# Capítulo 3. Síntesis de literatura consultada

**Estado del Arte**

En esta sección del capítulo se hará un resumen de algunas investigaciones encontradas acerca de los temas a tratar en el proyecto.  
Revisar todo el conjunto de saberes o el desarrollo que se ha conseguido en el área en la cual se va a investigar es clave para lograr el objetivo propuesto para toda esta investigación.

El estado del arte se dividirá en tres importantes pilares, la detección de audio, el almacenamiento de datos y por último se verá aplicaciones similares, las cuales cumplen con un objetivo similar al que se desea lograr.

**Detección de Audio**:

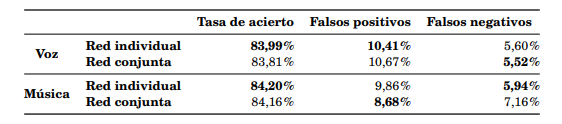
Primer Investigación:

* Título: “*Detección de voz y música en un corpus a gran escala de eventos de audio*”.
* Objetivo:   
    
  1. Estudio, comprensión y descripción de la base de datos AudioSet de Google.   
    
  2. Consulta del estado del arte del área, experimentos relacionados y resultados previos.   
    
  3. Diseño de modelos de clasificación basados en diferentes arquitecturas de redes neuronales.   
    
  4. Diseño y ejecución de experimentos para la evaluación del rendimiento de los modelos de clasificación propuestos.   
    
  5. Interpretación y discusión de los resultados obtenidos.   
    
  6. Generación de código orientado al manejo de la base de datos AudioSet y su uso en futuras investigaciones.
* Muestra: 77.396 segmentos de audio de 10 segundos (216 horas) pertenecientes a vídeos de YouTube, extraídos de la reciente base de datos AudioSet de Google.

Evaluados con arquitecturas de redes neuronales, basadas en DNN, CNN y LSTM.

* Palabras Claves: Aprendizaje automático, deep learning, redes neuronales, redes convolucionales, LSTM, audio, voz, música, reconocimiento de eventos de audio.
* Resultados:

Las redes entrenadas para las tareas de detección de voz (4.1.1), detección de música (4.1.2) y detección conjunta de voz y de música (4.1.3) alcanzan resultados muy similares en cuanto a tasas de acierto, que se recogen en la tabla 4.4. Las tasas de falsos positivos y falsos negativos se expresan como porcentaje sobre el total de segmentos clasificados. En ambos casos (música y voz), la precisión es levemente superior con las redes individuales, pero no lo suficiente como para considerarlo una diferencia significativa.



* Conclusión

Han sido evaluadas diferentes arquitecturas de redes neuronales, basadas en DNN, CNN y LSTM. Los resultados indican un rendimiento muy destacable de las arquitecturas basadas en redes convolucionales, que logran rendimientos muy similares en voz y en música, con tasas de acierto en torno al 84%. También se ha destacado el rendimiento de la red conjunta, que alcanza resultados semejantes a los de las dos redes individuales combinadas, pero contando aproximadamente el mismo número de parámetros entrañables que sólo una de ellas.

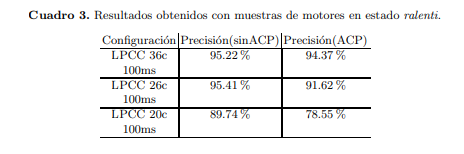
Segunda Investigación:

* Título: “*Análisis y Clasificación de Sonidos de Motores de Carros, para Diagnosticar Fallas, con Ayuda de Redes Neuronales*”.
* Objetivo:   
    
  El objetivo de la investigación es reconocer problemas en el motor a traves de su sonido, utilizando las vibraciones que este realiza cuando existe algún tipo de falla. Con ayuda también de redes neuronales.
* Muestra:

