaje de Jurafsky y Martin (2009).

A continuación, se presenta el esquema de lo que realiza el método.

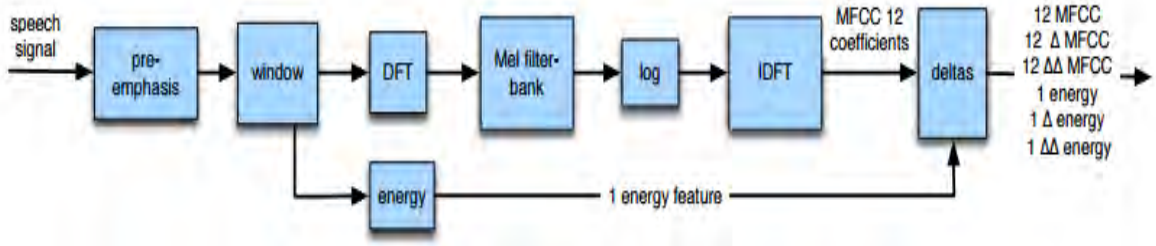
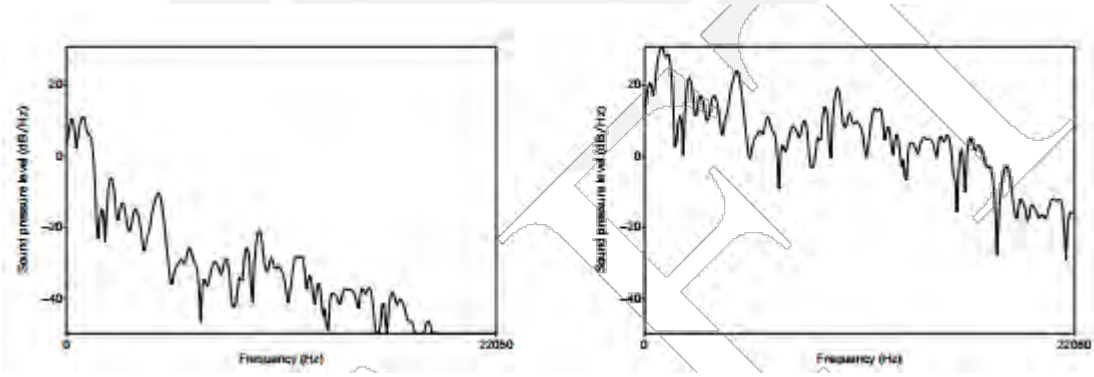


Figura 4.1 Esquema del funcionamiento del método MFCC

4.1.2. Pre-énfasis

El pre-énfasis es la primera etapa en la extracción de características con MFCC. La finalidad de esta etapa es aumentar la cantidad de energía a las frecuencias más altas. Esto se realiza debido a que el espectro para los segmentos de voz tiene más energía en bajas frecuencias que en altas frecuencias. Esto se lleva a cabo mediante el uso de un filtro del tipo filtro pasa-alto.



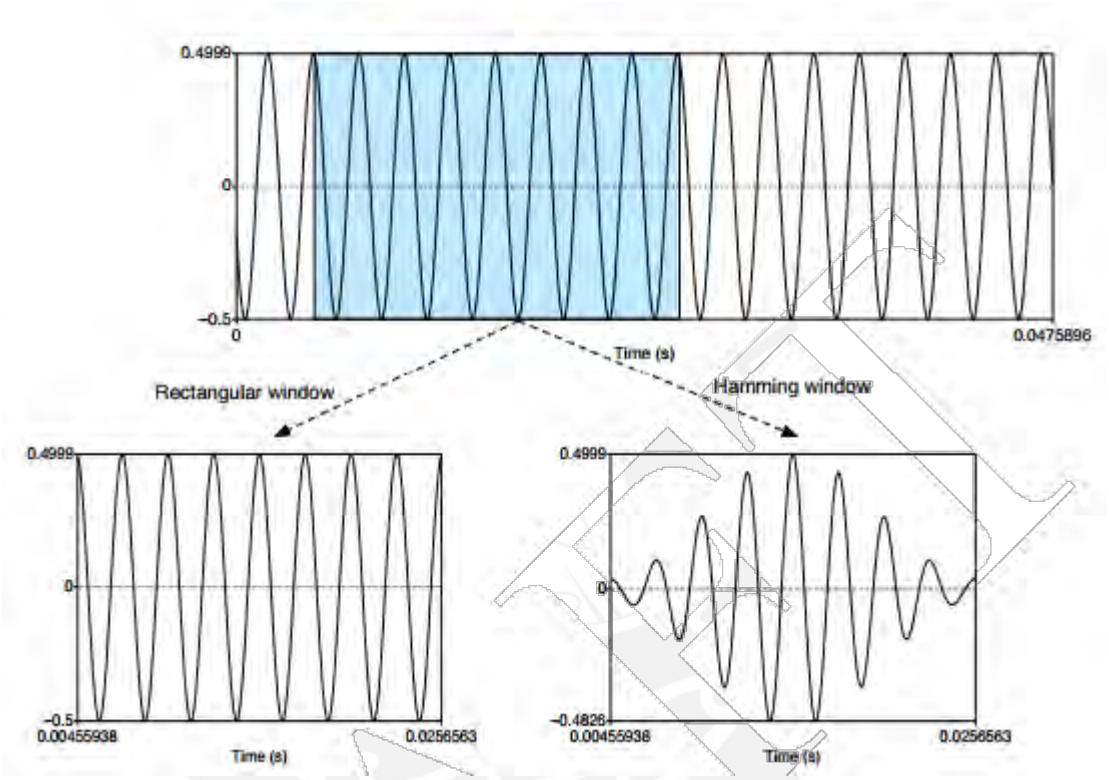
a) Señal antes del pre-énfasis b) Señal después del pre-énfasis

Figura 4.2 Comparación de una señal de audio en el dominio de la frecuencia a) antes del pre-énfasis y b) después del pre-énfasis

4.1.3. Windowing

Debido a que se trabaja con señales cuyo espectro cambia con mucha rapidez, se necesita de un proceso que pueda extraer pequeñas partes de la señal las cuales puedan asumirse de poca variación espectral. Por eso se tiene la etapa del windowing. Para esto, se debe definir primero el acho de cada ventana, la

diferencia entre cada una y el tamaño de la misma. Estas ventanas estarán definidas con cero en ciertas regiones y un no-cero en otras regiones. Luego, atravesarán toda la señal y se obtendrá una nueva onda del resultado de multiplicar el valor de la señal por el valor de la ventana en un tiempo dado. Cada parte de la señal extraída mediante una ventana se denomina marco. Existen distintos tipos de ventana y aunque la más fácil es la ventana rectangular, en el método MFCC se utiliza preferentemente una ventana de Hamming ya que evita las discontinuidades en la señal resultante. En la Figura 4.3 se puede apreciar la diferencia entre la señal obtenida de la ventana de Hamming y la ventana rectangular.



a) Windowing rectangular b) Función de Hamming

Figura 4.3 Resultado de una señal al pasar por la etapa de windowing del tipo a) rectangular y usando b) función de Hamming

4.1.4. Transformada Discreta de Fourier

Dentro de esta etapa se busca extraer la información del espectro para la señal extraída de la etapa anterior. Se necesita saber cuánta energía contiene la señal en diferentes bandas de frecuencia, es decir debemos convertir la señal del dominio del tiempo hacia el dominio de frecuencia. Para lograrlo, se aplica la transformada discreta de Fourier. Uno de los algoritmos comúnmente utilizados para calcularlo, es la transformada rápida de Fourier, la cual se ha utilizado en la implementación del presente trabajo.

4.1.5. Banco de Filtros Mel

La transformada rápida de Fourier extrae información sobre la energía en cada banda de frecuencias. Pero el oído humano es menos sensible a frecuencias altas mayores a 1000 Hertz. Por eso, modelar esta propiedad del oído humano durante la extracción de características mejorará la identificación del modelo de clasificación.

Para lograrlo, las frecuencias extraídas de la etapa anterior deben ser transformadas a la escala de mel. Un mel es una unidad de tono definida de modo que parejas de sonidos que son perceptualmente equidistantes en tono, sean separados por una cantidad igual de mels. La frecuencia de mel puede ser calculada de la siguiente manera:

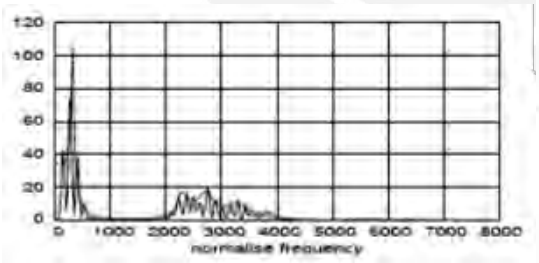
$$mel(f) = 1127 \ln(1 + \frac{f}{700})$$

Como paso final de esta etapa, a partir de esta fórmula, se calcula cada uno de los valores del espectro de mel. Usar el logaritmo hace que los resultados de las características sean menos sensibles a variaciones en la entrada de la señal (por ejemplo, variaciones debido a un cambio en la distancia del micrófono de la fuente de sonido).

