



Universidad
Señor de Sipán

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**Identificación automática de neumonía mediante el
procesamiento digital del sonido**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
DE SISTEMAS**

Autor (es)

Bach. Jara Tucto Alexander

ORCID: <https://orcid.org/0000-0002-2297-8232>

Asesor(a)

Dr. Tuesta Monteza Victor Alexci

ORCID: <https://orcid.org/0000-0002-5913-990X>

**Línea de Investigación
Infraestructura, Tecnología y Medio Ambiente.**

Pimentel – Perú

2023

IDENTIFICACIÓN AUTOMÁTICA DE NEUMONÍA MEDIANTE EL PROCESAMIENTO DIGITAL DEL SONIDO

Aprobación del jurado

MG. MEJIA CABRERA HEBER IVAN
Presidente del Jurado de Tesis

MG. MINGUILLO RUBIO CESAR AUGUSTO
Secretario del Jurado de Tesis

DR. TUESTA MONTEZA VICTOR ALEXCI
Vocal del Jurado de Tesis




DECLARACIÓN JURADA DE ORIGINALIDAD

Quien suscribe la **DECLARACIÓN JURADA**, soy Alexander Jara Tucto del Programa de Estudios de la Escuela de Ingeniería de Sistemas de la Universidad Señor de Sipán S.A.C, declaro bajo juramento que soy autor del trabajo titulado:

“Identificación Automática de Neumonía Mediante el Procesamiento Digital del Sonido”

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Jara Tucto Alexander	DNI: 73765114	
-----------------------------	----------------------	--

Pimentel, 01 de marzo de 2023.

Dedicatoria

Dedicatorias

La presente investigación está dedicada en primer lugar a Dios por haberme guiado para culminar una de mis metas.

Dedico también esta investigación a mi abuelita Consuelo Valdivia Grossi que en paz descanse, a mi abuelito Lorenzo Jiménez puerta y mi madre María Tuco Puscan quienes son mis cimientos de mi educación.

Y también está dedicada a mi asesor Ing. Tuesta Monteza Víctor por haberme apoyado en este largo camino.

Agradecimientos

En primer lugar, agradezco a Dios por haberme guiado durante todo este recorrido de mi carrera universitaria y por haberme iluminado para concluirlo.

Agradezco a mi abuelita Consuelo Valdivia Grossi y a mi abuelito Lorenzo Jiménez Puerta por haberme guiado en cada paso de mi vida desde muy pequeño; así como también por ser mi inspiración para luchar por mis sueños.

También agradezco a mi madre María Tucto Puscan por haber sido siempre mi apoyo constante ante cualquier dificultad que se me presento y por sus consejos para superar las adversidades de la vida.

Agradezco también a mis hermanos Carlos e Itamar porque de una u otra manera me han apoyado siempre.

A mi novia Bilma Silva Rojas por su apoyo incondicional en toda mi vida universitaria; así mismo mis tíos Kilmer, Augusto, Homar, Edwin, Licett, Margot y Silvia por su apoyo en cada momento.

Y finalmente pero no menos importante a mis asesor Dr. Tuesta Monteza Víctor por impulsarme a culminar mi tema de tesis.

Resumen

En la actualidad la neumonía es la principal causa de mortalidad infantil, alcanzando anualmente a cifra 1.1 millones de fallecimiento de niños menores de 5 años. Para diagnosticar oportunamente esta enfermedad se realiza por medio de la auscultación pulmonar, empleando un estetoscopio, el cual permite percibir los sonidos respiratorios y así descubrir algún signo anormal. El objetivo de este trabajo es realizar la Identificación de manera automática la neumonía mediante el procesamiento digital del sonido, la investigación inicia con la construcción de un protocolo para la adquisición de sonidos de enfermedades respiratorias, que sirvieron para la adquisición de sonidos respiratorios, posteriormente se transformó el audio a imagen para adquirir el espectrograma de cada uno, al tener el espectrograma se realizó el procesamiento de las imágenes utilizando Keras, así mismo creamos nuestra red neuronal convolucional y comenzamos a realizar el entrenamiento con el dataset de imágenes con un valor inicial de 200 épocas. Los resultados fueron satisfactorios debido a que se obtuvo un 75%, 69% y 75% de la exactitud, precisión y sensibilidad, respectivamente. Finalmente se evaluó el método, el cual tiene un buen desempeño con respecto a la identificación automática de la neumonía.

Palabras clave – Espectrograma, Dataset ,Keras, Neumonía, red neuronal convolucional.

Abstract

Currently, pneumonia is the leading cause of infant mortality, reaching 1.1 million deaths of children under five years of age every year. In order to diagnose this disease in a timely manner, it is carried out by means of auscultation of the lungs, using a stethoscope, which allows the perception of respiratory sounds and thus discovering any abnormal signs. The objective of this work is to automatically identify pneumonia by means of digital sound processing. The research begins with the construction of a protocol for the acquisition of sounds from respiratory diseases, which were used to acquire respiratory sounds, and then the audio was transformed into an image to acquire the spectrogram of each one. The results were satisfactory because 75%, 69% and 75% of the accuracy, precision and sensitivity were obtained, respectively. Finally, the method was evaluated by developing a program, which has a good performance regarding the automatic identification of pneumonia.

Keywords - Spectrogram, Dataset ,Keras, Pneumonia, convolutional, neural, network.

Índice de contenido

I.	INTRODUCCIÓN	10
1.1.	Realidad Problemática.....	10
1.2.	Formulación del Problema.....	15
1.3.	Hipótesis.....	15
1.4.	Objetivos.....	15
1.5.	Teorías relacionadas al tema.....	16
II.	MÉTODO	24
2.1.	Tipo y Diseño de Investigación.....	24
2.2.	Variables, Operacionalización.....	24
2.3.	Población de estudio, muestra, muestreo y criterios de selección.....	25
2.4.	Técnicas e instrumentos de recolección de datos, validez y confiabilidad.....	26
2.5.	Procedimiento de análisis de datos.....	26
2.6.	Criterios éticos.....	26
III.	RESULTADOS.....	27
3.1.	Resultados.....	27
3.2.	Discusión.....	29
3.3.	Aporte de la investigación.....	30
IV.	CONCLUSIONES Y RECOMENDACIONES.....	49
4.1.	Conclusiones.....	49
4.2.	Recomendaciones.....	50
	REFERENCIAS.....	51
	ANEXOS.....	54

Índice de figuras

Figura 1 Resultados para los indicadores de la variable dependiente (exactitud, sensibilidad, precisión).....	29
Figura 2 Método Propuesto.....	30
Figura 3 Características técnicas de los equipos.	31
Figura 4 Panel acústico para el micrófono	31
Figura 5 Implementación de los equipos adquiridos	32
Figura 6 Segmentación de la pared torácica para la auscultación.....	32
Figura 7 Ejercicio de respiración y espiración	33
<i>Figura 8</i> Software para la grabación de la auscultación	33
Figura 9 Selección y segmentación por tipo de infección respiratoria	34
Figura 10 Conversión de audio a imagen por tipo de infección respiratoria	34
Figura 11 Dirección de img_data con las imágenes convertidas.....	35
Figura 12 Procesamiento de imágenes utilizando	36
Figura 13 Código fuente de ImageDataGenerator	37
Figura 14 Imagen referencial de creación y división de la data	37
Figura 15 El 80% de entrenamiento y un 20% para prueba.	38
Figura 16 Representación gráfica de la ecuación	39
Figura 17 código fuente de la red neuronal convolucional	40
Figura 18 Código fuente del entrenamiento	42
Figura 19 Código Fuente para la entrada de entrenamiento	42
Figura 20 Gráfico de exactitud	44
Figura 21 Figura de sensibilidad	46
Figura 22 Código fuente para restablecer el test_set.....	48
Figura 23 Código fuente para guardar los resultados en un CSV	49

I. INTRODUCCIÓN

1.1. Realidad Problemática.

En la actualidad la neumonía es la principal causa de mortalidad infantil, alcanzando anualmente a cifra 1.1 millones de fallecimiento de niños menores de 5 años [1]. Para diagnosticar oportunamente esta enfermedad se realiza por medio de la auscultación pulmonar, empleando un estetoscopio, el cual permite percibir los sonidos respiratorios y así descubrir algún signo anormal.

Para la auscultación se utiliza el estetoscopio, este procedimiento tiene ciertas limitaciones con la sensibilidad ya que al realizar este procedimiento el estetoscopio entra en contacto con la piel del tórax del paciente, de tal manera que se percibir los sonidos generados por los órganos del cuerpo, así como el corazón, pulmón o intestinos. Para Analizar estas diferentes fuentes de sonido se necesita tener una de la capacidad auditiva, la destreza y conocimiento del especialista para identificar los sonidos respiratorios; para confirmar un diagnostico se recurre a otros exámenes que se ayuda corroborar dicho resultado.

Uno del instrumento infaltable de un médico es el estetoscopio o fonendoscopio; este instrumento es esencial a la hora de detectar problemas cardiacos, neumonía, asma y muchas otras enfermedades del pulmón, así también se perciben sonidos de nuestro organismo. Se ha utilizado el análisis de reconocimiento de la neumonía basado en el análisis matemático, así mismo combinó con otras características tales como la coeficientes de Mel Cepstral e índice de no gaussianidad[2].Para lo cual fue necesario la construcción de una base de datos de archivos digitales con la grabación de pacientes sanos y con infecciones respiratorias; así mismo se construyó un prototipo para la adquisición de sonidos pulmonares los cuales se clasifican por: crepitaciones que es el ruido de baja intensidad similar al que se produce cuando se frota un mechón de cabello, sibilancias que son sonidos similares a un silbido continuo, roncus que son una especie de ronquido que sugiere la presencia de secreciones en los bronquios, respiración soplante o también llamados soplo tubario que el sonido es similar al que se produce al soplar por un tubo estrecho, broncofonía o pectoriloquia es la voz se escucha nítidamente, aun haciendo la auscultación pulmonar en la pared torácica, frotos pleurales son sonidos similares al roce de dos cueros, soplo pleurítico que es similar a la respiración suplante[1].

Así mismo fue necesario convertir la base de datos de audios digitales a imágenes en formato PNG para así extraer el espectrograma; que son señales de audio que consisten en eventos de sonido que se repiten con el tiempo, como fonemas en el discurso y notas de instrumentos musicales [3]

Posteriormente se realizó la implementación de una red neuronal convolucional, es una herramienta de análisis gráfico que combina la información de las características y la estructura gráfica para representar en gráficos a través de la propagación de características y agregación [4].

De este modo, se evaluó el método propuesto para identificar la neumonía mediante el desarrollo de un programa. Por tanto, esta investigación proporciona una herramienta que sirve para realizar un análisis sobre la conducta de la señal respiratoria en pacientes sanos y con neumonía.

La presente investigación tiene como objetivo desarrollar un algoritmo de clasificación para identificar la neumonía mediante el procesamiento digital del sonido. Los resultados que obtuvieron en la presente investigación no fueron los más óptimos, logrando un 75% en sensibilidad, 75% en exactitud y un 69% en precisión; debido a que la base de datos no fue en ambientes controlados.

A continuación, el análisis de 10 trabajos relevantes relacionados con la clasificación y procesamiento de sonido:

wavelet augmented cough analysis for rapid childhood pneumonia diagnosis. [2] Este trabajo de investigación su objetivo principal extraer las características de wavelet de los sonidos de tos y los combinó con otras características tales como los coeficientes de Mel Cepstral e índice de no gaussianidad. Para ello se entrenó a un clasificador de regresión logística para separar la neumonía de otras enfermedades, como referencia estándar, utilizaron el diagnóstico de médicos asistidos con resultados de laboratorio y radiológicos. Con los métodos propuestos se logró una sensibilidad y especificidad del 94% y 63%, respectivamente, en la separación de pacientes con neumonía de no neumonía.

Analysis Of Acoustic Parameters From Respiratory Signal In Copd And Pneumonia Patients [5] El estudio de esta investigación fue identificar y clasifica la neumonía en

varias etapas como nivel, moderada y severa usando la matriz de ocurrencia del nivel de gris (GLCM) características de la textura. Para el examen de pulmón se realizó utilizando Philips MX8000 IDT 16 slice CT scan para todos los temas y guardado en Formato DICOM. Las imágenes obtenidas fueron sometidas a la mediana de filtrado para la eliminación de ruido; la normalización se hizo utilizando estiramiento de contraste. Posteriormente GLCM fue aplicado para extraer varias características de la textura a saber contraste, energía, probabilidad máxima, varianza, media, sesgo, entropía, desviación estándar, autocorrelación, mediana, modo, prominencia de grupo, sombra de grupo, Homogeneidad y curtosis. Como resultados los clasificadores de exactitud obtuvieron 96% red neuronal y 86% Support vector machine.