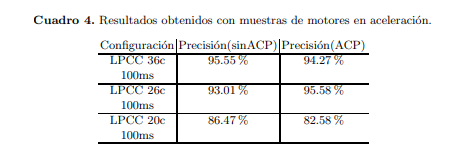
Las muestras de los sonidos fueron sacadas de tres talleres se especializan en motores de autos. Cada muestra grabada fue obtenida con ayuda de una grabadora digital ZicPlay y los archivos de salida fueron guardados en formato WAV, con las siguientes características: 8 khz, 16 bits y canal monoaural. Cada uno de estos archivos tiene un periodo de tiempo de 3 a 5 segundos, y fueron grabados en total 26 diferentes máquinas. Estas muestras fueron clasificadas en 4 categorías: motores en buenas condiciones (clase 1), problemas relacionados con el cigüeñal (clase 2), con pistones (clase 3) y con problemas en válvulas (clase 4)

* Resultados:

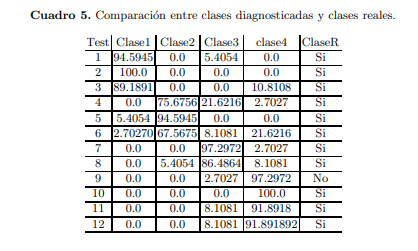
En el Cuadro 3, se pueden ver los resultados obtenidos usando muestras de máquinas en estado ralenti, el Cuadro 4, muestra los resultados obtenidos de muestras grabadas de máquinas en aceleración. Cada resultado, es el obtenido de la media aritmética de 10 experimentos previos cada uno. En ambos cuadros, la primera columna muestra el número de características extraídas por cada segmento de tiempo (ms).



Los resultados en ambas tablas muestran que la mejor precisión fue dada cuando usamos 36 y 26 LPCC por cada 100 ms, cuando usamos muestras de motores con aceleración; podemos decir que con 36 coeficientes por segmento de tiempo, el lapso de entrenamiento fue mayor y no había mucha diferencia en los resultados de reconocimiento, mostrando con esto, que 26 LPCC dan mejores resultados globales.



Tomando esto en cuenta, se seleccionaron al azar doce muestras de sonido en formato WAV, provenientes de ruido no usado en la etapa de entrenamiento, y separado para este fin; tomando una RNA previamente entrenada, se alimentaron estas muestras para demostrar la precisión que logra alcanzar dicha red. Los resultados obtenidos se muestran en el Cuadro 5.



* Conclusiones

Los resultados muestran que en un motor en malas condiciones existen diferencias acústicas, ´ comparándolo con un motor afinado o en buenas condiciones. Al menos estas diferencias entre sonidos pueden ser reconocidas por una Red Neuronal con Retraso en el Tiempo, claro está, dependiendo del tipo de falla que esté afectando la eficiencia del motor. Esta información puede ayudar al especialista del motor a dar diagnósticos rápidos, precios estimados por la reparación y permitirle trabajar directamente con el problema del motor. También podemos concluir que es viable trabajar en este problema e implementar un sistema barato que pueda ser usado por fabricantes de motores o dueños de talleres.

**Almacenamiento de Datos**:

Investigacion – Objetivo – Muestra – Conceptos Que Destaco, definiciones, etc – Factores – Resultados – Conclusiones.

**Aplicaciones Similares**: ***Shazam***

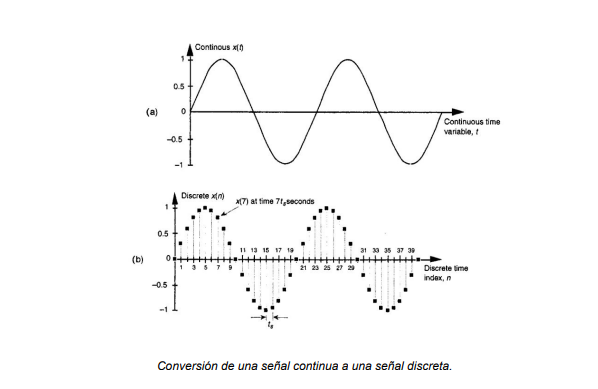
Titulo: “*Shazam: Cómo funciona el algoritmo de reconocimiento de canciones de la popular aplicación”*

Introducción: Shazam es una aplicación móvil que permite la identificación de canciones. La aplicación hace uso del micrófono que llevan incorporados la mayoría de smartphones para poder grabar una muestra de música de 10 segundos que se esté reproduciendo en el ambiente. A partir de esta breve muestra de audio, la aplicación es capaz de reconocer exactamente qué canción está sonando, brindándole al usuario información útil y precisa.