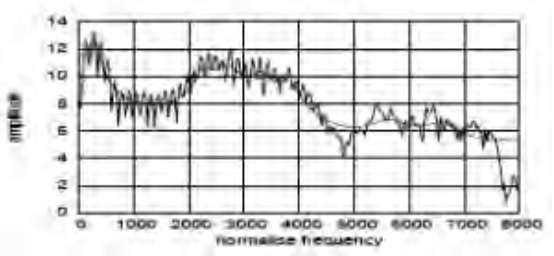
4.1.6. Cálculo del cepstrum

El cepstrum es una manera útil de separar la fuente y el filtro. También se puede decir que es el espectro del logaritmo del espectro. Para explicarlo, se empieza con la parte “el logaritmo del espectro”. Esto quiere decir que el cepstrum empieza con un espectro de magnitudes estándar y se reemplaza cada valor de amplitud en el espectro de magnitudes con su logaritmo. Al resultado de esta operación se la debe considerar como una señal en sí, teniendo una pseudo-señal de audio. Luego se extrae al espectro de esta señal, obteniendo así el cepstrum. Generalmente, solo se toman los primeros 12 valores, que serán los coeficientes que representarán la información de la señal. Una de las propiedades más útiles de estos coeficientes es que tienden a no tener correlación alguna, lo que no ocurre con los coeficientes espectrales.

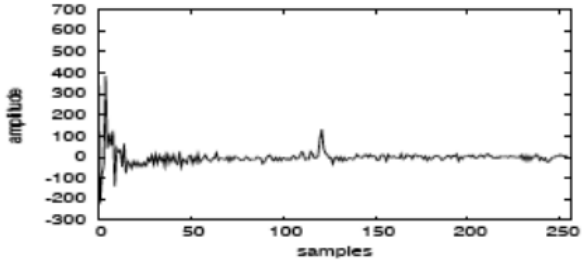
A continuación, se muestra el proceso por el que atraviesa la señal para obtener los coeficientes.



(a)



(b)



(c)

Figura 4.4 Fases de una misma señal durante el cálculo del cepstrum: primero la magnitud del espectro representado en un gráfico de amplitud vs frecuencia normalizada (a), luego el logaritmo de la magnitud del espectro, representado en una gráfico de amplitud (valores logarítmicos de a) vs frecuencia normalizada (b), y por último el cepstrum obtenido, representado en un gráfico de amplitud vs muestras (c)

Respecto a la última figura presentada, se debe aclarar que la unidad de medida del eje X cambia para la figura (c) porque para obtener el cepstrum, la señal pasa del dominio de la frecuencia al dominio del tiempo nuevamente, teniendo así que la unidad de medida de un cepstrum es una muestra (un valor de la señal en un tiempo determinado).

4.1.7. Cálculo de Delta y Energía

En la implementación usada, se utilizan 13 coeficientes como vector de características. De la etapa anterior extraemos 12 coeficientes pertenecientes al cepstrum, con lo cual el coeficiente faltante pertenece a la energía de cada marco. La energía se calcula mediante la sumatoria de la potencia de las muestras en cada marco en el tiempo.

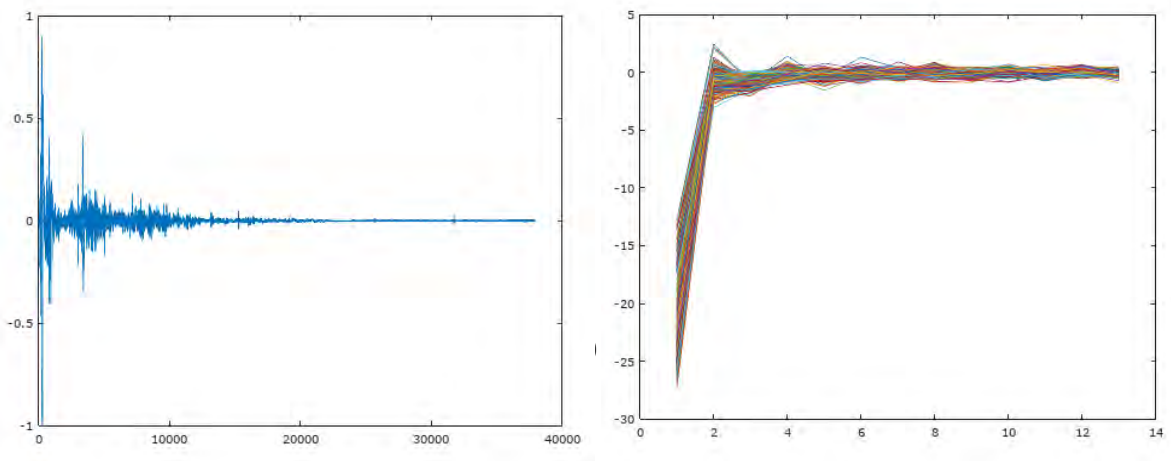
Se debe destacar que, como lo explican los autores Jurafsky y Martin mencionados previamente, debido a que la señal no es constante entre marco y marco, también se puede proveer información relevante de la señal al calcular el delta y doble delta de cada coeficiente [JURAFSKY y MARTIN 2009]. Se podrían obtener así 39 coeficientes en total que representen las características más relevantes. Estos deltas muestran el cambio entre los valores de las características en el tiempo y la forma de calcularlas es la siguiente, donde c(t) es el valor cepstral en el tiempo t.

$$d(t) = \frac{c(t+1) - c(t-1)}{2}$$

En el presente trabajo, se tomará en cuenta el uso de los coeficientes obtenidos a partir del cálculo de doble delta.

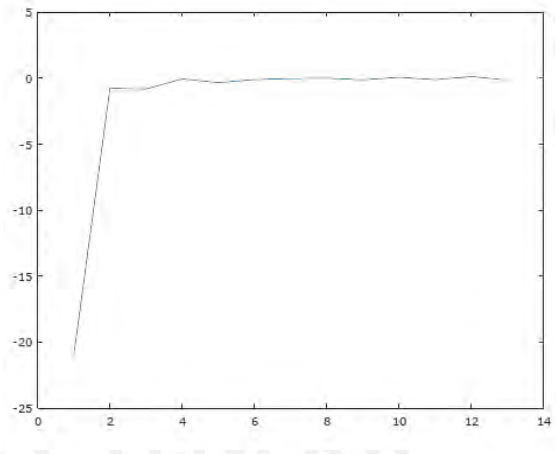
4.1.8. Conjunto de características

En la actual implementación del método de MFCC se obtiene un vector de 39 características por cada marco que se obtiene en la etapa de windowing, por lo que finalmente se genera una matriz de 13xn dimensiones, donde n es la cantidad de marcos generados. Por esto se decidió tomar el promedio de estos “n” marcos para una misma característica y generar un solo vector que represente a la señal introducida.



a) Señal inicial

b) Resultado MFCC, se observa la matriz generada de 13xn dimensiones donde n es la cantidad de marcos generados.



c) Resultado del promedio, se observa el vector final con el promedio de las 13 características base a partir de las cuales se calcularán los deltas y doble deltas

Figura 4.5 Resultados del método MFCC: a) señal inicial, b) resultado del MFCC y c) resultado promedio para obtener el vector de características final

4.2. Método PLP

4.2.1. Introducción

El segundo método que se ha optado por utilizar es el de coeficientes de predicción lineal perceptivos que de aquí en adelante, será abreviado como PLP por sus siglas en inglés *perceptual linear-prediction*. Este método, al igual que el primero, es ampliamente utilizado en el reconocimiento de voz humana y sigue inicialmente los mismos pasos de procesamiento que el método MFCC. A continuación mostraremos un esquema de las etapas del método.