Design Of A Clinical Decision Support Model For Predicting [6]

En esta investigación tienen como fin de identificar a los componentes para predecir la readmisión y diseñar un sistema de apoyo a la decisión clínica (CDSS) para predecir si un paciente lo hará ser readmitido por neumonía dentro de los 30 días posteriores al alta. Se recogieron factores clínicos y factores de utilización de la asistencia sanitaria, las variables seleccionadas se aplican para el diseño de CDSS utilizando el RBF- SVM. El modelo predictivo se construyó utilizando el RBF-SVM con 6 variables significativas y todas las 20 variables, respectivamente. La precisión predictiva para dos modelos diferentes logró 83.85% y 82.24% respectivamente, el modelo puede ser efectivo en identificación de pacientes con neumonía con alto riesgo de reingreso.

Finalmente lograron producir con éxito un modelo para identificar, antes de altas hospitalarias, neumonías con alto riesgo de reingreso. Si bien es cierto se necesitará mejorar ya que el máximo de capacidad predictiva sigue siendo inadecuada debido al estado socioeconómico y salud de los pacientes.

Research On Diagnostic Models Of Pneumonia Syndromes In Tcm Based On Fuzzy-Neural Net [7]

Esta investigación tiene como el objetivo de es el estudio de los resultados en la minería de datos de la neumonía con el criterio de selección, 1058 piezas de datos clínicos de los pacientes que se recogieron por métodos epidemiológicos clínicos.

Se utilizaron los modelos de redes neuronales; todo esto se construyó sobre una base de la red kohonen dinámica y su fiabilidad se probó con el dato de Fisher- Iris, luego, con la ayuda de los modelos se estudiaron los datos clínicos y el criterio de diagnóstico

para los síndromes de neumonía común se obtuvo de acuerdo con las medicina tradicional china y teorías básicas. Finalmente, la fiabilidad fue probada por el chequeo de datos y la tasa de diagnóstico de coincidente alcanzó el 86% en comparación con el criterio de diagnóstico y los datos de diagnóstico originales.

Spectral-Based Pneumonia Detection Tool Using Ultrasound Data From Pediatric Populations [8] En el presente trabajo, mencionan que para el diagnóstico de neumonía se realiza por criterios clínicos como la ecografía pulmonar; sin embargo, la interpretación es de acuerdo al criterio que cada doctor y requiere adiestramiento adecuado. Por lo que a través de su investigación ellos buscan realizar un algoritmo basado en el ancho de banda ya que hay un cambio descendente sobre la profundidad de la radiofrecuencia del ultrasonido.

Finalmente se construyó una función descriptora concatenando todos los valores ajustados de las líneas de RF para cada trama respectivamente. Cada función del descriptor fue luego un umbral para diferenciar entre regiones sanas y neumónicas. Se encontró que el umbral óptimo era de 0,46 MHz / cm y fue seleccionado basado en una característica del operador del receptor (ROC) análisis de curvas. Los resultados revelaron una tasa de precisión más alta, con más del 90 % de la muestra.

Analysis Of Acoustic Parameters From Respiratory Signal In Copd And Pneumonia Patients [8] Investigación tienen como propósito la extracción de características acústicas utilizando el software PRAAT v6.0.43 de la señal respiratoria, para evaluar estas características, se calculan y presentan la media y la desviación estándar. Dichos autores previos a su investigación realizaron un análisis estadístico para muchos parámetros, tales como la relación armónica a ruido (HNR), se aplicó la perturbación de tono y amplitud (jitter y shimmer) para compararlos en diferentes casos de la señal respiratoria.

Evaluation Of Classifiers To A Childhood Pneumonia Computer- Aided Diagnosis System [9] El tema que desarrollaron un sistema de diagnóstico para la detección de neumonía en bebés con imágenes radiográficas, con el objetivo de mejorar la precisión y robustez, implementando y compararon cinco clasificadores contemporáneos de aprendizaje automático, Naive Bayes, Vecino K-Nearest (KNN), Support Vector Machines (SVM), Perceptrón de capas múltiples (MLP) y Árbol de decisión, combinados con algoritmos de reducción de tres dimensiones: avance secuencial

Selección (SFS), Análisis de componentes principales (PCA) y Kernel Análisis de componentes principales (KPCA). Como resultados actuales Obtuvieron que el clasificador Naive Bayes combinado con KPCA produce los mejores resultados globales.

Analysis Of Cough Sound For Pneumonia Detection Using Wavelet Transform And Statistical Parameters [10] Los autores buscan realizar una herramienta sencilla para el diagnóstico de neumonía utilizando el análisis del sonido de la tos de paciente con neumonía. Ellos recogieron el sonido de la tos para luego realizar la descomposición wavelet del sonido y finalmente encuentran los parámetros estadísticos para dividir la tos en neumonía o no neumonía. Para lo cual aplicaron los siguientes pasos: FFT de la señal da la frecuencia en rango requerido, luego se calculan los parámetros estadísticos para segmentos de tos originales y también para coeficientes CWT. Finalmente obtiene una herramienta que es un método de prueba de detección de bajo costo, sin contacto, no invasivo para la detección de neumonía en campamentos de salud rurales.

The Development Of An Online Pneumonia Risk Prediction System[11] Esta investigación mencionan que existen modelos que se centrarse en la predicción de la mortalidad en pacientes hospitalizados, predicción del riesgo en el paciente con ictus, predicción del riesgo del ingreso en la UCI en pacientes con neumonía, y el riesgo en adultos mayores. Al identificar los factores de riesgo que contribuyen a Neumonía, se puede hacer prevención temprana para mejorar la salud. Este estudio se realizó para diseñar nuevos algoritmos para sistema de predicción de riesgo de neumonía basado en la vía de la infección a través de ocho sistemas del cuerpo humano.

Logrando desarrollaron un nuevo sistema de predicción de riesgos para detección de riesgo de neumonía. El sistema de predicción de riesgos propone con nuevos factores nombrados como factor HEL, siendo este el principal factor que contribuye a detectar el mayor riesgo de desarrollar neumonía. Obtuvieron como resultado de la prueba de precisión de una muestra modelo que el 73% preciso en la predicción de neumonía.

Cough Sound Analysis For Pneumonia And Asthma Classification In Pediatric Population[12] En este estudio, proponen un análisis del sonido de la tos, este método basado en diferenciar neumonía de asma. Sin embargo, se intentó usar el sonido de la tos para separar la neumonía y Asma en población pediátrica. Al desarrollar Nuestro

método se extrajo el sonido y las características como los coeficientes cepstrales de frecuencia de Mel, puntaje de gaussioidad, entropía de Shannon; todo esto se utilizó para desarrollar clasificadores de redes neuronales artificiales y finalmente se usó la técnica de validación Leave One Out en dieciocho temas, y así se logró sensibilidad, especificidad y Kappa del 89%, 100%, y 0,89 respectivamente. Los resultados muestran el potencial de Nuestro método para ser desarrollado como herramienta para diferenciar la neumonía por asma en zonas remotas.

Esta investigación está alineada: infraestructura, tecnología y medio ambiente, ciencias de la computación, procesamiento de señales de la Universidad Señor de Sipán. La importancia de esta investigación es contribuir al conocimiento científico de la Inteligencia Artificial, se evaluará el algoritmo más efectivo para el reconocimiento del sonido. Estos resultados ayudarán a otros investigadores a abordar problemas más específicos; así como una contribución social como se desarrollará un prototipo para la captación de sonido automatizada para personas con neumonía. Es técnicamente factible porque existen diferentes algoritmos en el mundo para hacer esta investigación.

1.2. Formulación del Problema.

¿Cómo identificar de manera automática la neumonía?

1.3. Hipótesis.

Utilizando algoritmo de Redes Neuronales se logrará identificar de manera automática la neumonía

1.4. Objetivos.

1.4.1. Objetivo general.

Identificación de manera automática de neumonía mediante el procesamiento digital del sonido.

1.4.2. Objetivos específicos.

- a) Construcción de un protocolo de adquisición de sonidos de la respiración de pacientes sanos y con infecciones respiratorias.
- b) Construir una base de datos de archivos digitales con la grabación de pacientes sanos y con infecciones respiratorias.
- c) Conversión de la base de datos de audio digitales a imágenes en formato PNG para extraer el espectrograma.
- d) Implementar algoritmos de clasificación para detectar la neumonía.
- e) Evaluar el método propuesto para identificar la neumonía.

1.5. Teorías relacionadas al tema.

En esta investigación se presentan teorías relacionadas con el tema.

1.5.1. El sonido

[13] El sonido es una combinación de varias frecuencias. Es por ello, que el sonido que percibimos no es una frecuencia limpia, si no es una mezcla de frecuencias bajas y frecuencias más altas.

El sonido de la voz y ruidos respiratorios en la combinación muy compleja de diversas frecuencias, lo que hace que sea fácil de modificar por varios factores patológicos.

1.5.1.1. Frecuencia

Una persona humana puede percibir 16 a 16.000 Hz. eso corresponde al número de ondas por unidad de tiempo. Es por ello que los sonidos respiratorios están entre las frecuencias que percibe el oído huma 70 a 4000 Hz.[13]

Para identificar la frecuencia respiratoria se calcula con el número de repeticiones de ciclos ventilatorios en transcurso de 1 minuto. El valor es variable ya que los adultos están entre 16 a 22 repeticiones por minuto, en cambio los niños se encuentran entre 20 a 30, y niñas de 30 a 50 repetidores por minuto [14]. Con estos resultados podemos comparas alguna anomalía presente.

1.5.1.2. Intensidad

La intensidad en una onda sonora de amplitud, representa a la energía que lleva el sonido. Una persona nos llega a percibir la longitud de onda fácilmente, cuando la frecuencia es de 1,000 a 5000 Hz.

Para manifestar las intensidades sonoras se utiliza una escala en decibelios (dB) que está dividida en potencias de diez, tal como se aprecia en la siguiente ecuación de conversión:

$$S=10\log I/I_0$$

$I_0 = 10^{-12}$ W/m² y su nivel está representado de 0 decibelios (dB) y el umbral del dolor afecta a la intensidad de 120dB o 1 W/m².

Aunque la intensidad acústica de diez dB en diez veces a su energía y una intensidad de 0 dB entonces si tenemos una intensidad de 20 dB es equivalente a 100 veces mayor de energía.

1.5.2. Procesamiento de Señales

a) Análisis de Fourier

Utiliza una técnica de parametrización donde las características espectrales de la señal auditable se derivan del análisis de Fourier, se describe mediante la siguiente ecuación:

$$S_x(t, f) = \int_{-\infty}^{\infty} x(\tau + t) \cdot w(\tau) \cdot e^{-j2\pi f\tau} \cdot d\tau$$

Donde $x()$ es la señal de audio y $w()$ representada la función de análisis. De tal manera que representa un análisis local de la señal aplicando una ventana $w()$ a la señal, en el tiempo "t", analizada en todas las frecuencias consideradas "f".

b) Limitaciones del análisis de Fourier

Transformada de Fourier (TF) es usada para el análisis de señales (AS) y el procesamiento de datos, ya que tiene un alto índice de solución con respecto al AS donde dicho espectro difiere con respecto al tiempo (T) de las señales no estacionarias.

[11] Con la función f al descomponer una señal con respecto al tiempo, se puede decir que la función de base Fourier perdura infinito en el T en los espacios de la frecuencia.

Fourier identifica una frecuencia que no brinda información con respecto al T de las propiedades espectrales de la señal. No se logra identificar de forma correcta las señales estacionarias debido a que las señales finitas tanto como las de inicio y las de fin presentan una aparición de una anomalía de señal transitoria.

En el caso de señales transitorias y no estacionarias, se suele utilizar la ventana TF.

b) Transformada de Fourier con ventana

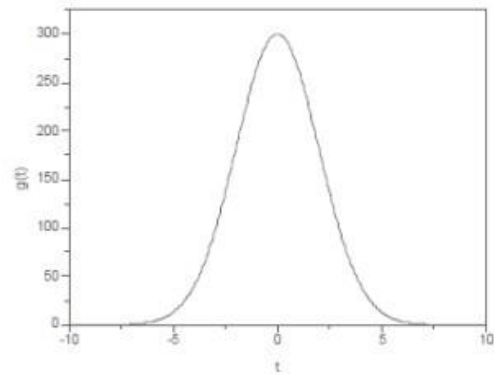
Para analizar una señal no estacionaria, se debe realizar un análisis de espectro de tiempo. La señal fija se divide en segmentos de tiempo, donde la señal se considerada cuasi – estacionaria y se aplica Fourier a cada segmento de señal.

La transformada de Fourier de tiempo corto nace en 1940 por Gabor, también conocida como ventana deslizante:

$$S_f(\omega, \tau) = \int f(t) g^*(t - \tau) \exp(-i\omega t) dt$$

(M. Vetterli, 1995) Donde $g(t)$ es una ventana deslizante, la cual tiene un ancho fijo y cambia a lo largo de su eje x por un factor es por ello que a la función Gaussiana con la función ventana $g(t)$ y así se demuestra que Fourier es una ventana Gaussiana.

$$g(t) = \frac{1}{s} \exp\left(-\frac{\pi t^2}{s^2}\right)$$



La función definida:

Con la Transformada de Fourier con Ventana no permite identificar con mayor eficiencia donde hay la aparición de anomalía en la señal. Esto solo es reflejado en el tiempo (t) que se produce la singularidad, todo esto depende al ancho que se elige en la dicha función ventana.