Funcionamiento: La aplicación se compone de un amplio catálogo de “huellas digitales” de audio que son almacenadas y catalogadas en una base de datos. Cuando un usuario graba con su teléfono una canción durante algunos segundos, esa grabación es procesada inmediatamente por la aplicación, creando una huella acústica (más adelante profundizaremos sobre este concepto) y enviándola a los servidores de Shazam a través de Internet. Una vez allí, se ejecuta una búsqueda de coincidencias en una base de datos que contiene millones de canciones (entre 8 y 11 millones, según algunas fuentes). Cuando existe una coincidencia, el usuario recibe en pocos segundos una respuesta con información de la pista original como el título de la canción, artista, álbum, e información adicional.

¿Como se logra este funcionamiento?

El micrófono del teléfono convierte la señal de presión de aire continua producida por la misma canción, en una señal de voltaje analógica. De aquí se toma un muestreo que luego será analizado.



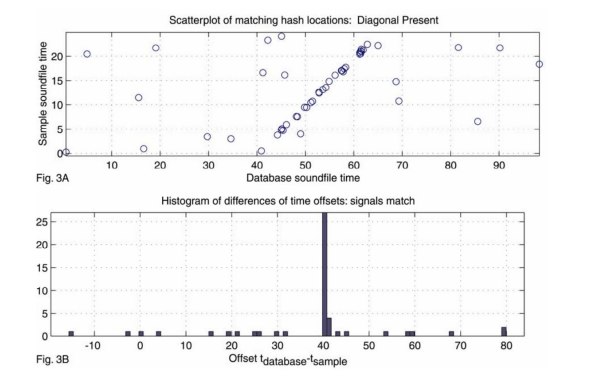
Si utilizamos esa frecuencia de muestreo, podremos reproducir la señal posteriormente a partir de las muestras tomadas.

Gracias a estas muestras se consigue a través de un largo procedimiento las denominadas “**Huellas Acústicas**” las cuales son un “resumen” digital del archivo, generado a partir de la señal de audio original. Gracias a este procedimiento, es posible identificar o localizar unívocamente un archivo de audio en una base de datos, sin necesidad de contar con información adicional.

Luego de analizar estas huellas acústicas, se pasa a la **detección de coincidencias**:

Ahora que hemos creado la huella acústica de nuestro archivo de audio, la información codificada en un conjunto de hashes con cada uno de los pares de puntos de anclaje de la grabación del usuario se envía a la base de datos de Shazam. Esta búsqueda devolverá las huellas acústicas de todas las canciones que contengan coincidencias de hash, es decir, que tengan los mismos pares de puntos de anclaje.

Si visualizamos este proceso en un diagrama de dispersión donde el eje y representa el momento en que se produce el hash en la grabación y el eje x representa el tiempo en que se produce el hash en el archivo de audio de la base de datos, podemos ver cómo se formarán los puntos coincidentes en la línea diagonal:



El desplazamiento en el tiempo se calcula restando el tiempo de ocurrencia del par de puntos de anclaje en la señal de entrada del tiempo de hash correspondiente en el archivo de audio de la base de datos. Si una cantidad significativa de hashes coincidentes tienen el mismo coeficiente de desplazamiento, se determina que esa canción es una coincidencia.

Este método de búsqueda de audio es lo suficientemente preciso para encontrar coincidencias incluso cuando la señal de entrada contiene ruido, como personas hablando, ruido de calle o incluso otras canciones sonando en simultáneo. Debido a que el número de hashes de puntos de anclaje creados por una huella acústica es mucho mayor que la cantidad de coincidencias de puntos de anclaje requeridas para devolver un resultado de búsqueda positivo, los hashes de puntos de anclaje que están enmascarados por el ruido externo no son suficientes para evitar que Shazam siga buscando coincidencias recursivamente en otros fragmentos de la señal.

**Marco Teórico**

Investigacion de todas las variables de las que voy a tratar

* Mercanica.
* Python.
* DataScience.
* Reconocimiento de sonidos.
* Reconocimiento de datos. Otros detectores.

Mecanica

Sonidos,

Espectrogramas.

Informatica,

* Redes neuronales
* Técnica: Python, Java , AudioSet
* Inteligencia Artificial
* Bases de datos
* Aplicaciones Móvil