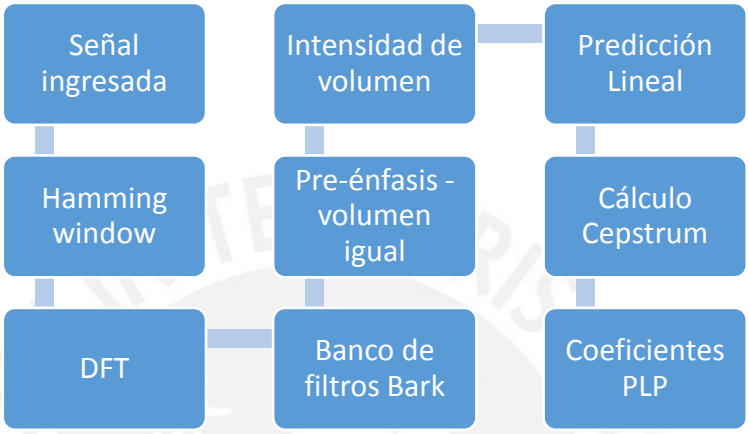


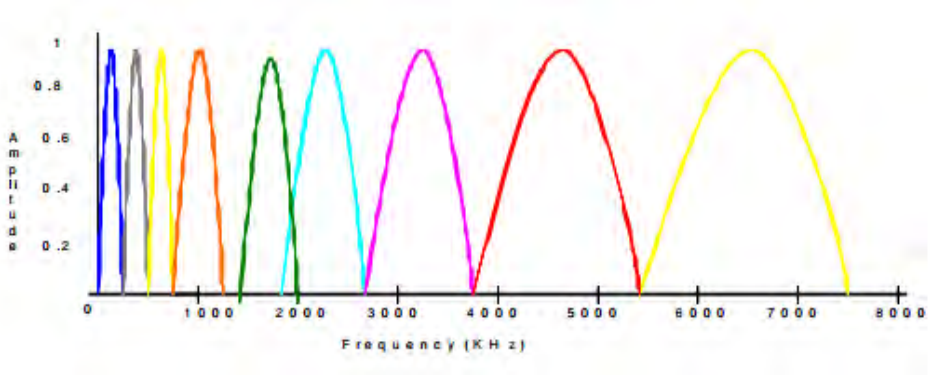
Figura 4.6 Esquema del funcionamiento del método PLP

Se puede apreciar que las dos primeras etapas, tanto *windowing* como la Transformada Discreta de Fourier (la cual se abrevia como DFT en el gráfico), también se encuentran en el método anteriormente descrito, por lo cual no se detallarán en este capítulo.

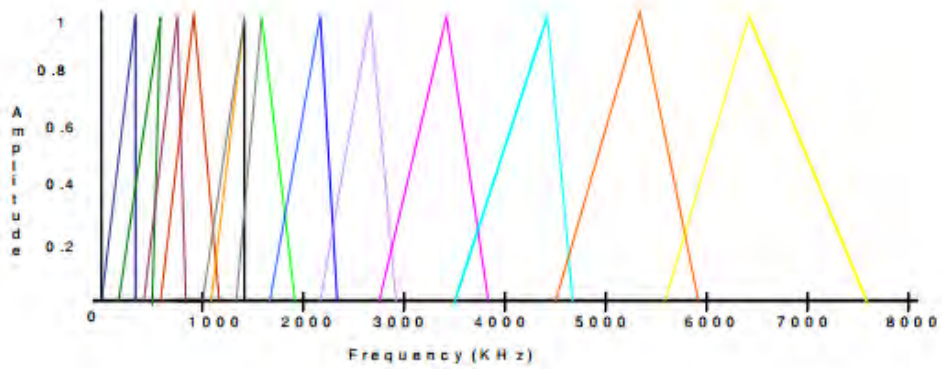
4.2.2. Banco de filtros Bark

Durante esta etapa, se realiza un proceso similar a la etapa de Banco de Filtros Mel en el método de MFCC. Principalmente, se convierte la señal ingresada a la escala de Bark mediante un envolvente de la frecuencia. Luego, se suaviza la señal y se reduce la proporción del muestreo a intervalos de aproximadamente 1 Bark. Estos tres pasos son integrados en un solo banco de filtros llamado banco de filtros Bark [DAVE 2013].

La diferencia principal entre los filtros Bark y Mel es el número y el ancho de los filtros, teniendo este último una mayor cantidad de filtros y un ancho más reducido produciéndose una superposición de hasta la mitad de cada uno [HONIG Y OTROS 2005]. Esto ocurre debido a que la escala de Mel está diseñada para simular un filtro pasa banda que es aproximadamente lineal por debajo de 1 kHz en el dominio de la frecuencia, mientras que la escala de Bark considera que la escala se vuelve lineal por debajo de los 500 Hz [GHOSH Y OTROS 2012].



(a) Banco de filtros Bark



(b) Banco de filtros Mel

Figura 4.7 Diferencia entre a)Banco de filtros Bark y b)Banco de filtros Mel

4.2.3. Pre-énfasis e igualdad de volumen

En esta etapa se busca que la señal adquiera una mayor energía, por lo cual sería similar al pre-énfasis realizado en el método MFCC con la diferencia en que en este método se basan en el volumen de la señal para lograrlo teniendo en consideración la sensibilidad a la frecuencia que tiene el oído humano. Tal y como se describe en el nombre de la etapa, se busca que el volumen sea uniformizado, procesando cada coeficiente del espectro de energía de la señal que se multiplica por un peso que depende de la frecuencia en Hertz [HONIG Y OTROS 2005].

4.2.4. Intensidad de volumen

En esta etapa, los valores normalizados obtenidos de la etapa anterior son transformados de acuerdo a la función potencial de Stevens elevándolos a la potencia de 0.33. Esta conversión, disminuye la variabilidad dinámica de la señal y aplana los picos encontrados en la señal. El modelo espectral resultante tiene una forma suavizada con picos menos pronunciados [HONIG Y OTROS 2005].

4.2.5. Predicción Lineal y cálculo de cepstrum

Durante la etapa de predicción lineal, se calculan los coeficientes de predicción de una señal hipotética que tiene como espectro envolvente a un espectro de energía. En la última etapa, los coeficientes cepstrales son obtenidos a partir de estos

coeficientes por una recursión que es equivalente al logaritmo del modelo del espectro seguido de una transformada inversa de Fourier [DAVE 2013], que se logra de manera similar al método MFCC.

4.2.6. Conjunto de características

De igual forma que con el método MFCC, del método PLP se pueden obtener 13 coeficientes base, de los cuales se puede calcular el valor delta hasta dos veces, obteniendo un vector de características de 39 coeficientes. Debido a que se obtienen estos coeficientes por cada marco de la señal, también se calculará un promedio por marco para cada coeficiente.

5. Modelo de reconocimiento e identificación de audios

5.1. Flujo del modelo de identificación

El modelo planteado cuenta con dos fases principales: la fase de entrenamiento y la fase de clasificación. La fase de entrenamiento necesita un archivo en formato csv para iniciar. Este archivo debe contar con todos los vectores resultantes de las señales de audio procesadas. Cada línea debe contener 14 o 40 valores diferentes: 13 o 39 correspondientes al resultado del método MFCC según se usen los deltas o no, y 1 correspondiente a la clase a la que pertenece dicha señal. En este caso, se utilizó “0” para representar a las voces y “1” para representar a las pisadas.

La siguiente fase es la de clasificación. Esta fase necesita también de un archivo en formato csv y además el modelo ya entrenado obtenido de la primera fase. El archivo debe ser similar al de la primera fase, pero no debe contener la clase a la que pertenece cada señal de audio ya que el objeto de salida de este proceso será el archivo ingresado pero con las correspondientes clases que el modelo debió predecir correctamente.

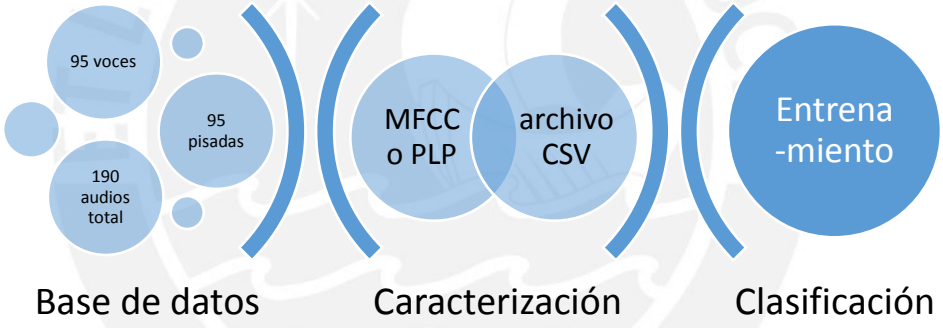


Figura 5.1Flujo y elementos de la etapa de entrenamiento

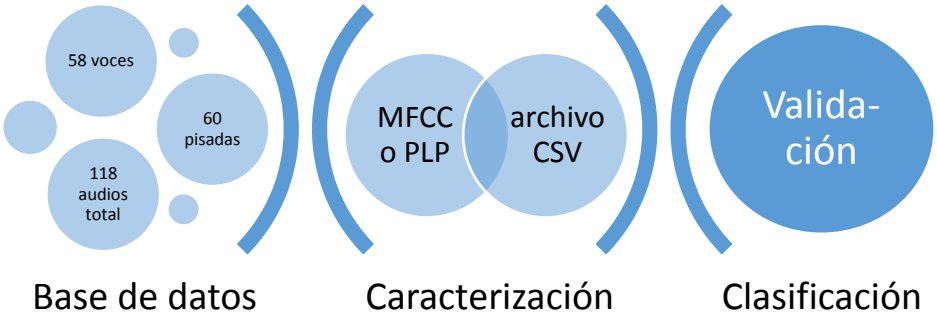


Figura 5.2 Flujo y elementos de la etapa de clasificación

5.2. Modelo de clasificación

Para la clasificación se utilizaron modelos de aprendizaje supervisados, en este caso se utilizó una red neuronal de varias capas.