Dichos resultados no serán resueltos ya que aparecen muy cercanos, esto nos limita a poder diferenciar el comportamiento de la amplitud de la ventana.

Un instrumento matemático para desarrollar muchos problemas es la transformada de wavelet.

[15] Esta transformada en identificar fenómenos transitorios y de alta frecuencia con respecto a Fourier con ventana; una vez identificado el tamaño de dicha ventana, el resto de frecuencias serán analizadas de la misma manera con tiempo y frecuencia, muy diferente a Wavelet que adapta la frecuencia de la ventana. Con las imágenes Fourier es una función $F(x)$ de un soporte finito que este entre $[-\infty, +\infty]$. Después se inserta a un algoritmo de análisis de $F(x)$, esto hace una pérdida de información al hacer la antitransformación del intervalo finito. Wavelet por otra parte está en un intervalo finito eso hace que no exista una pérdida de información al ejecutar la antitransformación.

c) Híbrida Cepstrum

Tiene su origen en la palabra en latín SPECTRUM, cuya definición en si misma denota una condición la cual no puede ser limitada, debido a que podrá albergar una amplia gama de valores, datos que precisamente serán

el objeto de estudio en este trabajo. Entonces, la palabra CEPSTRUM, es el resultado de invertir las cuatro (4) primeras letras de ésta en latín, que va desde su inicio SPECTRUM=>CEPSTRUM.

Ahora bien, el Cepstrum como tal será un instrumento que permitirá evaluar las muestras de los sonidos captados, dando una data estructurada de esos sonidos y/o señales de voz, que derivará en la Transformada de Fourier del espectro logarítmico de esa data. En ese sentido se observa que hay dos tipos de Cepstrum, cuya función logarítmica estará dispuesta para valores complejos o reales, a saber: Complejo que permitirá reconstruir la señal y uno Real que lo impide por disipar la data obtenida.

Para esta investigación se pretende obtener los datos a partir del análisis del Cepstrum de cada sonido captado, aplicando una ecuación, tal como se indicará más adelante, lo cual permitirá entender el cálculo del Cepstrum, el cual debe tener una secuencia estable en $x[n]$, donde Z esta transformada en coordenadas polares, en esta ecuación:

$$X(z) = |X(z)| \cdot e^{j\angle X(z)}$$

Donde $|X(z)|$ y $\angle X(z)$ son la magnitud y el ángulo del cambio en Z de $x[n]$ respectivamente.

La señal $x[n]$ es estable, en la región de convergencia de $X(z)$ que consiste en el círculo unitario y existe un Fourier de $x[n]$ e igual a $X(e^{j\omega})$.

El Cepstrum complejo correspondiente a $x[n]$ se define como una secuencia estable $\hat{X}[n]$, la Z transformada se puede definir:

$$\hat{X}(z) = \log[X(z)]$$

Para $\hat{X}[n]$ se encuentre estable, a la región de convergencia tiene que integrar el círculo unitario, por lo que el Cepstrum complejo se resuelve utilizando la transformada inversa de Fourier, como se puede observar:

$$\hat{x}[n] = \frac{1}{2\pi} \cdot \int_{-\pi}^{\pi} \log[X(e^{j\omega})] \cdot e^{j\omega n} \cdot d\omega$$

La diferencia con el Cepstrum complejo, el $C_x[n]$ o Cepstrum real de una señal, es la transformada inversa de Fourier del log de la magnitud de Fourier, así como se observa:

$$c_x[n] = \frac{1}{2\pi} \cdot \int_{-\pi}^{\pi} \log|X(e^{j\omega})| \cdot e^{j\omega n} \cdot d\omega$$

[16] El Cepstrum real es utilizado en diversas aplicaciones, ya que es independiente de la fase de $X(e^{j\omega})$ ya que es más sencillo al resolver el Cepstrum complejo, no obstante $x[n]$ no se podrá restablecerse en $c_x[n]$.

1.5.3. Lenguajes de programación

En informática, el mundo se atribuye a las matemáticas, los matemáticos programadores que escribieron sus programas y realizaron sus pruebas con palabras matemáticas, es decir, usan símbolos matemáticos y eso es donde nacieron los lenguajes de programación y cada uno es muy útil en ciertos contextos. La presente investigación tiene como propósito extenderse sobre 3 idiomas diferentes con estilos de programación no trivial, destacando sus ventajas y características. [17]

1.5.3.1. Python

Python es otro lenguaje orientado a objetos como C++ y Java. A diferencia de C++, que es un lenguaje interpretado, Python se ejecuta en diferentes plataformas como Mac OS X de Apple, Linux y Microsoft Windows.

Así es como los programadores de todo el mundo se interesan en python desarrollando muchas bibliotecas y a su vez muchos programas, esto hace que python sea muy popular debido a la portabilidad y la capacidad de las bibliotecas. La comparación de python conduce a una filosofía entre los

desarrolladores que piensan que usar el lenguaje conducirá al éxito final, existen múltiples versiones creadas a lo largo del tiempo, que veremos a continuación.

1.5.4. Aprendizaje Automático

Si alguna vez ha escrito un programa, estará familiarizado con el diagrama que se muestra en la siguiente figura. Escribes un programa, alimentas algunos datos y obtienes tu salida. Por ejemplo, puede escribir un programa para realizar algunas tareas de contabilidad para su negocio. En este caso, los datos recopilados incluirían sus registros de ventas, sus listas de inventario, etc. Luego, el programa tomaría los datos y calcular sus ganancias o pérdidas en función de sus registros de ventas. Quizás también pueda producir algunos gráficos bonitos y fantasiosos que muestren su desempeño en ventas. En este caso, el resultado es el estado de pérdidas y ganancias, así como otros gráficos.

a) Soporte de máquinas de vectores

Los métodos SVM utilizan condiciones lineales para separar las clases entre sí. La idea es usar una condición lineal que separe las dos clases entre sí lo mejor posible. [18].

En tal caso, la condición de división en el caso multivariante también se puede utilizar como condición independiente para la clasificación. Este, un clasificador SVM, se puede considerar como un árbol de decisión de un solo nivel, siempre que la división multivariada se elija con mucho cuidado. Claramente, dado que la efectividad del enfoque depende solo de un solo plano de separación, es crítico definir esta separación cuidadosamente. (Aggarwal et al., 2014).

Se puede usar tanto para desafíos de clasificación como de regresión. SVM modela la situación creando un espacio de características, que es un espacio vectorial de dimensión finita. El principio básico de SVM es separar el espacio de características en dos clases encontrando un hiperplano que distinga bien las dos clases. Los métodos SVM son poderoso para estudiar la interacción entre variantes comunes y variantes raras dentro de una

pequeña familia de muestras. Chen et al han desarrollado un método eficiente y efectivo para el clasificador SVM que utiliza un algoritmo genético paralelo de grano grueso CGPGA que puede usarse ampliamente para seleccionar conjuntamente el subconjunto de características y optimizar los parámetros para SVM en muchas aplicaciones prácticas de la ciencia biológica. (Machine Learning and IoT: A Biological Perspective - Google Libros, n.d.).

[18] En general, los datos de dimensión d se supone que el hiperplano dividido tiene la forma $\bar{W} \cdot \bar{X} + b = 0$. Aquí W es un vector de dimensión d que representa los coeficientes del hiperplano disociado, y constante. Sin pérdida de generalidad, se puede suponer (a través de la escala adecuada de los coeficientes) que dos vectores soporte simétricos de forma $\bar{W} \cdot \bar{X} + b = 1$ y $\bar{W} \cdot \bar{X} + b = -1$. Los coeficientes W y b deben aprenderse de los datos de entrenamiento D para maximizar la amplitud de separación entre estos dos hiperplanos paralelos. A partir del álgebra lineal básica, se puede demostrar que la distancia entre estos dos hiperplanos es $2 / \|W\|$. Maximizar esta función objetivo es equivalente a minimizar $\|W\|^2 / 2$. Las restricciones del problema están determinadas por el hecho de que los puntos de datos de entrenamiento para cada clase se encuentran en un lado del vector. Por lo tanto, estas limitaciones son las siguientes:

$$\bar{W} \cdot \bar{X} + b = 1 \geq +1 \quad \forall i: y_i = +1$$

$$\bar{W} \cdot \bar{X} + b = -1 \geq -1 \quad \forall i: y_i = -1$$

b) K-NN

(scikit-learn, 2019) Proporciona funcionalidad para métodos de aprendizaje basados en vecinos no supervisados y supervisados. El vecino más cercano es la base de muchos otros métodos de aprendizaje, especialmente el aprendizaje múltiple y la síntesis del espectro.

El principio detrás de los métodos del vecino más cercano es encontrar un número predeterminado de muestras de entrenamiento la distancia más cercana al nuevo punto y predecir la etiqueta a partir de ellas. El número de muestras puede ser una constante definida por el usuario (k-aprendizaje del vecino más cercano), o puede variar según la densidad del punto local

(aprendizaje del vecino basado en el radio). En general la distancia es cualquier métrica: la distancia euclidiana estándar es la opción más común. Los métodos basados en vecinos se denominan métodos de aprendizaje automático no generalizados, porque simplemente "recuerdan" todos sus datos de entrenamiento (que terminan convirtiéndose en una estructura de indexación rápida, por ejemplo, como rodamiento de árboles o ejes KD).

A pesar de su simplicidad, los vecinos más cercanos tienen éxito debido a la gran cantidad de problemas de clasificación y regresión, que puede resolver rápidamente. Como método sin parámetros, siempre tiene éxito en problemas de clasificación donde el límite de decisión es irregular.

1.5.5. Anaconda

La distribución de código abierto Anaconda Distribution es la forma más fácil de realizar ciencia de datos Python / R y aprendizaje automático en Linux, Windows y Mac OS X. Con más de 15 millones de usuarios en todo el mundo, es el estándar de la industria para el desarrollo, prueba y capacitación en una sola máquina, que permite a los científicos de datos individuales [20]

II. MÉTODO

2.1. Tipo y Diseño de Investigación.

2.1.1. Tipo de investigación

La presente investigación es de tipo cuantitativa, ya que los indicadores establecidos brindaron información para medir numéricamente los resultados, ya que de esta forma se podrá dar estadísticas de porcentaje.

2.1.2. Diseño de investigación

Por tipo de investigación, el diseño utilizado es Cuasi Experimental, porque genera interrogantes a través de Hipótesis.

2.2. Variables, Operacionalización.

2.2.1. Variable

2.2.1.1. Variable Independiente

Algoritmos de clasificación

2.2.1.2. Variable Dependiente

Identificación de la neumonía

2.2.2. Operacionalización de Variable

Tabla 1 Operacionalización de variables

Variable	Dimensión	Indicadores	Técnicas de Recolección
Algoritmos de Clasificación	Eficiencia	Precisión (Prec) $\text{Prec} = \frac{\sum VP}{\sum \text{Total clasificadores positivo}}$	Observación Científica Ficha de Observación
		Sensibilidad (Sen) $\text{Sen} = \frac{\sum VP}{\sum \text{Total Positivos}}$	
		Exactitud (Exa) $\text{Exa} = \frac{\sum VP + \sum VN}{\text{Total}}$	
Identificación de la neumonía	Rendimiento	Tiempo Memoria Procesador	

Nota. Fuente: Elaboración propia.

2.3. Población de estudio, muestra, muestreo y criterios de selección.

2.3.1. Población

El total incluye 920 grabaciones anotadas con diferentes duraciones: de 10 a 20 segundos. Estas grabaciones se consideraron de 126 pacientes. Hay un total de

5.5 horas de registros que contienen 6898 ciclos de respiración: 1864 contienen crepitaciones, 886 contienen sibilancias y 506 contienen crepitaciones y sibilancias.

2.3.2. Muestra

La muestra determinada por conveniencia de la investigación será 920 grabaciones anotadas.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

Observación Científica: A través de esta técnica de observación científica, se utilizará durante el desarrollo de la tesis para verificar los resultados de los métodos de identificación automática de la neumonía propuestos.

2.5. Procedimiento de análisis de datos.

Se va a utilizar la matriz de confusión en función a la evaluación del desempeño de un algoritmo de clasificación, ya que nos dará una mejor idea de cómo se está desarrollando nuestro algoritmo de clasificación seleccionado, a partir de un conteo tanto de aciertos como de errores, basándose en la fórmula de la precisión en forma que se esté desarrollando de forma correctamente.