5.3. Implementación del modelo

Para la implementación del modelo se utilizó la librería Weka bajo un entorno de desarrollo en Java. Este clasificador utiliza el algoritmo Backpropagation y los nodos de la red tienen una función sigmoide. Existen parámetros iniciales para esta implementación los cuales se detallan a continuación:

- Ratio de aprendizaje: 0.1
- Momentum: 0.2
- Tiempo de entrenamiento (iteraciones): 2000
- Capas ocultas: 8

Se debe aclarar que estos parámetros fueron escogidos de manera empírica puesto que no existe un método definido para calcularlos. Inicialmente, fueron basados en la configuración por defecto de Weka, y a partir de ello se hicieron variaciones que permitieron definir estos valores como los más apropiados para el presente trabajo.

5.4. Integración en el prototipo

El presente trabajo plantea un prototipo bajo el cual se pueda probar los resultados presentados de las etapas de clasificación y procesamiento. Es por esto que a continuación se presentan lo módulos obtenido finalmente y la interacción entre ellos. Para esto se ha definido que el prototipo se encontrará en lenguaje Java.

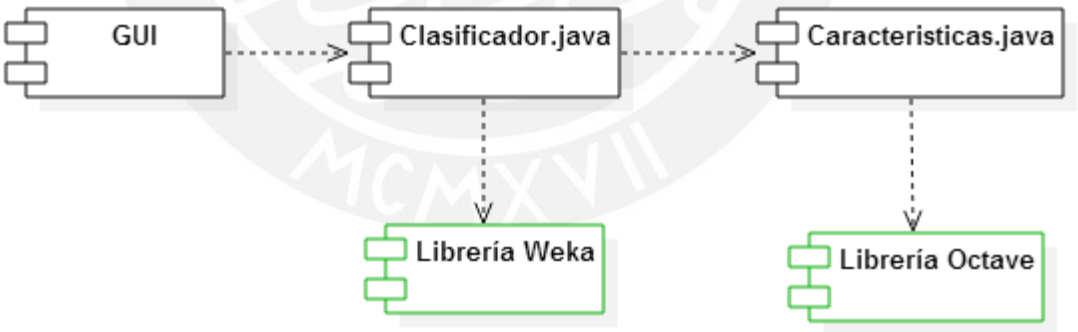


Figura 5.3 Esquema de la integración del prototipo

6. Resultados de la experimentación

6.1. Resultados con audios de entornos controlados

A continuación, se presentarán los diferentes escenarios con los cuales se procesaron los audios que fueron grabados en entornos controlados y poco ruido, para la diferenciación entre pisadas y voces humanas. Los datos de entrada fueron los mismos para todos los escenarios, teniendo un total de 118 audios para el entrenamiento, 58 de voces humanas y 60 de pisadas.

6.1.1. Escenario 1: Uso de MFCC con 13 coeficientes

Para la primera experimentación se observa que del total de audios, 96 fueron clasificados correctamente. Esto llevó a un resultado de 81% de precisión total. Dentro de los audios utilizados, 58 correspondieron a voces tanto de hombres como de mujeres, y 60 audios fueron de pisadas en distintos terrenos.

Dentro de los audios pertenecientes a voces, se observa que 44 fueron clasificados correctamente, lo que equivale a un 76% de precisión respecto a las voces. Por otro lado, de los 60 audios de pisadas, se obtuvo 52 aciertos en la clasificación, lo que representa al 87% de la cantidad total de audios. Así, podemos ver que hubo una mayor precisión sobre los audios de pisadas que sobre los audios de voces.

	Aciertos	Errores	Total
Voces	76% (44)	24% (14)	49% (58)
Pisadas	87% (52)	13% (8)	51% (60)
Total	81% (96)	19% (22)	100% (118)

Tabla 6.1 Resultados porcentuales de la clasificación de audios del escenario 1. Las cantidades se encuentran representadas entre paréntesis.

6.1.2. Escenario 2: Uso de MFCC con 39 coeficientes

Para la segunda experimentación, se observa un aumento en la precisión del algoritmo. En este escenario, la cantidad de aciertos en total aumentó a 114, que corresponde al 97% de audios clasificados correctamente. Dentro de ellos, los audios de voces clasificados correctamente corresponden a 54 de los 58 audios en total. Esto indica que el 93% del total de voces fue clasificado correctamente. Por otro lado, los audios de pisadas fueron correctamente clasificados en su totalidad, aumentando significativamente la precisión del algoritmo.

	Aciertos	Errores	Total
Voces	93% (54)	7% (4)	49% (58)
Pisadas	100% (60)	0% (0)	51% (60)
Total	97% (114)	3% (4)	100% (118)

Tabla 6.2 Resultados porcentuales de la clasificación de audios para el escenario 2. Las cantidades se encuentran representadas entre paréntesis.

6.1.3. Escenario 3 y 4: Uso de PLP con 13 y 39 coeficientes

En el escenario de uso del método PLP, tanto para 13 coeficientes como para 39 coeficientes, se observó una precisión del 100%, indicando que todos los audios de prueba fueron clasificados correctamente.

	Aciertos	Errores	Total
Voces	100% (58)	0% (0)	100% (58)
Pisadas	100% (60)	0% (0)	100% (60)
Total	100% (118)	0% (0)	100% (118)

Tabla 6.3 Resultados porcentuales de la clasificación de audios para el escenario 3 y 4. Las cantidades se encuentran representadas entre paréntesis.

6.1.4. Comparación de voces y pisadas

Dentro de los audios de voces, hemos observado un aumento continuo en la precisión del algoritmo al momento de clasificarlos correctamente. En los dos últimos escenarios, no se obtuvo ningún error en las voces y estos resultados los podemos observar en el gráfico siguiente. Este gráfico nos muestra por cada escenario los resultados respecto a los audios de voces, representando la proporción entre errores y aciertos del total de audios de voces puesto a prueba.

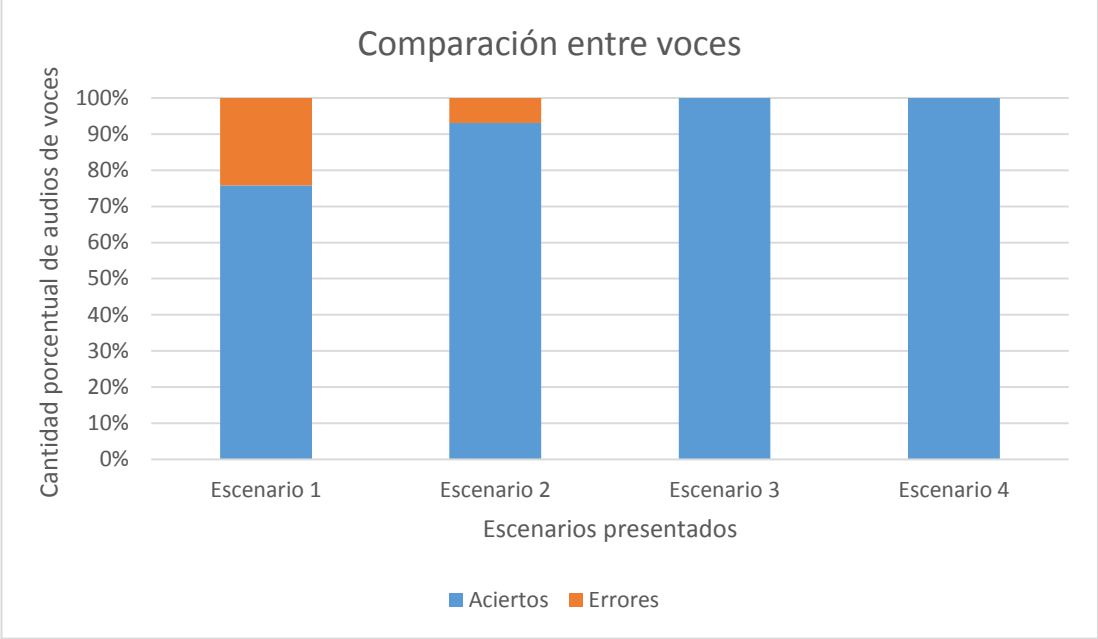


Figura 6.1 Gráfico de comparación entre resultados de voces en cada escenario

Respecto a los audios de pisadas, se aprecia que tan solo en el primer escenario se obtuvieron errores en la clasificación, mientras que en los tres escenarios siguientes, su precisión fue de un 100%. Estos resultados se pueden ver resumidos en el gráfico a continuación, en donde se tiene una barra representando cada escenario, y el porcentaje que representan los aciertos y errores del total de audios de pisadas que se tuvieron en el conjunto de pruebas.

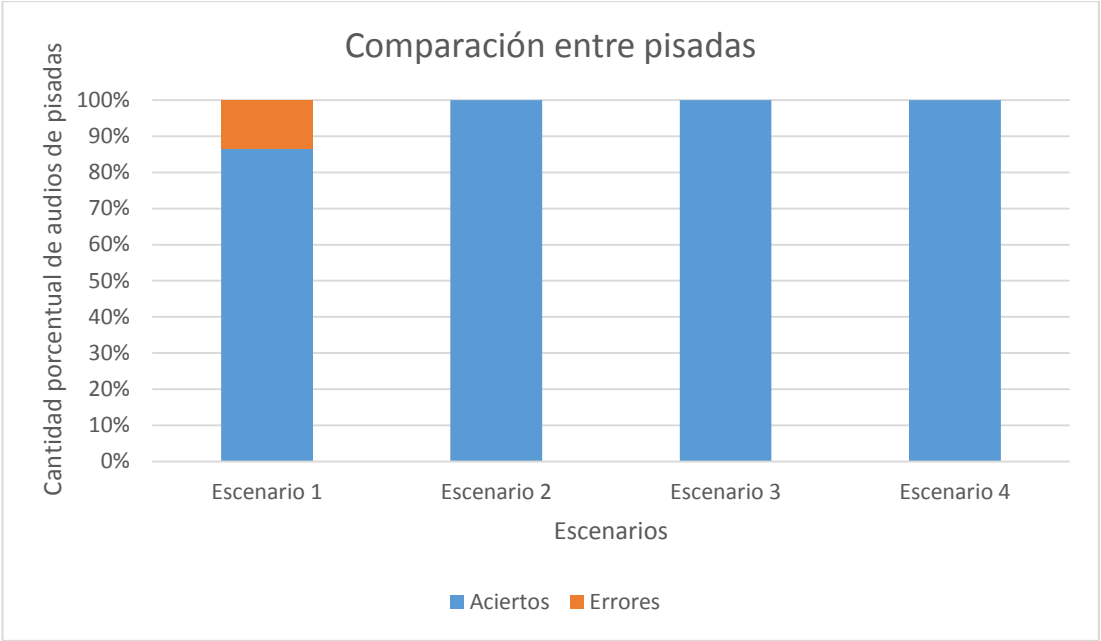


Figura 6.2 Gráfico de comparación entre resultados de pisadas en cada escenario

6.1.5. Comparación de métodos

Sobre los resultados con este conjunto de audios, se observa que en general la precisión ha ido aumentando en cada escenario, obteniéndose el mejor resultado en el método de PLP tanto con 13 como con 39 coeficientes. Así también, se resalta que han sido con los audios de pisadas que se ha tenido una mejor precisión al momento de la clasificación. Los resultados presentados se han resumido en la tabla que se presenta a continuación, la cual presenta una comparación entre los cuatro algoritmos y muestra la cantidad de audios que fueron correctamente clasificados en cada caso.

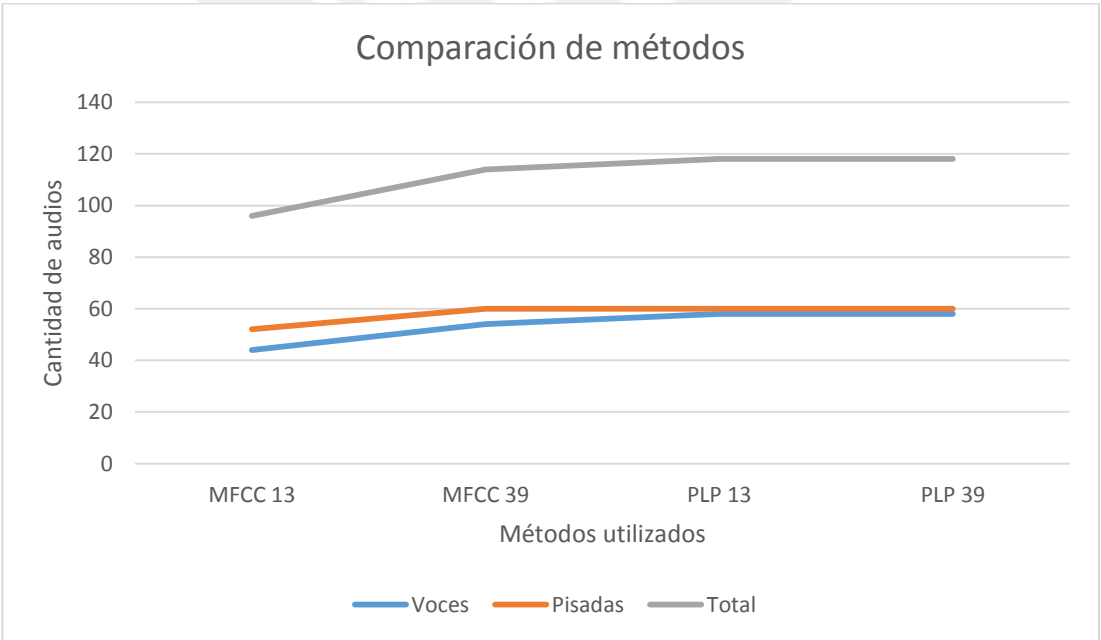


Figura 6.3 Gráfico de comparación entre resultados de cada método utilizado

6.2. Resultados de audios con sonidos mezclados

A continuación mostraremos 2 escenarios correspondientes a los resultados de la clasificación en audios de duración variable que contienen sonidos de pisadas o voces elegidos aleatoriamente e intercalados. Para esto, se utilizará el método de PLP con 39 coeficientes ya que, por un lado, el método PLP ha probado ser más efectivo respecto al MFCC para la clasificación. Y por otro lado, si bien se obtuvieron los mismos resultados con 13 y 39 coeficientes para PLP, en el caso de MFCC los resultados probaron ser mejores con 39 coeficientes, lo que podría implicar que el uso de 39 coeficientes sea más efectivo. En los escenarios se compararán los distintos métodos de pre-procesamiento y cómo influyen en el resultado final.

6.2.1. Escenario 1: Uso de envolvente de picos con muestreo 1000

Para el primer escenario se utilizó la envolvente basada en los picos de la frecuencia con un muestreo de cada 1000 muestras. De acuerdo a los resultados, la diferencia entre el tiempo inicial y final deseado y el obtenido a partir del pre-procesamiento, tiene en promedio una diferencia de 0.05 segundos, siendo de 0.02 para el inicio y 0.03 para el final. Los tiempos de inicio y fin ideales han sido identificados a partir de un análisis auditivo de la señal acústica de cada audio presentado, marcando los tiempos de acuerdo a la escucha del sonido y comparándolo con la gráfica de tiempo vs frecuencia respectiva. Una de las observaciones más relevantes es que algunos sonidos fueron divididos en dos, como ocurrió con la última pisada del audio 1. Esta pisada tenía tiempos ideales de inicio y fin de 0.958 y 1.501 respectivamente, y el modelo la segmentó como dos sonidos con tiempos de inicio y fin de 0.96, 1.36, 1.38 y 1.50 respectivamente, siendo ambos clasificados como pisadas. Por otro lado, también se obtuvieron errores los cuales se pueden ubicar en rojo. Estos errores ocurrieron en el módulo de pre-procesamiento, cuando se detectó un sonido cuando no había ninguno o también al no detectarse sonidos que sí se encontraban presentes en el audio. Para el primer caso, se ha optado por indicar la clase “silencio”, mientras que en el segundo caso la clase obtenida se ha dejado de la forma “—“, lo que indica que simplemente no se reconoció ni se clasificó este tramo de audio. Por estas razones es que finalmente de los 69 sonidos que debieron obtenerse, para este escenario se obtienen en realidad 80 sonidos. Parte de ellos es por la división de un mismo sonido, o por la detección de señales en las que no había picos que representaran un sonido a identificar. Estos casos se encuentran en la tabla presentada a continuación, donde se encuentra a detalle los resultados obtenidos de este primer escenario para cada archivo de audio.