Loa verdaderos positivos. Este es el número de positivos que se han clasificado correctamente como modelo positivo (VP).

$$Precision = \frac{VP}{total\ clasificados\ positivos}$$

2.6. Criterios éticos.

En este sentido deben tomarse en cuenta criterios importantes para esta investigación serán dos (2) según se indica a continuación:

Confidencialidad: En estricto sensu, se entiende ésta como la limitación en el acceso a determinada información que pueden tener un número reducido de personas, sobre hechos o situaciones fácticas, las cuales requerirán de un manejo discreto, so pena de sanciones en caso de incurrir en una develación de ésta sin la autorización respectiva por parte de quien, debidamente demostrado ostente su autoría u origen.

En este orden de ideas, el manejo de la confidencialidad en el caso de los datos que se obtienen de pacientes vistos por personal tanto médico como no médico, tales como informantes, técnicos, enfermeras, etc.; se rigen por Códigos de Ética, cuyo principal objetivo es la protección y cuidado de la integridad de los pacientes, su identificación, pudiendo utilizarse un pseudónimo para mantenerse incognito.

Derechos de Autor: para esta investigación, se hace uso de un sistema referencial y de citas, los cuales serán señalados con los autores, según la norma vigente a tal efecto, de esta manera reconocer su trabajo intelectual. Los datos presentados serán reales.

Derechos de Autor: Se referenciarán y citarán los materiales utilizados para el desarrollo del estudio.

Criterios de Rigor Científico.

Esta investigación se llevará a cabo en base a una evaluación científica, asegurando la calidad de la investigación.

Esta es una forma de monitorear la coherencia metodológica durante el desarrollo de la investigación, mediante el muestreo de datos que se extraerán al azar.

III. RESULTADOS.

3.1. Resultados.

El resultado de la investigación propuesta me arrojó una matriz de confusión que se evidencia en la tabla

Tabla 2 *Matriz de confusión de resultados (Elaboración Propia)*

	Sanos (0)	Neumonía (1)
--	------------------	---------------------

Clasifico Sanos (0)	VN = 12	FP=4
Clasifico Neumonía (1)	FN=3	VP=9
	15	12

Nota. Fuente: Elaboración Propia.

a) Precisión = $VP / (FP + VP)$

$$= 9 / (4 + 9)$$

$$= 0.69 = 69\%$$

b) Sensibilidad = $VP / (VP + FN)$

$$= (9) / (9+3)$$

$$= 0.75 = 75\%$$

c) Exactitud = $(VP+VN) / (VP +VN+ FP + FN)$

$$= (9+12) / (9 +12+ 4 + 3)$$

$$= 0.75 = 75\%$$

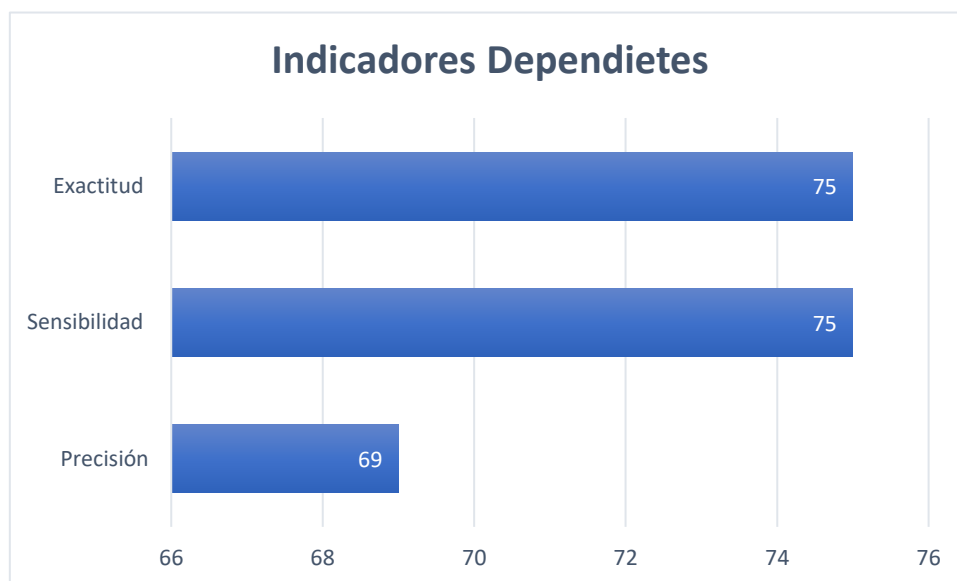
d) Tiempo = 2 minutos

e) Memoria = 1 GB

f) Procesador = 3.1 GZ.

Variables dependientes:

Figura 1 Resultados para los indicadores de la variable dependiente (exactitud, sensibilidad, precisión).



Nota. Fuente: Elaboración propia.

- En el indicador de exactitud obtengo un 75%, lo que significa que el modelo de red convolucional propuesto tiene problemas para detectar los espectrogramas de las enfermedades porque el porcentaje es inferior al 90%.
- En el indicador de sensibilidad obtuve un 75%, es decir un 75% dieron positivos en personas con neumonía, 25% de imágenes de personas sanas.
- En el indicador de precisión obtuve un 69%, significa que no fueron correctamente clasificadas con respecto a los otros indicadores

3.2. Discusión.

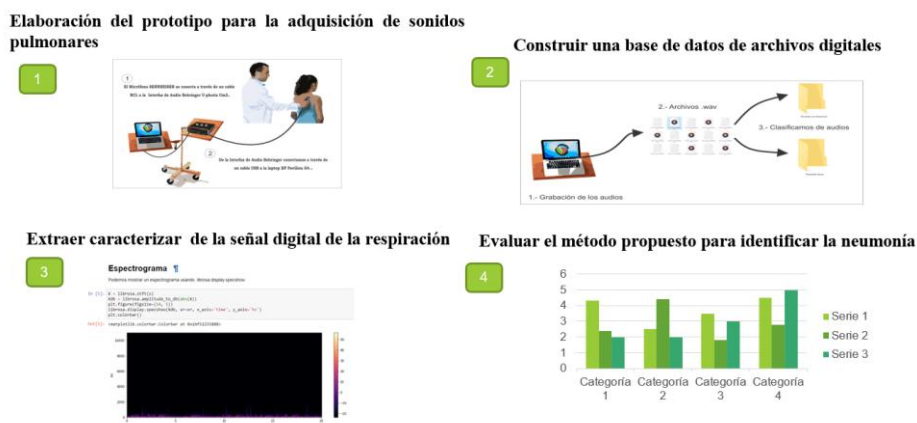
En el trabajo de [8] buscaron diagnosticar la neumonía mediante algoritmos basados en el espectrograma, los resultados revelaron una tasa de precisión más alta, con más del 90 % de la muestra. Cada función del descriptor fue luego un umbral para diferenciar entre regiones sanas y neumónicas. Se encontró que el umbral óptimo era de 0,46 MHz / cm y fue seleccionado basado en una característica del operador del receptor (ROC) análisis de curvas. Los resultados que obtuvieron fueron de un 94%, 90% y 94% con respecto a la exactitud, precisión y sensibilidad; en comparación a la presente investigación los resultados fueron adversos logrando un

75% en sensibilidad, 75% en exactitud y un 69% en precisión; debido a que la base de datos no fue en ambientes controlados.

3.3. Aporte de la investigación.

En esta investigación, el método propuesto fue estructurado en 4 pasos. Inició con la construcción de un protocolo de adquisición de sonidos de las respiraciones de pacientes sanos y con infecciones respiratorias. En la segunda parte se construyó una base de datos de archivos digitales con las grabaciones de pacientes sanos y con infecciones respiratorias. En el tercer paso se convirtió de la base de datos de audios digitales a imágenes en formato PNG con la finalidad de extraer el espectrograma, para así procesar en una red neuronal convolucional. En el cuarto se evaluó el método propuesto para lograr identificar la neumonía.

Figura2 Método Propuesto



Nota. Fuente: Elaboración propia.

3.3.1. Construcción de un Protocolo de Adquisición de Sonido de la Respiración de Pacientes Sanos y con infecciones respiratorias

Para la adquisición de sonidos respiratorios elabore un prototipo en el cual se utilizó micrófono sennheiser e945, Interfaz de audio Behringer U-pho los cuales se conectan a una laptop hp pavilion g4-1386la, para mejorar la calidad del sonido se fabricó un panel acústico para el micrófono de una

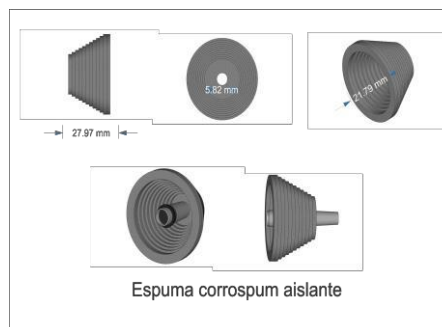
espuma corrospum aislante que permitirá la eliminación de ruido y/o sonidos externos; de esta forma facilitara que se obtengan grabaciones limpias y sin ruido, dichas grabaciones se obtuvieron con ayuda del software Reape.

Figura 3 Características técnicas de los equipos.



Nota. Los equipos con sus características técnicas que utilizamos para en prototipo se pueden observar en la figura 3 - Elaboración propia.

Figura 4 Panel acústico para el micrófono



Nota. Fuente: Elaboración Propia.

Con todos los componentes requeridos se implementó dicho prototipo que facilitaría las grabaciones a pacientes sanos y con infecciones respiratorias, como se observa en la figura 5.

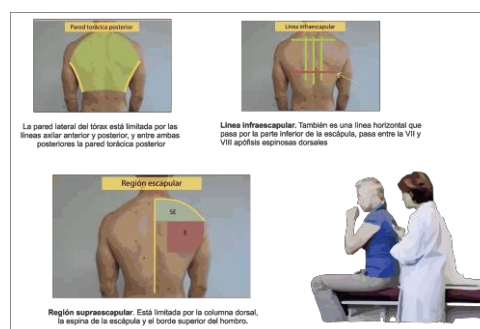
Figura 5 Implementación de los equipos adquiridos



Nota. Figuras: Elaboración Propia.

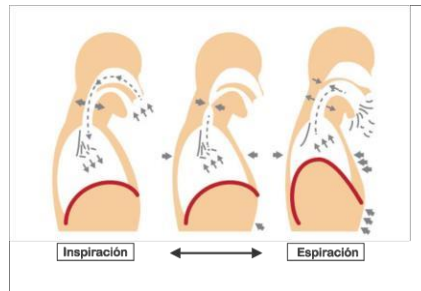
Una vez implementado el prototipo de adquisición de sonidos, se requiere la colaboración del paciente; donde se explica un protocolo referente al ejercicio del tracto respiratorio (inspiración/expiración). Para ello se debe realizar la identificación de la pared torácica posterior, a continuación, trazamos una línea infra escapular referencial para poder identificar la región a auscultar que observamos en la figura 6.

Figura 6 Segmentación de la pared torácica para la auscultación



Nota. Fuente: Elaboración Propia.

Figura 7 Ejercicio de respiración y espiración



Nota. Fuente: Elaboración Propia

Después de la identificación de la pared torácica se procede a la grabación de sonidos con ayuda del software Reape, lo cual se visualiza en la figura 8.

Figura 8 Software para la grabación de la auscultación

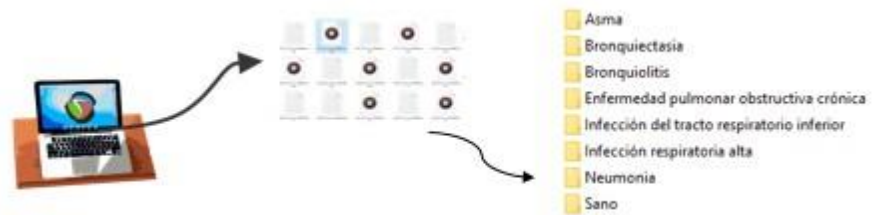


Nota. Fuente: Elaboración Propia

3.3.2. Construcción de una base de datos de archivos digitales con la grabación de pacientes sanos y con infecciones respiratorias.

Los audios adquiridos se clasificaron en carpetas con respecto al tipo de enfermedad respiratoria. Para así construir nuestra base de datos de audio digitales.

Figura 9 Selección y segmentación por tipo de infección respiratoria



Nota. Fuente: Elaboración Propia.

3.3.3. Conversión de la base de datos de audios digitales a imágenes en formato PNG para extraer el espectrograma.

Teniendo clasificados nuestros audios en carpetas con respecto a las enfermedades respiratorias, se procede a la conversión a imagen en formato PNG. Dichas imágenes se clasificaron con respecto a su origen, para así extraer el espectrograma de cada audio posteriormente se implementó en librería librosa Python.

Figura 10 Conversión de audio a imagen por tipo de infección respiratoria

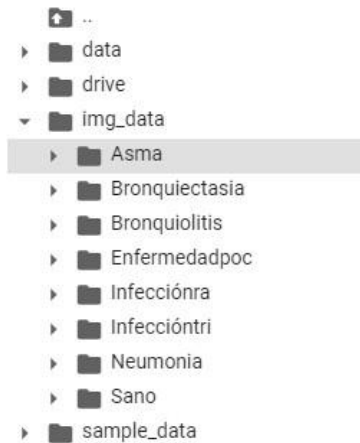
```
[ ] genres = 'Asma Bronquiectasia Bronquiolitis Enfermedadpoc Infeccióntri Infecciónra Neumonía Sano'.split()
for g in genres:
    pathlib.Path(f'img_data/{g}').mkdir(parents=True, exist_ok=True)
    for filename in os.listdir(f'./drive/My Drive/BD_sonidos/{g}'):
        songname = f'./drive/My Drive/BD_sonidos/{g}/{filename}'
        y, sr = librosa.load(songname, mono=True, duration=5)
        print(y.shape)
        plt.specgram(y, NFFT=2048, Fs=2, Fc=0, noverlap=128, sides='default', mode='default', scale='dB');
        plt.axis('off');
        plt.savefig(f'img_data/{g}/{filename[:-3].replace(".", "")}.png')
        plt.clf()

(110250,)
(110250,)
(110250,)
```

Nota. Fuente: Elaboración Propia.

Al implementar nuestro código nos crea un directorio `img_data` que contiene todas las imágenes clasificadas por tipo de sonido con respecto a su origen de cada audio.

Figura 11 Dirección de `img_data` con las imágenes convertidas



Nota. Fuente: Elaboración Propia.

Los parámetros de los componentes sonoros de los sonidos reales suelen cambiar con el tiempo en lugar de permanecer estables. Por lo tanto, para analizar fragmentos de sonido, a menudo nos interesa obtener gráficos que muestren los cambios en los valores de amplitud y frecuencia de cada componente a lo largo de la duración del sonido. Esta representación se suele denominar espectrograma.

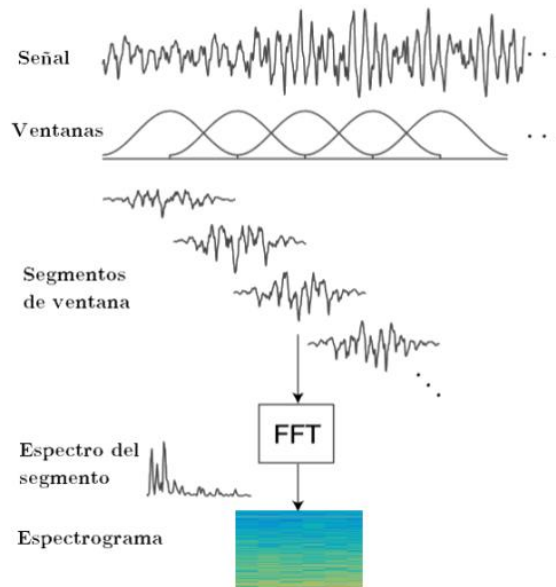
Un espectrograma es una representación visual que nos permite identificar diferentes variaciones en la frecuencia e intensidad del sonido durante un período de tiempo. Para estudiarlo se realiza un análisis espectral. Se trata principalmente del uso de sonidos o fragmentos musicales que constan de varios componentes.

Cada eje del espectrograma se traza de manera diferente

-El tiempo está representado por un eje horizontal de izquierda a derecha.

-El eje vertical muestra la frecuencia; Cuanto mayor sea la imagen emitida por el espectrograma, mayor será el tono.

-El color marcado en el espectrograma ayuda a determinar la intensidad del tono, cuanto más fuerte sea el color mostrado, más fuerte será el tono.



Como se puede observar las dos características principales utilizadas son la estructura de la secuencia de señal y espectro de segmento. El primero se representa como un oscilograma y el segundo como un sonograma o espectrograma.

Al terminar con la conversión de archivos de sonido a imagen realice el procesamiento de imágenes para ello utilice un Keras.

El Keras a través de su clase ImageDataGenerator nos permitió realizar aumentos de imagen sobre la marcha de una manera muy fácil.

Figura 12 Procesamiento de imágenes utilizando

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1./255, # rescale all pixel values from 0-255, so aftr this step all our pixel values are in range (0,1)
    shear_range=0.2, #to apply some random tranformations
    zoom_range=0.2, #to apply zoom
    horizontal_flip=True) # Image will be flipper horiz
test_datagen = ImageDataGenerator(rescale=1./255)
```

Nota. Fuente: Elaboración Propia.

Para ello utilizo tres métodos `flow()`, `flow_from_directory()` y `flow_from_dataframe()` para poder leer las imágenes de una gran matriz numpy ya que las imágenes estaban en carpetas.

Figura 13 Código fuente de `ImageDataGenerator`

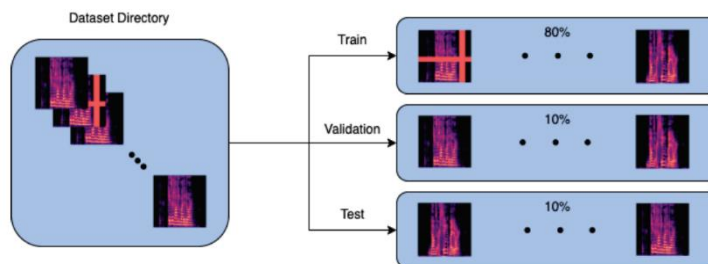
```
▶ training_set = train_datagen.flow_from_directory(
    './data/train',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False)
test_set = test_datagen.flow_from_directory(
    './data/val',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False )
```

↳ Found 731 images belonging to 8 classes.
Found 188 images belonging to 8 classes.

Nota. Fuente: Elaboración Propia.

Una vez generado nuestro dataset final de imágenes, están deben ser procesadas con la creación y división de la data.

Figura 14 Imagen referencial de creación y división de la data.



Nota. Fuente: `ImageDataGenerator`.

De este modo `plit_folders` nos permite dividir nuestra data en un 80% de entrenamiento y un 20% para prueba.

Figura 15 El 80% de entrenamiento y un 20% para prueba.

```
[ ] import split_folders
# To only split into training and validation set, set a tuple to `ratio`, i.e, `(.8, .2)`.
split_folders.ratio('./img_data/', output="./data", seed=1337, ratio=(.8, .2)) # default values
Copying files: 919 files [00:00, 2683.56 files/s]
```

Nota. Fuente: Elaboración Propia.

3.3.4. Implementación de algoritmos de clasificación para detectar la neumonía

Realice un cuadro comparativo para determinar cuál de las redes neuronales nos convendría para clasificar la neumonía.

Tabla 3 Comparación entre una Red Neuronal Artificial y Red Neuronal de Convolución.

Red Neuronal Artificial (ANN)	Red Neuronal de Convolución (CNN)
Utiliza capas completamente conectadas	Utiliza capas parcialmente conectadas, el tamaño de la conexión depende del tamaño del filtro.
Solo se puede utilizar para imágenes pequeñas.	Se puede utilizar para cualquier imagen.
Considera la imagen completa para aprender los patrones.	Considera solo la parte de la imagen que se encuentra en el campo receptivo del filtro.
El número de parámetros será muy alto.	El número de parámetros será muy inferior en comparación con ANN.
No es muy eficiente para las imágenes de reconocimiento facial.	Muy eficiente para imágenes y menos costoso en comparación con ANN.

No aprende los patrones en la jerarquía espacial, es decir, las capas superiores de ANN no combinan los resultados de las capas inferiores.	Aprende la jerarquía espacial de patrones, es decir, las capas superiores de CNN se forman combinando capas inferiores. Esto ayuda a identificar los patrones de manera más efectiva que ANN.
Requiere que las imágenes sean entrenadas nuevamente si varían algún patrón como el giro o desplazamiento de la imagen.	El aprendizaje se puede utilizar incluso si la imagen se gira o se desplaza.

Nota. Fuente: Elaboración Propia.

Se llegó a utilizar un algoritmo de clasificación de red neuronal convolucional (CNN) ya que sus enfoques son más estables para el reconocimiento de objetos en imágenes que otros tipos de redes neuronales, utilizando principios de álgebra lineal para el reconocimiento de patrones a través de la multiplicación de matrices

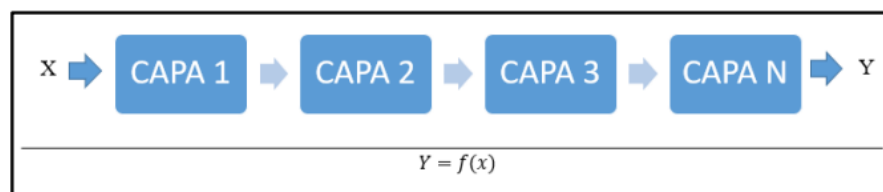
Antes de la implementación de las redes neuronales convolucionales, se realizaron algunas pruebas para determinar la estructura red.

Donde:

X=es la entrada

Y=es la predicción

Figura 16 Representación gráfica de la ecuación



$$y_1 = \text{Capa}_1(x)$$

$$y_2 = \text{Capa}_2(y_1)$$

$$y_3 = \text{Capa}_3(y_2)$$

$$Y = \text{Capa}_n(y_3)$$

$$Y = (\text{Capa}_3(\text{Capa}_2(\text{Capa}_1(X))))$$

Cada capa se encarga de encontrar patrones y estos son patrones aprendidos llamado peso.

Para este caso las CNN fueron utilizadas para clasificar las enfermedades respiratorias. Las imágenes obtenidas de la etapa de segmentación se utilizaron para el proceso de clasificación, para ello, cada imagen segmentada se utiliza como entrada del clasificador.

Para la clasificación se utilizó una Red Neuronal Convolutiva. Dado el bajo número de muestras de entrenamiento, 731, se construyó una CNN poco profunda, se usó el modelo "Sequential". Se usó 3 capas de convolución la primera convolución con 32 neuronas, la segunda convolución se creó con 64 neuronas igual que la tercera, así mismo se aplica el Pooling de tipo Average (Average pooling) para el redimensionamiento de las imágenes.

En la segunda capa, la primera convolución fue creada con 64 neuronas, la segunda convolución se creó con 64 neuronas y el Pooling de tipo Average y así mismo la tercera capa; al final de ambas capas se creó una neurona con activación sigmoid para representar las probabilidades de ser enferma o sana.

Figura 17 código fuente de la red neuronal convolutiva

```
[ ] model = Sequential()
input_shape=(64, 64, 3)
#1st hidden layer
model.add(Conv2D(32, (3, 3), strides=(2, 2), input_shape=input_shape))
model.add(AveragePooling2D((2, 2), strides=(2,2)))
model.add(Activation('relu'))
#2nd hidden layer
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(AveragePooling2D((2, 2), strides=(2,2)))
model.add(Activation('relu'))
#3rd hidden layer
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(AveragePooling2D((2, 2), strides=(2,2)))
model.add(Activation('relu'))
#Flatten
model.add(Flatten())
model.add(Dropout(rate=0.5))
#Add fully connected layer.
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(rate=0.5))
#Output layer
model.add(Dense(10))
model.add(Activation('softmax'))
model.summary()
```


Nota. Fuente: Elaboración Propia.

En cada capa, al entrenar dos clases, el algoritmo de clasificación pudo identificar objetos en función de las características que encontró la CNN en cada píxel. Se especificó que se ejecutara con tres capas convolucionales. Cuando la imagen de entrada se importa a la CNN y se pasa a las capas convolucionales, el algoritmo busca el mismo patrón en cada píxel en diferentes lugares de la imagen, moviendo una ventana a lo largo de la imagen.

Una vez que ha completado su tarea en la capa de convolución, la imagen ha entrado en la capa de pooling, que se dedica a reducir el tamaño de la imagen, en este caso también mueve una ventana, pero se logra la reducción, esto se hace tomando el valor máximo o promedio de cada ventana.

Así se entrenó la red neuronal. Después de eso, fueron probados y validados.

Tabla 4 Estructura del modelo entrenado.

Capa	Salida	Parametros
conv2d (Conv2D)	(None, 31, 31, 32)	896
average_pooling2d (AveragePooling2D)	(None, 15, 15, 32)	0
activación (Activación)	(None, 15, 15, 32)	0
onv2d_1 (Conv2D)	(None, 15, 15, 64)	18496
average_pooling2d_1 (AveragePooling2D)	(None, 7, 7, 64)	0
activation_1 (Activation)	(None, 7, 7, 64)	0
(None, 7, 7, 64)	(None, 7, 7, 64)	36928
average_pooling2d_2 (AveragePooling2D)	(None, 3, 3, 64)	0
activation_2 (Activation)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
dropout (Dropout)	(None, 576)	0
dense (Dense)	(None, 64)	36928
activation_3 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0

dense_1 (Dense)	None, 10)	650
activation_4 (Activation)	None, 10)	0

Nota. Fuente: Elaboración Propia.


De esta forma, entrenamos a un conjunto de imágenes que es el 80% del Dataset, con un valor inicial constante de 200 épocas correspondiente al número total de iteraciones completas por conjunto de imágenes de entrenamiento.

Figura 18 Código fuente del entrenamiento