Nombre Archivo	Clase	Tiempo inicio ideal	Tiempo fin ideal	Tiempo inicio obtenido	Tiempo fin obtenido	Clase obtenida
Audio 1	pisada	0.088	0.344	0.095	0.317	pisada
	pisada	0.43	0.695	0.428	0.698	pisada
	pisada	0.781	0.894	0.841	0.904	pisada
	pisada	0.958	1.501	0.968	1.365	pisada
				1.38	1.5	pisada

audio 2	pisada	0.138	0.657	0.143	0.651	pisada
	pisada	0.832	1.006	0.841	1.016	pisada
	pisada	1.19	1.835	1.206	1.667	pisada
				1.683	1.810	pisada
	voz	2.082	2.379	2.079	2.381	pisada
audio 3	pisada	0.225	0.844	0.2625	0.83125	pisada
	pisada	1.054	1.51	1.05	1.488	pisada
	voz	1.719	2.04	1.75	2.1	voz
	pisada	2.249	2.7	2.231	2.669	voz
audio 4	pisada	0.123	0.293	"--"	"--"	"--"
	silencio	0.3	0.53	0.381	0.444	pisada
	pisada	0.534	0.696	0.524	0.714	pisada
	voz	0.85	1.157	0.841	1.175	voz
	voz	1.295	1.596	1.317	1.619	pisada
audio 5	pisada	0.137	0.651	0.219	0.350	pisada
				0.394	0.700	voz
	voz	0.811	1.176	0.788	1.181	voz
	pisada	1.357	2.193	1.356	1.925	pisada
				2.013	2.231	voz
	pisada	2.441	2.908	2.450	2.800	pisada
				2.844	2.975	voz
audio 6	pisada	0.147	0.259	0.143	0.206	pisada
	voz	0.431	0.554	0.444	0.556	pisada
	pisada	0.715	0.856	"--"	"--"	"--"
	voz	1.009	1.24	1.032	1.254	pisada
audio 7	pisada	0.152	0.268	0.175	0.350	pisada
	voz	0.467	0.866	0.481	0.919	voz
	voz	1.1	1.24	1.050	1.400	voz
	pisada	1.429	1.598	1.444	1.663	pisada
audio 8	pisada	0.159	0.399	0.159	0.365	pisada
	voz	0.578	0.947	0.587	0.952	pisada
	voz	1.186	1.396	1.206	1.429	voz
	voz	1.607	1.914	1.619	1.937	pisada
audio 9	voz	0.123	0.398	0.131	0.481	voz
	pisada	0.554	0.743	0.525	0.788	voz
	pisada	0.889	1.116	0.831	1.181	voz
	pisada	1.271	1.418	1.269	1.488	pisada
audio 10	voz	0.15	0.35	0.131	0.393	pisada
	pisada	0.515	1.037	0.568	1.05	voz
	pisada	1.248	1.658	1.268	1.75	voz
	voz	1.863	2.059	1.88	2.1	voz
audio 11	voz	0.164	0.707	0.175	0.831	voz
	pisada	0.933	1.071	0.918	1.093	pisada
	voz	1.359	1.686	1.4	1.75	pisada

	pisada	1.93	2.23	1.925	2.493	voz
audio 12	voz	0.139	0.491	0.131	0.525	voz
	pisada	0.678	1.228	0.656	1.268	voz
	voz	1.444	1.783	1.443	1.793	voz
	voz	2.065	2.448	2.1	2.493	pisada
	pisada	2.792	3.208	2.8	3.237	voz
audio 13	silencio	0	0.15	0.044	0.131	voz
	voz	0.158	0.593	0.175	0.656	voz
	voz	0.816	1.051	0.875	1.138	voz
	pisada	1.317	1.557	1.313	1.663	pisada
	pisada	1.789	2.05	1.794	2.013	pisada
	silencio	2.05	2.33	2.056	2.144	voz
	voz	2.33	2.72	2.406	2.756	voz
audio 14	voz	0.133	0.516	0.143	0.524	voz
	pisada	0.682	0.828	0.683	0.746	pisada
				0.762	0.825	voz
	pisada	1.022	1.258	"--"	"--"	"--"
	voz	1.402	1.733	1.413	1.762	voz
	pisada	1.932	2.145	1.937	2.048	pisada
				2.079	2.159	pisada
audio 15	voz	0.115	0.364	0.131	0.394	voz
	voz	0.547	0.836	0.613	0.919	voz
	voz	1.019	1.213	1.050	1.313	voz
	pisada	1.379	1.491	1.400	1.488	voz
	pisada	1.691	1.808	1.663	1.881	voz
audio 16	voz	0.121	0.38	0.131	0.438	pisada
	voz	0.613	0.885	0.656	0.919	voz
	voz	1.081	1.227	1.094	1.269	pisada
	voz	1.43	1.668	1.444	1.706	voz
	pisada	1.879	2.395	1.925	2.275	pisada
				2.319	2.450	voz

Tabla 6.4 Resultados detallados por audios para envolvente con picos

Finalmente, se obtiene que solo 58 sonidos fueron detectados adecuadamente, representando un 84% de precisión del módulo de pre-procesamiento. Respecto a la clasificación de las clases, para poder calcular la precisión en los casos en los que el sonido fue dividido en dos, se ha considerado sumar 0.5 por cada mitad que fue clasificada correctamente de acuerdo a la clase del sonido completo. Es así que habría un total de 44.5 aciertos, representando una precisión de 61%.

	Aciertos	Errores	Total
Voces	72% (23)	28% (9)	46% (32)
Pisadas	58% (21.5)	42% (15.5)	54% (37)
Total	64% (44.5)	36% (24.5)	100% (69)

Tabla 6.5 Resultados porcentuales de la clasificación de audios. Las cantidades se encuentran representadas entre paréntesis.

6.2.2. Escenario 2: Uso de envolvente de RMS con muestreo de 1000

Para este escenario se utilizó la función envolvente RMS con muestreo de 1000. En este caso, la diferencia entre los tiempos iniciales y finales deseados y obtenidos es menor respecto al caso anterior. Para los tiempos de inicio, la diferencia es en promedio de 0.02 segundos mientras que para los tiempos de fin, el promedio es de 0.03. Además de esto, se ve una notable mejora respecto a la identificación de sonidos en dos partes, ya que solo ocurre este caso en el audio 2, para la tercera pisada. El resto de sonidos fueron identificados correctamente, además de no haber errores respecto a intervalos de tiempo de silencios identificados como sonidos. Por esto, para esta tabla no se encontrarán valores en rojo. A continuación se presenta la tabla con los resultados detallados por cada audio.

Nombre Archivo	Clase	Tiempo inicio ideal	Tiempo fin ideal	Tiempo inicio obtenido	Tiempo fin obtenido	Clase obtenida
Audio 1	pisada	0.088	0.344	0.095	0.349	pisada
	pisada	0.43	0.695	0.444	0.698	pisada
	pisada	0.781	0.894	0.794	0.905	pisada
	pisada	0.958	1.501	0.968	1.508	pisada
audio 2	pisada	0.138	0.657	0.143	0.667	pisada
	pisada	0.832	1.006	0.841	1.016	pisada
	pisada	1.19	1.835	1.206	1.778	pisada
				1.794	1.857	voz
	voz	2.082	2.379	2.095	2.397	voz
audio 3	pisada	0.225	0.844	0.263	0.744	pisada
	pisada	1.054	1.51	1.094	1.488	pisada
	voz	1.719	2.04	1.750	2.100	voz
	pisada	2.249	2.7	2.275	2.669	pisada
audio 4	pisada	0.123	0.293	0.127	0.302	pisada
	pisada	0.534	0.696	0.540	0.714	pisada
	voz	0.85	1.157	0.857	1.175	voz
	voz	1.295	1.596	1.302	1.619	pisada
audio 5	pisada	0.137	0.651	0.175	0.656	voz
	voz	0.811	1.176	0.831	1.225	pisada
	pisada	1.357	2.193	1.356	2.231	voz
	pisada	2.441	2.908	2.450	2.931	pisada
audio 6	pisada	0.147	0.259	0.159	0.286	pisada
	voz	0.431	0.554	0.460	0.571	pisada
	pisada	0.715	0.856	0.730	0.841	pisada
	voz	1.009	1.24	1.016	1.254	pisada
audio 7	pisada	0.152	0.268	0.175	0.350	pisada
	voz	0.467	0.866	0.525	0.919	voz
	voz	1.1	1.24	1.138	1.313	voz
	pisada	1.429	1.598	1.444	1.663	pisada
audio 8	pisada	0.159	0.399	0.175	0.413	pisada

	voz	0.578	0.947	0.587	0.968	pisada
	voz	1.186	1.396	1.190	1.413	voz
	voz	1.607	1.914	1.619	1.937	pisada
audio 9	voz	0.123	0.398	0.131	0.438	voz
	pisada	0.554	0.743	0.569	0.788	voz
	pisada	0.889	1.116	0.919	1.181	pisada
	pisada	1.271	1.418	1.313	1.488	pisada
audio 10	voz	0.15	0.35	0.175	0.438	pisada
	pisada	0.515	1.037	0.525	1.050	voz
	pisada	1.248	1.658	1.269	1.706	pisada
	voz	1.863	2.059	1.881	2.100	voz
audio 11	voz	0.164	0.707	0.175	0.744	voz
	pisada	0.933	1.071	0.963	1.138	pisada
	voz	1.359	1.686	1.356	1.750	voz
	pisada	1.93	2.23	1.969	2.494	pisada
audio 12	voz	0.139	0.491	0.175	0.569	pisada
	pisada	0.678	1.228	0.700	1.225	voz
	voz	1.444	1.783	1.444	1.838	voz
	voz	2.065	2.448	2.100	2.494	pisada
	pisada	2.792	3.208	2.800	3.238	voz
audio 13	voz	0.158	0.593	0.175	0.656	voz
	voz	0.816	1.051	0.831	1.138	voz
	pisada	1.317	1.557	1.356	1.575	pisada
	pisada	1.789	2.05	1.794	2.056	pisada
	voz	2.33	2.72	2.363	2.756	voz
audio 14	voz	0.133	0.516	0.143	0.524	voz
	pisada	0.682	0.828	0.698	0.841	pisada
	pisada	1.022	1.258	1.032	1.270	pisada
	voz	1.402	1.733	1.413	1.762	voz
	pisada	1.932	2.145	1.937	2.175	pisada
audio 15	voz	0.115	0.364	0.131	0.438	voz
	voz	0.547	0.836	0.569	0.919	voz
	voz	1.019	1.213	1.050	1.269	voz
	pisada	1.379	1.491	1.400	1.531	pisada
	pisada	1.691	1.808	1.706	2.056	voz
audio 16	voz	0.121	0.38	0.131	0.438	pisada
	voz	0.613	0.885	0.613	0.963	voz
	voz	1.081	1.227	1.094	1.269	pisada
	voz	1.43	1.668	1.444	1.706	voz
	pisada	1.879	2.395	1.881	2.450	voz