```

▶ training_set = train_datagen.flow_from_directory(
    './data/train',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False)
test_set = test_datagen.flow_from_directory(
    './data/val',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False )

```

 Found 731 images belonging to 8 classes.
 Found 188 images belonging to 8 classes.

Nota. Fuente: Elaboración Propia.

Se puede predecir la salida utilizando la entrada con el entrenamiento, para luego evaluar el rendimiento y así comparar con la salida de la prueba. Por lo tanto, nos proporciona una medida de rendimiento de precisión en este caso.

Figura 19 Código Fuente para la entrada de entrenamiento

```

# Modelo de evaluación del
modelo.evaluate_generator (generator = test_set, steps = 50)

```

Nota. Fuente: Elaboración Propia.

Finalmente, los resultados de exactitud se pueden observar durante la fase de entrenamiento. Con un rango de 74% a 75%, faltan en los datos obtenidos mostrando a continuación.

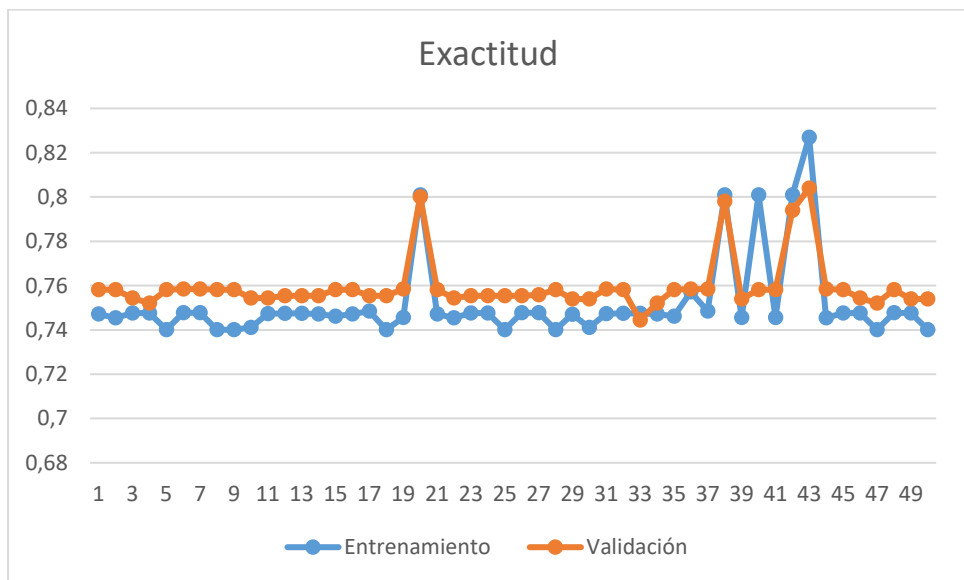
Tabla 5 Resultados de exactitud

Épocas	Exactitud		Prueba	
	Entrenamiento	Validación	Entrenamiento	Validación
1	0.7472	0.7581	0.3156	0.0692
2	0.454	0.7581	0.0152	0.028
3	0.7476	0.7944	0.0018	0.0194
4	0.7476	0.7521	0.0091	0.1311
5	0.7401	0.7581	0.0345	0.0779
6	0.7478	0.7584	0.0065	0.1047
7	0.7477	0.7584	3.37E-04	0.0504
8	0.801	0.7581	2.00E-04	0.0317
9	0.747	0.7581	5.90E-04	0.1584
10	0.7411	0.7944	0.0389	0.0699
11	0.7473	0.794	0.0115	0.0962
12	0.7475	0.794	0.0116	0.0963
13	0.7475	0.794	0.0025	0.0962
14	0.7472	0.794	0.0105	0.0965
15	0.7461	0.7581	0.023	0.1379
16	0.7971	0.7581	0.0143	0.103
17	0.7485	0.794	0.0056	0.0962
18	0.801	0.794	4.16E-05	0.0962
19	0.7456	0.7584	0.0307	0.1868
20	0.801	0.7581	2.67E-05	0.1893
21	0.7472	0.7581	0.0063	0.077
22	0.454	0.7944	0.0244	0.1126
23	0.7476	0.794	0.0111	0.064
24	0.7476	0.794	0.0226	0.1012
25	0.7401	0.794	0.0795	0.0881
26	0.7478	0.794	0.015	0.1949
27	0.7477	0.7581	0.0077	0.0468
28	0.801	0.7581	0.0165	0.0451
29	0.747	0.794	0.0286	2.65E-04
30	0.7411	0.794	0.0343	0.2277
31	0.7473	0.7584	0.0038	0.0967
32	0.7475	0.7581	0.0152	0.172
33	0.7475	0.7944	0.0018	0.1868
34	0.7472	0.7521	0.0091	0.1595
35	0.7461	0.7581	0.0345	0.1037

36	0.7971	0.7584	0.0065	0.1841
37	0.7485	0.7584	3.37E-04	0.207
38	0.801	0.7581	2.00E-04	0.0962
39	0.7456	0.794	5.90E-04	0.173
40	0.801	0.7581	0.0389	0.1924
41	0.7456	0.7581	0.0115	0.172
42	0.801	0.794	0.0116	0.1868
43	0.7472	0.794	0.0025	0.1595
44	0.454	0.7584	0.0105	0.1037
45	0.7476	0.7581	0.023	0.1841
46	0.7476	0.7944	0.0143	0.207
47	0.7401	0.7521	0.0056	0.0962
48	0.7478	0.7581	4.16E-05	0.064
49	0.7476	0.794	0.0307	0.1012
50	0.7401	0.794	2.67E-05	0.0881

Nota. Fuente: Elaboración Propia.

Figura 20 Gráfico de exactitud



Nota. Fuente: Elaboración Propia.

Durante la fase de entrenamiento, puedes asegurarte de que los resultados están ahí 73% y 75%, de sensibilidad los datos resultantes son los que se muestran en la siguiente figura.

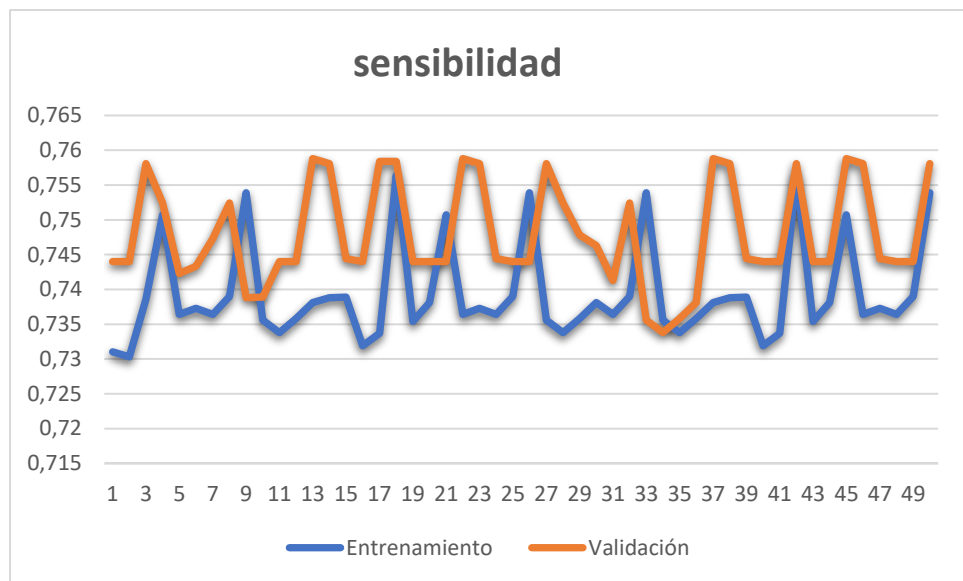
Tabla 6 Resultados de sensibilidad

Épocas	Sensibilidad		Prueba	
	Entrenamiento	Validación	Entrenamiento	Validación
1	0.731	0.744	1.6018	1.671
2	0.7303	0.744	1.5872	1.8139
3	0.7386	0.7581	1.5729	1.10606
4	0.7507	0.7524	1.559	1.23074
5	0.7364	0.7423	1.5452	1.4541
6	0.7373	0.7433	1.5319	1.8709
7	0.7364	0.7473	1.52E+00	1.8645
8	0.739	0.7524	1.51E+00	1.5933
9	0.7539	0.7388	1.49E+00	1.5777
10	0.7356	0.7389	1.4822	1.5622
11	0.7338	0.744	1.4716	1.5477
12	0.7358	0.744	1.4592	1.5331
13	0.7381	0.7588	1.4479	1.5189
14	0.7388	0.7581	1.4364	1.5046
15	0.7389	0.7444	1.4259	1.4916
16	0.7319	0.744	1.4716	1.4779
17	0.7337	0.7584	1.4592	1.4649
18	0.7564	0.7584	1.4479	1.7619
19	0.7354	0.744	1.4364	1.71468087
20	0.7381	0.744	1.4479	1.1254
21	0.7507	0.744	1.4364	1.6922
22	0.7364	0.7588	1.6018	1.67489
23	0.7373	0.7581	1.5872	1.6957
24	0.7364	0.7444	1.5729	1.2424
25	0.739	0.744	1.559	1.6892
26	0.7539	0.744	1.5452	1.07359
27	0.7356	0.7581	1.5319	1.39827
28	0.7338	0.7524	1.5192	1.294
29	0.7358	0.7478	1.5064	1.04762
30	0.7381	0.7463	1.494	1.5933
31	0.7364	0.7413	1.4822	1.5777
32	0.739	0.7524	1.4716	1.5622
33	0.7539	0.7356	1.4592	1.5477
34	0.7356	0.7338	1.4479	1.5331
35	0.7338	0.7358	1.4364	1.5189
36	0.7358	0.7381	1.4259	1.5046
37	0.7381	0.7588	1.41E+00	1.4916
38	0.7388	0.7581	1.41E+00	1.4779
39	0.7389	0.7444	1.39E+00	1.4649

40	0.7319	0.744	1.3861	1.4524
41	0.7337	0.744	1.3762	1.4404
42	0.7564	0.7581	1.4364	1.4281
43	0.7354	0.744	1.4259	1.4163
44	0.7381	0.744	1.41E+00	1.4045
45	0.7507	0.7588	1.40333	1.3937
46	0.7364	0.7581	1.39348	1.3829
47	0.7373	0.7444	1.38E+00	1.3732
48	0.7364	0.744	1.33333	1.3616
49	0.739	0.744	1.3612	1.3514
50	0.7539	0.7581	1.35E+00	1.3714

Nota. Fuente: Elaboración Propia.

Figura 21 Figura de sensibilidad



Nota. Fuente: Elaboración Propia.

En la fase de entrenamiento del modelo se puede observar que los resultados van de 65% al 66% y los datos obtenidos se muestran a continuación.

Tabla 7 Resultado de Precisión

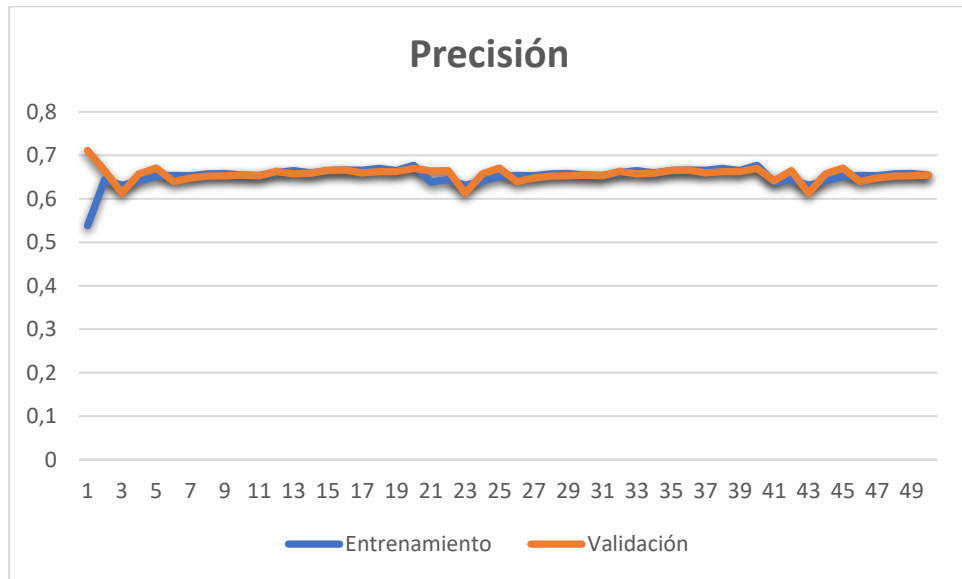
Épocas	Precisión		Prueba	
	Entrenamiento	Validación	Entrenamiento	Validación
1	0.5383	0.711	1.6018	1.5933

2	0.6442	0.6647	1.5872	1.5777
3	0.6309	0.6131	1.5729	1.5622
4	0.6413	0.6577	1.559	1.5477
5	0.6518	0.6703	1.5452	1.5331
6	0.6537	0.6397	1.5319	1.5189
7	0.6529	0.6482	1.52E+00	1.5046
8	0.6573	0.652	1.51E+00	1.4916
9	0.6579	0.6527	1.49E+00	1.4779
10	0.6541	0.6541	1.4822	1.4649
11	0.6538	0.6531	1.4716	1.4524
12	0.6601	0.6636	1.4592	1.4404
13	0.665	0.6577	1.4479	1.4281
14	0.6598	0.6591	1.4364	1.4163
15	0.6644	0.6657	1.4259	1.4045
16	0.666	0.6664	1.4149	1.3937
17	0.6658	0.6594	1.4055	1.3829
18	0.6699	0.6622	1.3949	1.3732
19	0.6651	0.6629	1.3861	1.3616
20	0.6771	0.6699	1.3762	1.3514
21	0.6383	0.6641	1.6018	1.5933
22	0.6442	0.6647	1.5872	1.5777
23	0.6309	0.6131	1.5729	1.5622
24	0.6413	0.6577	1.559	1.5477
25	0.6518	0.6703	1.5452	1.5331
26	0.6537	0.6397	1.5319	1.5189
27	0.6529	0.6482	1.5192	1.5046
28	0.6573	0.652	1.5064	1.4916
29	0.6579	0.6527	1.494	1.4779
30	0.6541	0.6541	1.4822	1.4649
31	0.6538	0.6531	1.4716	1.4524
32	0.6601	0.6636	1.4592	1.4404
33	0.665	0.6577	1.4479	1.4281
34	0.6598	0.6591	1.4364	1.4163
35	0.6644	0.6657	1.4259	1.4045
36	0.666	0.6664	1.4149	1.3937
37	0.6658	0.6594	1.41E+00	1.3829
38	0.6699	0.6622	1.39E+00	1.3732
39	0.6651	0.6629	1.39E+00	1.3616
40	0.6771	0.6699	1.3762	1.3514
41	0.6383	0.6411	1.6018	1.5933
42	0.6442	0.6647	1.5872	1.5777
43	0.6309	0.6131	1.5729	1.5622
44	0.6413	0.6577	1.56E+00	1.5477
45	0.6518	0.6703	1.5452	1.5331

46	0.6537	0.6397	1.5319	1.5189
47	0.6529	0.6482	1.52E+00	1.5046
48	0.6573	0.652	1.5064	1.4916
49	0.6579	0.6527	1.494	1.4779
50	0.6541	0.6541	1.48E+00	1.4649

Nota. Fuente: Elaboración Propia.

Tabla 8 Figura de Precisión



Nota. Fuente: Elaboración Propia.

Al final el modelo realizo algunas predicciones sobre el conjunto de datos de prueba; Para ello se debe restablecer test_set antes que llame al redict_generator.

Figura 22 Código fuente para restablecer el test_set