Tabla 6.6 Resultados detallados por audios para envolvente con RMS

En este escenario, la precisión del módulo de pre-procesamiento mejoró notablemente. De los 69 sonidos, 68 fueron reconocidos correctamente, representando un 98% de precisión en el módulo de pre-procesamiento. Esto

también nos lleva a tener una mejor precisión en el momento de clasificación. Siguiendo el mismo esquema que en el escenario anterior, donde se sumaba una cantidad de 0.5 aciertos por cada parte del sonido reconocida correctamente (en el caso de audios divididos), tenemos un total de 49.5 aciertos. Esto representaría a un 72% de precisión del algoritmo en su totalidad.

	Aciertos	Errores	Total
Voces	66% (21)	34% (11)	46% (32)
Pisadas	77% (28.5)	23% (8.5)	54% (37)
Total	72% (49.5)	28% (19.5)	100% (69)

Tabla 6.7 Resultados porcentuales de la clasificación de audios. Las cantidades se encuentran representadas entre paréntesis.

6.3. Resultados con audios no controlados

En este escenario, se decidió juntar los audios y obtener un audio largo de 25 segundos. Se utilizó el método de recorte de envolvente RMS y la clasificación se hizo con el método PLP de 39 coeficientes. Los resultados se pueden ver en la tabla a continuación.

Clase	Tiempo inicio ideal	Tiempo fin ideal	Tiempo inicio obtenido	Tiempo fin obtenido	Clase obtenida
Pisada	2.613	2.905	"--"	"--"	"--"
Pisada	3.347	3.657	3.365	3.444	pisada
Pisada	4.132	4.424	4.127	4.254	pisada
Pisada	4.929	5.118	4.92	5.031	pisada
Pisada	5.732	5.981	5.714	5.841	voz
Pisada	6.505	6.754	6.507	6.65	pisada
Pisada	7.253	7.497	7.254	7.333	voz
silencio	"—"	"—"	7.36	7.49	pisada
Pisada	8.032	8.273	8.03	8.127	pisada
Pisada	8.826	9.103	8.825	8.88	pisada
Voz	9.53	9.846	9.53	9.634	voz
Pisada	10.258	10.519	10.254	10.333	voz
Pisada	10.967	11.142	10.952	11.079	pisada
Voz	14.089	14.654	14.079	14.603	voz
Voz	15.589	16.076	15.571	16.063	voz
Voz	16.955	17.364	16.952	17.349	pisada
silencio	"—"	"—"	18.44	18.55	voz
Pisada	19.637	19.89	19.619	19.746	voz
Pisada	20.317	20.524	20.301	20.444	pisada
Pisada	20.972	21.156	20.968	21.111	pisada
silencio	"—"	"—"	21.126	21.206	pisada
pisada	21.628	21.886	21.619	21.761	pisada
silencio	"—"	"—"	21.761	21.873	voz
voz	22.262	22.383	22.254	22.365	voz

Tabla 6.8 Resultados detallado por tiempos y clasificación

En este caso podemos ver que ocurren casos similares del escenario que usaba la envolvente de picos, del escenario 1 de la sección anterior. De la misma forma, el módulo de recorte no logra identificar adecuadamente las regiones de sonidos buscados, por lo que omite el primer sonido, y en los demás casos marca como sonidos lo que en verdad son ruidos de fondo. Se debe tener en consideración también que el ruido aumenta considerablemente en este audio. Por lo mismo, el valor de límite entre ruido y sonido se debió modificar respecto a la experimentación anterior. En este caso, se estableció el valor a 0.05. Respecto a la clasificación, podemos ver que hay 14 aciertos en total, respecto a 24 sonidos diferentes en los cuales se incluyen los silencios que fueron recortados como sonidos relevantes (en total son 4). Esto representaría un 58% de precisión del algoritmo. Del total de sonidos, 15 pertenecen a pisadas y 5 pertenecen a voces. De ellos, 10 aciertos fueron de pisadas y 4 de voces. Estos resultados están resumidos en la siguiente tabla.

	Aciertos	Errores	Total
Voces	80% (4)	20% (1)	25% (5)
Pisadas	67% (10)	33% (5)	75% (15)
Total	70% (14)	30% (6)	100% (20)

Tabla 6.9 Resultados porcentuales de la clasificación de audios. Las cantidades se encuentran representadas entre paréntesis.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

En el presente trabajo, se ha estudiado y desarrollado un algoritmo que permitiría reconocer las pisadas y voces humanas evaluando los patrones que tienen las señales de audio de cada una. A continuación se presentan las conclusiones.

En primer lugar, respecto al módulo de pre-procesamiento se concluye que es mejor utilizar una señal envolvente del tipo RMS ya que identifica mejor las regiones de interés en la señal acústica. Sin embargo, se debe tener a consideración el valor mínimo especificado que establece el límite entre un sonido significativo, y ruido o sonido de fondo. En este trabajo, debido a que los audios no tenían un nivel significativo de ruido y era homogéneo en todos ellos, fue fácil establecer empíricamente un nivel adecuado de umbral entre el sonido y el fondo del audio. Pero en un caso donde el entorno tenga mucho ruido o no esté controlado, el módulo de pre-procesamiento se volverá mucho más complejo pudiendo llegar incluso a necesitar de un algoritmo de aprendizaje supervisado que le permita identificar los sonidos.

En segundo lugar, se utilizaron dos métodos para la extracción del vector de características. Uno de ellos, utilizaba la escala de Mel mientras que el otro método utilizaba la escala de Bark para representar las señales de audio. Estas escalas difieren básicamente en el rango de frecuencia que toman para la unidad de medida. Ambas trabajan con énfasis de la señal y con el dominio del espectro. De estos dos métodos, más eficaz probó ser el que utilizó las escalas de Bark, obteniendo mejores resultados con los conjuntos de datos probados. Esto se origina debido a las frecuencias de los audios que se utilizan y los sonidos a reconocer. Probablemente, el comportamiento lineal de la escala Bark debajo de 500 Hz estaría representando mejor las señales de voces y pisadas humanas al momento de extraer las características del dominio de la frecuencia y el espectro de la señal.

En tercer lugar, se utilizaron dos conjuntos de datos para las pruebas. Un conjunto tenía audios en ambientes controlados mientras que otro conjunto fue extraído de ambientes con ruido moderado. En estos resultados se ve la diferencia entre ambos escenarios, ya que no solo disminuyó la precisión considerablemente, sino que también se observa una tendencia en los sonidos de pisadas a tener una menor cantidad de aciertos respecto a los sonidos de voces. Analizando los audios de pisadas, se concluye que estos difieren en ambos conjuntos, ya que las pisadas en entornos controlados tienen un volumen alto y es fácil identificarlas, mientras que las pisadas en entornos no controlados no son tan fáciles de identificar al oído humano y se confunden fácilmente con el ruido de fondo. Al respecto se podría concluir que es necesario un mayor pre-procesamiento específicamente para las pisadas en entornos ruidosos, donde sería lo ideal poder reducir el ruido al mínimo y elevar el volumen e intensidad que genera el ruido de la pisada.

Por último, se concluye que el algoritmo propuesto depende mucho de la calidad de audio que se provee para clasificar. Por esto, es importante considerar un módulo

de limpieza de la señal, de modo que el algoritmo propuesto no solo tenga buenos resultados con audios limpios de ruido y en volumen elevado, sino también con audios que puedan ser captados en ambientes no controlados.

7.2. Trabajos futuros

En esta sección, se presentan algunas propuestas que no se han podido cubrir en el presente trabajo pero que están estrechamente relacionadas y ayudarían tanto a mejorar los resultados como proveer una mejor explicación sobre el funcionamiento del algoritmo.

Desarrollo de un módulo de limpieza de las señales acústicas recibidas, que pueda reducir al mínimo la cantidad de ruido en el fondo del audio y también que permita resaltar la frecuencia en la que se encuentra el sonido buscado, de modo que se genere una señal con mayor volumen o mayor intensidad.

Desarrollo de un clasificador que considere los sonidos de fondo. Esto se refiere a que, en caso no se haga un adecuado recorte de la señal, de todas formas no habría problema al momento de clasificación porque se habría entrenado previamente al algoritmo con sonidos que representen al ruido de fondo de la señal. De esta forma ya no se clasificaría ni como pisada ni como voz, sino simplemente como ruido de fondo y probablemente aumentaría la precisión del clasificador y disminuye su dependencia del módulo de recorte.