```
test_set.reset ()
pred = model.predict_generator (test_set, pasos = 50, detallado = 1)
```

Nota. Fuente: Elaboración Propia.

Por consiguiente, procedí a guardar los resultados en un archivo CSV, para luego plasmar los resultados a través de un gráfico.

Figura 23 Código fuente para guardar los resultados en un CSV

```
results = pd.DataFrame ({"Nombre de archivo": nombres de archivo,  
                        "Predictions": predictions}, orient = 'index')  
results.to_csv ("prediction_results.csv", index = False)
```

Nota. Fuente: Elaboración Propia.

De este modo, se evaluó el método propuesto para identificar la neumonía mediante el desarrollo de un programa. Por tanto, esta investigación proporciona una herramienta que sirve para realizar un análisis sobre la conducta de la señal respiratoria en un paciente sano o con neumonía. Para las técnicas y recolección de datos utilizamos la observación científica, dicho instrumento es la ficha de observación que ayudo a validar los resultados de la implementación de los métodos planteados anteriormente.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

- a) En la presente investigación construí un protocolo de adquisición de sonidos de pacientes sanos y con infecciones respiratorias, para la elaboración de una base de datos.
- b) La construcción de base de datos de los archivos digitales se segmento por tipo de enfermedades respiratorias tales como: Asma, Bronquiectasia, Bronquiolitis, Enfermedad pulmonar obstructiva crónica, infección del tracto respiratorio inferior, infección respiratoria alta, neumonía y pacientes sanos.
- c) Para la conversión de la base de datos de audios digitales a imágenes en formato PNG utilice la biblioteca de librosa Python para extraer Spectrogram para cada archivo de audio.
- d) La implementación de red neuronal convolucional se realizó utilizando Colaboratory te permite ejecutar y programar en Python en tu navegador, ya que no requiere configuración ni acceso a GPUs e incluye todas las librerías de Python.
- e) Se concluye que en la evaluación del método propuesto los resultados fueron de un 75% en sensibilidad, 75% en exactitud y un 69% en precisión.

4.2. Recomendaciones.

Se recomienda realizar grabaciones propias en ambientes controlados para así tener mejores resultados, si alguien desea repetir el método y así obtener mejores resultados.

REFERENCIAS.

- [1] J. Fiz, M. Lozano, and R. Jané, “Avances en el análisis de sonidos respiratorios para la monitorización de pacientes con patologías respiratorias,” *Medicina respiratoria*, vol. 8, no. 2, pp. 17–24, 2015, [Online]. Available: <http://www.neumologiaysalud.es/descargas/R8/R82-3.pdf>
- [2] K. Kosasih, U. R. Abeyratne, V. Swarnkar, and R. Triasih, “Wavelet Augmented Cough Analysis for Rapid Childhood Pneumonia Diagnosis,” *IEEE Trans Biomed Eng*, vol. 62, no. 4, pp. 1185–1194, 2015, doi: 10.1109/TBME.2014.2381214.
- [3] J. Nikunen, A. Diment, and T. Virtanen, “Separation of Moving Sound Sources Using Multichannel NMF and Acoustic Tracking,” *IEEE/ACM Trans Audio Speech Lang Process*, vol. 26, no. 2, pp. 281–295, Feb. 2018, doi: 10.1109/TASLP.2017.2774925.
- [4] D. Cao *et al.*, “Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review,” *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1029–1042, Nov. 2020, doi: 10.35833/MPCE.2020.000552.
- [5] Z. Neili, M. Fezari, and R. Abdeghani, “Analysis of Acoustic Parameters from Respiratory Signal in COPD and Pneumonia patients,” *2018 International Conference on Signal, Image, Vision and their Applications, SIVA 2018*, vol. 1, pp. 5–8, 2019, doi: 10.1109/SIVA.2018.8661111.
- [6] J. S. Huang, Y. F. Chen, and J. C. Hsu, “Design of a clinical decision support model for predicting pneumonia readmission,” *Proceedings - 2014 International Symposium on Computer, Consumer and Control, IS3C 2014*, pp. 1179–1182, 2014, doi: 10.1109/IS3C.2014.306.
- [7] L. Jiansheng, H. Jinliang, S. Jianjing, W. Zhiwan, L. Suyun, and W. Jiehua, “Research on diagnostic models of pneumonia syndromes in TCM based on fuzzy-neural net,” *2009 Chinese Control and Decision Conference, CCDC 2009*, pp. 933–937, 2009, doi: 10.1109/CCDC.2009.5191911.
- [8] O. Zenteno, B. Castaneda, and R. Lavarello, “Spectral-based pneumonia detection tool using ultrasound data from pediatric populations,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2016-Octob, pp. 4129–4132, 2016, doi: 10.1109/EMBC.2016.7591635.

- [9] R. T. Sousa *et al.*, “Evaluation of classifiers to a childhood pneumonia computer-aided diagnosis system,” *Proc IEEE Symp Comput Based Med Syst*, pp. 477–478, 2014, doi: 10.1109/CBMS.2014.98.
- [10] T. H. Pingale and H. T. Patil, “Analysis of Cough Sound for Pneumonia Detection Using Wavelet Transform and Statistical Parameters,” *2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017*, pp. 1–6, 2018, doi: 10.1109/ICCUBEA.2017.8463900.
- [11] W. Afifi, I. Warsito, ... M. S.-2017 international, and undefined 2017, “The development of an online pneumonia risk prediction system,” *ieeexplore.ieee.org*, Accessed: Feb. 28, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8308063/>
- [12] Y. Amrulloh, U. Abeyratne, V. Swarnkar, and R. Triasih, “Cough Sound Analysis for Pneumonia and Asthma Classification in Pediatric Population,” *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, vol. 2015-Octob, pp. 127–131, 2015, doi: 10.1109/ISMS.2015.41.
- [13] R. Moreno, *Departamento de enfermedades respiratorias Universidad Católica de Chile*. 2015.
- [14] W. C. Gómez, *El Manual Moderno*. Bogotá, 2003.
- [15] O. y Schafer, “Salcedo y Teixeira,” 2009.
- [16] B. Logan, “Mel frequency cepstral coefficients for music modeling,” *International Symposium on Music Information Retrieval*, 2000.
- [17] “Lenguajes de programación y su papel en la Ingeniería de Software.” <https://repository.uaeh.edu.mx/revistas/index.php/xikua/article/download/3207/3180?inline=1> (accessed Feb. 28, 2023).
- [18] C. C. Aggarwal, M. A. Bhuiyan, and M. Al Hasan, “Frequent pattern mining algorithms: A survey,” *Frequent Pattern Mining*, vol. 9783319078212, pp. 19–64, Jul. 2014, doi: 10.1007/978-3-319-07821-2_2/COVER.
- [19] “Machine Learning and IoT: A Biological Perspective - Google Libros.” [https://books.google.com.pe/books?hl=es&lr=&id=cEUPEAAAQBAJ&oi=fnd&pg=PP1&dq=\(Sen,+Datta,+%26+Mitra,+2019\).&ots=ElRZCVw7Iv&sig=4XVis570vfWHIx46SLCd-rV8BA4#v=onepage&q=\(Sen%2C%20Datta%2C%20%26%20Mitra%2C%202019\).&f=false](https://books.google.com.pe/books?hl=es&lr=&id=cEUPEAAAQBAJ&oi=fnd&pg=PP1&dq=(Sen,+Datta,+%26+Mitra,+2019).&ots=ElRZCVw7Iv&sig=4XVis570vfWHIx46SLCd-rV8BA4#v=onepage&q=(Sen%2C%20Datta%2C%20%26%20Mitra%2C%202019).&f=false) (accessed Feb. 28, 2023).

- [20] G. Kamath and C. Tzamos, “Anaconda: A non-adaptive conditional sampling algorithm for distribution testing,” *Proc West Mark Ed Assoc Conf*, pp. 679–693, 2019, doi: 10.1137/1.9781611975482.43.

ANEXOS.

Anexo 1. Resolución de aprobación del proyecto de investigación

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N° 0938-A-2019/FIAU-USS

Chiclayo, 22 de julio de 2019

VISTO:

El Acta de Reunión N° 1507-2019 de fecha 15 de julio de 2019, en el que el Comité Evaluador aprueba el Tema de Tesis: *"IDENTIFICACIÓN AUTOMÁTICA DE NEUMONÍA MEDIANTE EL PROCESAMIENTO DIGITAL DEL SONIDO"*, presentado por el(los) estudiante(s) **JARA TUCTO ALEXANDER** de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS**, y;

CONSIDERANDO:

Que, acorde con lo establecido en el Artículo 8° de la Ley Universitaria, Ley N° 30220, la autonomía inherente a las Universidades se ejerce de conformidad con lo establecido en la Constitución, la presente Ley y demás normativa aplicable. Esta autonomía se manifiesta en los siguientes regímenes: normativo, de gobierno, académico, administrativo y económico.

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a la letra dice: *"La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas."*;

Que, de conformidad con el Reglamento de Investigación versión 3.0 de la Universidad Señor de Sipán en su artículo 76° que a la letra dice: *"El docente de las asignaturas de Investigación I e Investigación II, serán denominados Asesores de Investigación (AI)";* y

Estando a lo expuesto y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

Anexo 2. Resolución de Ampliación de vigencia de tesis



Universidad
Señor de Sipán

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO RESOLUCIÓN N°0139-2023/FIAU-USS

Pimentel, 10 de marzo de 2023

VISTO:

El Acta de reunión N°0803-2023 del Comité de investigación de la Ingeniería de Sistemas remitida mediante Oficio 0059-2023/FIAU-IS-USS de fecha 09 de marzo de 2023, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma."

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

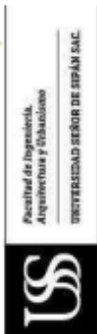
Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N° 059 - 2023, de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y AMBIENTE, a cargo de los estudiantes y /o egresados del Programa de estudios INGENIERÍA DE SISTEMAS, hasta la fecha que indica la presente resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1: AMPLIAR VIGENCIA, de la Tesis a cargo de los estudiantes y /o egresados del Programa de estudios de INGENIERÍA DE SISTEMAS que se detallan en el anexo de la presente Resolución.

ARTÍCULO 2: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.





FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°0139-2023/FIAU-USS

Pimentel, 10 de marzo de 2023

ANEXO

AMPLIACIÓN DE VIGENCIA DE TEMA DE TESIS

	APELLIDOS	TESIS	OBSERVACION	RESOLUCION APROBACIÓN/AMPLIACIÓN
1	MERA TAPIA GABRIEL HUGO	ESTRATEGIA TECNOLÓGICA PARA MEJORAR LA PLANEACIÓN DE TECNOLOGÍA, INFORMACIÓN Y COMUNICACIÓN DE ELECTRO ORIENTE S.A	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	2109-2020/FIAU-USS
2	TORRES CHIMOY EDILBERTO PAÚL DÍAZ SANCHEZ RICARDO RICARDO	DESARROLLO DE UN MÉTODO DE IDENTIFICACIÓN DE CRYPTOHACKING MEDIANTE EL ANÁLISIS DE CÓDIGO JAVASCRIPT UTILIZANDO APRENDIZAJE AUTOMÁTICO	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1347-2020/FIAU-USS
3	GUZMAN LOPEZ RICARDO ARTURO	EVALUACIÓN DE LA USABILIDAD DE UNA APLICACIÓN MÓVIL DE SOPORTE TÉCNICO UTILIZANDO LA NORMA ISO/IEC 25010	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	2320-2020/FIAU-USS
4	GARCIA GUTIERREZ KEVIN GIANMARCO GUEVARA RAMIREZ CESAR ALBERTO	DETECCIÓN DE PHISHING POR ENVENENAMIENTO DEL SERVIDOR DE NOMBRE DE DOMINIO PARA EVITAR EL ROBO DE INFORMACIÓN EN APLICACIONES WEB DE MICROEMPRESAS PERUANAS UTILIZANDO APRENDIZAJE DE MÁQUINA	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1329-2020/FIAU-USS
5	JARA TUCTO ALEXANDER	IDENTIFICACIÓN AUTOMÁTICA DE NEUMONÍA MEDIANTE EL PROCESAMIENTO DIGITAL DEL SONIDO	AMPLIAR HASTA EL 31 DE JULIO DEL 2023	0938-2019/FIAU-USS
6	GONZALES GONZALES CHRISTIAN ERICK	ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP PARA EL RENDIMIENTO DE APLICACIONES WEB	AMPLIAR HASTA EL 30 DE MAYO DEL 2023	1120-2017/FIAU-USS



Universidad
Señor de Sipán

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°0139-2023/FIAU-USS

Pimentel, 10 de marzo de 2023

	APELLIDOS	TESIS	OBSERVACION	RESOLUCION APROBACIÓN
7	REUPO ATTO PAUL GIANMARCO	EVALUACIÓN DE LA EFICIENCIA DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA LA IDENTIFICACIÓN DE ENFERMEDADES EN LOS CULTIVOS DE CAÑA DE AZÚCAR	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1332-2020/FIAU-USS
8	ECCA CHAVEZ ORLANDO ISMAEL SECLÉN FUENTES WALTHER JUNIOR	EVALUACIÓN DE EXPERIENCIA DE USUARIO EN EL DESARROLLO DE APLICACIONES WEB EDUCATIVAS UNIVERSITARIAS BASADO EN EL ESTÁNDAR ISO/IEC 25022. CASO DE ESTUDIO: CAMPUS DE LA UNIVERSIDAD SEÑOR DE SIPÁN	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1321-2020/FIAU-USS
9	TINEO GONZÁLES EDWIN EUSEBIO	MODELO DE EVALUACION PARA LA ADOPCIÓN DE SOFTWARE COMO SERVICIO DE CLOUD COMPUTING EN LAS PYMES PARA LA REGIÓN LAMBAYEQUE	AMPLIAR HASTA EL 30 DE MAYO DEL 2023	0832-2016/FIAU-USS
10	DURAND MORILLO DARWIN	DESARROLLO DE UN MODELO DE GESTIÓN DE INCIDENTES DE TI BASADO EN ESTÁNDARES DE BUENAS PRÁCTICAS PARA MEJORAR EL SERVICIO DE TI EN GOBIERNOS LOCALES. CASO DE ESTUDIO: MUNICIPALIDAD DISTRITAL DE CERRO COLORADO	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1337-2020/FIAU-USS
11	GUERRERO SERNAQUE LUIGGI EMILIO LOZA VARGAS FERNANDO AUGUSTO	DESARROLLO DE UNA APLICACIÓN MIDDLEWARE PARA EL ENVÍO Y VALIDACIÓN DE COMPROBANTES ELECTRÓNICOS A SUNAT	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	2363-2020/FIAU-USS



Dr. Victor Alexi Tuesta Montez
Decano (E) / Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.



Dr. Halyn Alvarez Vásquez
Secretario Académico Facultad de
Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

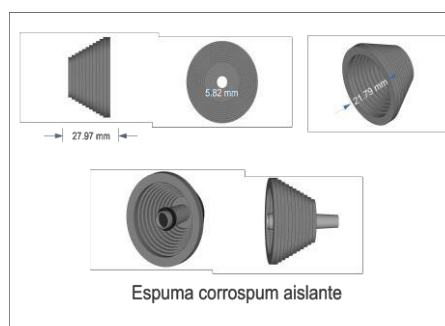
Anexo 3. Características técnicas de los equipos

EQUIPOS	CARACTERÍSTICAS
 Micrófono SENNHEISER e 945	<ul style="list-style-type: none"> ➤ Dimensiones 47 x 186 mm ➤ Sensibilidad 2,0mV/Pa = -54dB ➤ Peso 365 g ➤ frecuencia del Micrófono - 18.000 Hz
 Panel acústico	<ul style="list-style-type: none"> ➤ Espuma corrosputum aislante
 Cable De Micrófono	<ul style="list-style-type: none"> ➤ Tipo De Conector Xlr ➤ Longitud 5 Metros
 Behringer U-Phoria UM2	<ul style="list-style-type: none"> ➤ Resolución de 48kHz para calidad de audio profesional. ➤ Interfaz de audio USB 2x2 para grabación de micrófonos ➤ Puerto USB para conexión y alimentación.
 hp pavilion g4-1386la	<ul style="list-style-type: none"> ➤ Sistema Operativo: Win. 10 ➤ Procesador: - Intel Core i5-2450M (2,5 GHz). ➤ Memoria: - DDR3 de 4 GB (1 x 4). ➤ Disco rígido: - SATA de 750 GB (5400 rpm).
 Software REAPE	<ul style="list-style-type: none"> ➤ REAPER es una estación de trabajo de audio digital y un software secuenciador MIDI. ➤ La versiones disponibles para Microsoft Windows, macOS y Linux.

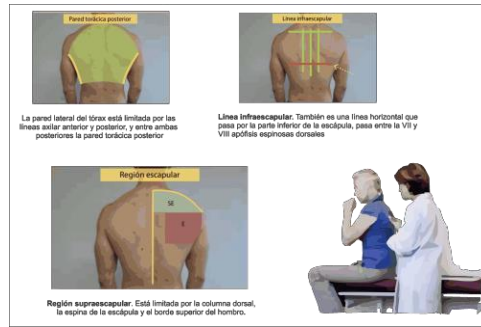
Anexo 4. Implementación de los equipos adquiridos



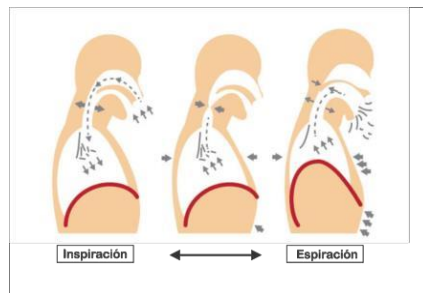
Anexo 5. Panel acústico para el micrófono



Anexo 6. Segmentación de la pared torácica para la auscultación



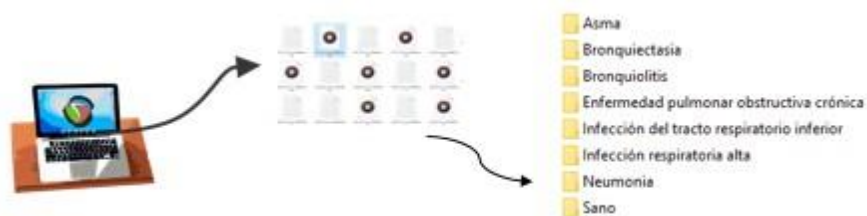
Anexo 7. Ejercicios de inspiración/inspiración



Anexo 8. Software para la grabación de la auscultación



Anexo 9. Selección y segmentación por tipo de infección respiratoria

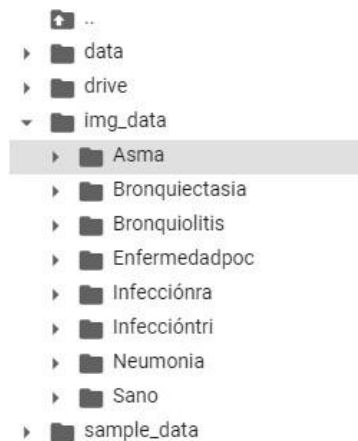


Anexo 10. Conversión de audio a imagen por tipo de infección respiratoria

```
[ ] genres = 'Asma Bronquiectasia Bronquiolitis Enfermedadpoc Infeccióntri Infecciónra Neumonia Sano'.split()
for g in genres:
    pathlib.Path(f'img_data/{g}').mkdir(parents=True, exist_ok=True)
    for filename in os.listdir(f'./drive/My Drive/BD_sonidos/{g}'):
        songname = f'./drive/My Drive/BD_sonidos/{g}/{filename}'
        y, sr = librosa.load(songname, mono=True, duration=5)
        print(y.shape)
        plt.specgram(y, NFFT=2048, Fs=2, Fc=0, noverlap=128, sides='default', mode='default', scale='dB');
        plt.axis('off');
        plt.savefig(f'img_data/{g}/{filename[:-3].replace(".", "")}.png')
        plt.clf()

(110250,)
(110250,)
(110250,)
(110250,)
(110250,)
(110250,)
(110250,)
(110250,)
(110250,)
(110250,)
```

Anexo 11. Dirección de img_data con las imágenes convertidas



Anexo 12. Procesamiento de imágenes utilizando Keras

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1./255, # rescale all pixel values from 0-255, so aftr this step all our pixel values are in range (0,1)
    shear_range=0.2, #to apply some random tranformations
    zoom_range=0.2, #to apply zoom
    horizontal_flip=True) # image will be flipper horiz
test_datagen = ImageDataGenerator(rescale=1./255)
```

Anexo 13. Código fuente de ImageDataGenerator.

```
▶ training_set = train_datagen.flow_from_directory(
    './data/train',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False)
test_set = test_datagen.flow_from_directory(
    './data/val',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False )
```

↳ Found 731 images belonging to 8 classes.
Found 188 images belonging to 8 classes.

Anexo 14. El 80% de entrenamiento y un 20% para prueba

```
[ ] import split_folders
# To only split into training and validation set, set a tuple to `ratio`, i.e. `(.8, .2)`.
split_folders.ratio('./img_data/', output='./data', seed=1337, ratio=(.8, .2)) # default values
```

↳ Copying files: 919 files [00:00, 2683.56 files/s]

Anexo 15. Código fuente de la red neuronal convolucional

```
[ ] model = Sequential()
input_shape=(64, 64, 3)
#1st hidden layer
model.add(Conv2D(32, (3, 3), strides=(2, 2), input_shape=input_shape))
model.add(AveragePooling2D((2, 2), strides=(2,2)))
model.add(Activation('relu'))
#2nd hidden layer
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(AveragePooling2D((2, 2), strides=(2,2)))
model.add(Activation('relu'))
#3rd hidden layer
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(AveragePooling2D((2, 2), strides=(2,2)))
model.add(Activation('relu'))
#Flatten
model.add(Flatten())
model.add(Dropout(rate=0.5))
#Add fully connected layer.
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(rate=0.5))
#Output layer
model.add(Dense(10))
model.add(Activation('softmax'))
model.summary()
```

Anexo 16. Código fuente del entrenamiento

```
▶ training_set = train_datagen.flow_from_directory(
    './data/train',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False)
test_set = test_datagen.flow_from_directory(
    './data/val',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical',
    shuffle = False )
```

↳ Found 731 images belonging to 8 classes.
Found 188 images belonging to 8 classes.

Anexo 17. Código Fuente para la entrada de entrenamiento

```
# Modelo de evaluación del
modelo.evaluate_generator (generator = test_set, steps = 50)
```

Anexo 18. Código fuente para restablecer el test_set

```
test_set.reset ()
pred = model.predict_generator (test_set, pasos = 50, detallado = 1)
```

Anexo 19. Código fuente para guardar los resultados en un CSV

```
results = pd.DataFrame ({"Nombre de archivo": nombres de archivo,
                        "Predictions": predictions}, orient = 'index')
results.to_csv ("prediction_results.csv", index = False)
```