8. Bibliografía

ABDALLA, Mahmoud y ALI, Hanaa
2010 "Wavelet-Based Mel-Frequency Cepstral Coefficients for Speaker Identification using Hidden Markov Models". In *Journal of Telecommunications.* Vol 1. No 2. Pp 16-21

ADAMS, Stephen y otros
2016 "Feature Selection for Hidden Markov Models and Hidden Semi-Markov Models". In *IEEE Access.* Vol 4. PP 1642-1657.

ALIAS, E. y otros
2014 "A Novel Acoustic Fingerprint Method for Audio Signal Pattern Detection". In *Fourth International Conference on Advances in Computing and Communications.* Pp 64-68.

APPLE
2014 "Siri". Consultado: 05 de junio del 2016.
http://www.apple.com/ios/siri/

BARFORD, Lee y otros
1992 "An Introduction to Wavelets".

http://www.hpl.hp.com/techreports/92/HPL-92-124.pdf

BENETSY, Jacob y otros
2009 "Noise Reduction Algorithms in a Generalized Transform Domain". In *IEEE transactions on audio, speech, and language processing.* Vol. 17, No. 6.

BLUNSOM, Phil
2004 "Hidden Markov Models".
http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf

CHEN, Jingdong y otros
2006 "New Insights Into the Noise Reduction Wiener Filter". *In IEEE transactions on audio, speech, and language processing*. Vol. 14. No. 4

COCHRAN, William
1967 "What is the Fast Fourier Transform?". In *IEEE Transactions on Audio and Electroacoustics.* Vol 15. No. 2

CRISTANI, M. y otros
2007 "Audio-Visual Event Recognition in Surveillance Video Sequences" In *IEEE Transactions on multimedia,* Vol. 9, No. 2. Pp 257-267

CZUNI, Laszlo y ZOLTAN, Peter
2014 "Lightweight accoustic detection of logging in Wireless sensor networks". In *The Internatioinal Conference of Digital Information, Networking and Wireless Communications.* Pp 120 – 125.

DAVE, Namrata
2013 "Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition". In I*nternational Journal for Advance Research in Engineering and Technology.* Vol 1. No 6.

EKIMOV, Alexander y SABATIER, James M.
2010 "Rhythm analysis of orthogonal signals from human walking". In *The Journal of the Acoustical Society of America.*

GENCOGLU, O. y otros
2014 "Recognition of acoustic events using deep neural networks". In *22nd European Signal Processing Conference (EUSIPCO).* Pp 506-510.

GHOSH, D. y otros
2012 "A comparative study of performance of fpga based mel filter bank & bark filter bank". In *International Journal of Artificial Intelligence & Applications (IJAIA).* Vol 3. No 3. Pp 37-54.

GRIGORYAN, Ayrton
2005 "Fourier Transform Representation by Frequency-Time Wavelets". In *IEEE Transactions on Signal Processing.* Vol 53. No 7.

GUODONG, Guo
2014 "Soft Biometrics from Face Images Using Support Vector Machines". In *Support Vector Machines Applications*. Pp 269-270.

GUYON, Isabelle
2006 "Feature extraction, foundations and applications". Springer. Pp. 1-23.

HONIG, Florian y otros
2005 "Revising Perceptual Linear Prediction (PLP)". In *Interspeech 2005.*Pp. 2997-3000.

INSTITUTO NACIONAL DE ESTADISTICA E INFORMÁTICA
2014 "Perú: Anuario de estadísticas ambientales 2013". Lima: INEI.

https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1140/

JAIN, Anil K. y otros
2004 "An introduction to biometric recognition". In *IEEE Transactions on circuits and systems for video technology.* Vol 14. No 1.

JURAFSKY, Daniel y MARTIN, James
2009 "Automatic Speech Recognition" *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Pearson Education, pp. 11-19.

KHODASKAR, A.A. y LADHAKE, S.A.
2014 "Pattern Recognition: Advanced Development, Techniques and Application for Image Retrieval". In *International Conference on Communication and Network Technologies (ICCNT).* PP.74-78.

KOTSIANTIS, S.B. y otros
2007 "Data preprocessing for Supervised Learning". In *International Journal of Computer, Electrical, Automation, Control and Information Engineering.* Vol. 1. No.12.

MATHWORKS

2016 “MATLAB Product Description”. Consultado: 04 de junio del 2016
http://www.mathworks.com/help/matlab/learn_matlab/product-description.html

MEHMOOD, Asif y otros
2012 “Discrimination of bipeds from quadrupeds using seismic footstep signatures”. In *IEEE International Geoscience and Remote Sensing Symposium*.

METCALFE, Tom
2017 “Can these drones save elephants from extinction?”. *NBC News.* 14 de noviembre. Consulta: 28 de octubre de 2018.

https://www.nbcnews.com/mach/science/can-these-drones-save-elephants-extinction-ncna820441

MICROSOFT
2015 “Visual Studio 2015”. Consultado: 01 de junio de 2016
https://www.visualstudio.com/vs-2015-product-editions

MITRA, V. y otros
2014 “Medium-duration Modulation Cepstral Feature for Robust Speech Recognition”. In *IEEE International Conference on Acoustic, Speech and Signal Processing.* Pp 1749-1753

MUDA, Lindasalwa y otros
2010 “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques”. In *Journal of Computing.* Vol 2. No. 3

NOBLE, William
2006 “What is a support vector machine?”. In *Nature Biotechnology.* Vol 24. No 24. PP 1565-1567.

OCTAVE
2017 “About GNU Octave”. Consultado: 06 de julio de 2017
https://www.gnu.org/software/octave/about.html

PAPAVASSILIOU, Christos
2008 “Transducers and Sensors”. Material del curso *Instrumentation.* Londres: Imperial College. Consulta: 22 de Abril de 2016

http://cas.ee.ic.ac.uk/people/dario/files/E302/1-Sensors.pdf

PARK, Hyung O. y otros
2009 “Cadence analysis of temporal gait patterns for seismic discrimination between human and quadruped footsteps”. In *IEEE International Conference on Acoustics, Speech and Signal Processing.*

PATIL, Rajesh
2015 “Noise Reduction using Wavelet Transform and Singular Vector Decomposition”. In *Eleventh International Multi-Conference on Information Processing – 2015.*PP 849-853

PRESS, William

2008 "Unit 19: Wiener filtering (and some Wavelets)". Material del curso *Computational Statistics with Application to Bioinformatics*. Consulta: 29 de Mayo de 2016

http://numerical.recipes/CS395T/lectures2008/19-WienerFiltering.pdf

SADLIER, D.A. y otros
2011 "Image-based Vehicle Indexing for a Seaport Transportation Surveillance System". In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance.* Pp 367-372

SANDBERG, Kristian
2003 "Introduction to MATLAB". Consultado: 04 de junio de 2016
http://www.math.utah.edu/~wright/misc/matlab/matlabintro.html

SHE, B.
2004 "Framework of footstep detection in in-door environment". In *18th International Congress on Acoustics.* Pp 715-718.

SKINNER, Nicole
2014 "African elephant numbers collapsing". *Nature News.* 19 de agosto. Consulta: 22 de abril de 2016.

http://www.nature.com/news/african-elephant-numbers-collapsing-1.15732

VALENZISE, G. y otros
2007 "Scream and Gunshot Detection and Localization for Audio-Surveillance Systems". In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on.* Pp 21-26.

WALL, Matthew
2014 "Can drones help tackle Africa's wildlife poaching crisis?". *BBC News.* 21 de julio. Consulta: 22 de abril de 2016.

http://www.bbc.com/news/business-28132521

WANG, C. y otros
2015 "Automatic Recognition of Audio Event Using Dynamic Local Binary Patterns". In *International Conference on Consumer Electronics-Taiwan (ICCE-TW).* Pp 246-247.

WANG, Lipo
2005 "Support Vector Machines: Theory and applications". Springer Science & Business Media

XU, Yansun y otros.
1994 "Wavelet Transform Domain Filters: A Spatially Selective Noise Filtration Technique". In *IEEE Transactions on image processing*. Vol. 3. No. 6.

YUAN, Xiao-Chen y otros
2014 "Robust Mel-Frequency Cepstral coefficients feature detection and dual-tree complex wavelet transform for digital audio watermarking". In *Information Sciences.* Vol 298. Pp 159-179

ZAJDEL, W. y otros

2007 “CASSANDRA: audio-video sensor fusion for aggression detection” In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on.* Pp. 200-205

ZHANG, Yongli y YANG, Yuhong
2015 “Cross-validation for selecting a model selection procedure”. In *Journal of Econometrics.* Vol 187. No. 1. PP 95-112

ZIEGER, C. y otros
2009 “Acoustic Based Surveillance System For Intrusion Detection”. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*. Pp. 314-319.