Neiva, ___04/04/2022_____

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

JUAN CARLOS CABRERA MUÑOZ_____, con C.C. No. 1075278751_____,

JESÚS HERNEY SANCHEZ RODRÍGUEZ_____, con C.C. No. 1075306339_____,

_____, con C.C. No. _____,

_____, con C.C. No. _____,

Autor(es) de la tesis y/o trabajo de grado o _____

Titulado DISEÑO E IMPLMENTACIÓN DE SISTEMA DE RECONOCIMIENTO POR VOZ BASADO EN INTELIGENCIA ARTIFICIAL Y PROCESAMIENTO DIGITAL DE SEÑALES_____

_____

_____

presentado y aprobado en el año ___2022___ como requisito para optar al título de

INGENIERO ELECTRÓNICO

_____;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.

- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.

- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores” , los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE: JUAN CARLOS CABRERA

Firma: Juan Carlos Cabrera

EL AUTOR/ESTUDIANTE:

Firma: _____

EL AUTOR/ESTUDIANTE: JESÚS HERNEY SANCHEZ

Firma: _____

EL AUTOR/ESTUDIANTE:

Firma: _____

TÍTULO COMPLETO DEL TRABAJO: DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE RECONOCIMIENTO POR VOZ BASADO EN INTELIGENCIA ARTIFICIAL Y PROCESAMIENTO DIGITAL DE SEÑALES

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Sánchez Rodríguez	Jesús Hemey
Cabera Muñoz	Juan Carlos

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Cortés Cabezas	Albeiro

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería Electrónica

CIUDAD: Neiva **AÑO DE PRESENTACIÓN:** 2022 **NÚMERO DE PÁGINAS:** 55

TIPO DE ILUSTRACIONES (Marcar con una **X**):

Diagramas_X_ Fotografías___ Grabaciones en discos___ Ilustraciones en general_x_ Grabados___ Láminas___ Litografías___ Mapas___ Música impresa___ Planos___ Retratos___ Sin ilustraciones___ Tablas o Cuadros_x_

SOFTWARE requerido y/o especializado para la lectura del documento:

MATERIAL ANEXO:

PREMIO O DISTINCIÓN *(En caso de ser LAUREADAS o Meritoria)*:

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

Español	Inglés	Español	Inglés
1. Voz	Voice	**6.** Inteligencia Artificial	Artificial Intelligence
2. DSP	DSP	**7.** Software Libre	Open Source Sorftware
3. Espectrograma	Spectrogram	**8.** _____	_____
4. Reconocimiento	Recognition	**9.** _____	_____
5. Biometría	Biometry	**10.** _____	_____

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El objetivo de este proyecto fue diseñar e implementar un sistema que permita reconocer la voz de un locutor haciendo uso de teorías de procesamiento digital de señal es y conceptos de inteligencia artificial. Para esto, se definió una serie de etapas correspondiente a la adquisición de los datos correspondientes a audios dónde se definen las características con los que deben ser grabados, posteriormente su preprocesamiento mediante teoremas de tratamiento de audio tanto en el dominio temporal como en el dominio frecuencial, para así poder generar los coeficientes Cepstrales en la frecuencias de mel, que permiten representar mediante un espectrograma las características de la voz de un locutor, lo que se convierte en los datos que alimentan los modelos de inteligencia artificial, convirtiendo así la voz de un audio a una imagen.

Una vez estructurado un dataset con el conjunto de espectrogramas, se diseñó tres modelos de inteligencia artificial con el propósito de poder comparar las capacidades de sus topologías, para finalmente poder ejecutar el modelo escogido mediante un script que genera una interacción entre el sistema y el usuario, logrando definir un camino de cómo se puede construir un sistema de reconocimiento de locutor por medio de su voz, con el uso de lenguajes de programación libre, y conceptos que permitan reducir su complejidad contando con un resultado óptimo basado en los recursos implementados.

ABSTRACT: (Máximo 250 palabras)

The aim of this project is to design and develop a voice recognition system using digital signals processing theorems and artificial intelligence concepts. In order to do it, it was defined a series of steps to follow as the audios acquisition environment, then the audios preprocessing implementing time domain and frequency domain signals processing theorems, so it could help to generate the Mel Frequency Cepstrals Coeficients that are the base of the matrix shown graphically as a Spectrogram, where the speaker voice characteristics are shown. So, it drives to transform a voice from audio to image.

Once defined the spectrograms dataset, there were designed a series of three artificial intelligence models with the aim to compare their topologies capabilities, then as a result, writing a script to execute the chosen model with a user-system interaction. So it is possible to define that it can be build a voice recognition system using open source programming languages and low complexity/cost concepts.

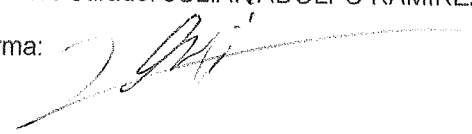
APROBACION DE LA TESIS

Nombre Jurado: VLADIMIR MOSQUERA CERQUERA

Firma:

Nombre Jurado: JULIAN ADOLFO RAMIREZ GUTIÉRREZ

Firma:



DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE RECONOCIMIENTO POR VOZ BASADO EN INTELIGENCIA ARTIFICIAL Y PROCESAMIENTO DIGITAL DE SEÑALES

JESÚS HERNEY SÁNCHEZ RODRÍGUEZ
JUAN CARLOS CABRERA MUÑOZ

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA
2021

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE RECONOCIMIENTO POR VOZ BASADO EN INTELIGENCIA ARTIFICIAL Y PROCESAMIENTO DIGITAL DE SEÑALES

JESÚS HERNEY SÁNCHEZ RODRÍGUEZ
JUAN CARLOS CABRERA MUÑOZ

TRABAJO DE GRADO

DR. ALBEIRO CORTÉS CABEZAS
DIRECTOR

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA
2021

Nota de Aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Neiva, 15 de marzo de 2022

Dedicamos este trabajo a todos quienes nos acompañaron a lo largo de estos años, dándonos su apoyo y motivación para llegar hasta este punto y así conseguir este importante logro.

AGRADECIMIENTOS

Dedicamos este logro a nuestros padres que, con sus esfuerzos y paciencia, nos apoyaron a conseguir este importante escalón en nuestras vidas. Reconocemos también a nuestros docentes y amigos, que nos acompañaron en esta larga carrera y con quienes pudimos contar incondicionalmente para llegar a la meta. Sin ellos, lograr este gran título resultaría una tarea aún más difícil de lo que representa persé.

CONTENIDO

Pág.

1. INTRODUCCIÓN 13
2. OBJETIVOS 14
2.1 OBJETIVO GENERAL 14
2.2 OBJETIVOS ESPECÍFICOS 14
3. PLANTEAMIENTO DEL PROBLEMA 15
3.1 DEFINICIÓN DEL PROBLEMA 15
3.2 JUSTIFICACIÓN 16
4. MARCO TEÓRICO 17
5. DESARROLLO DEL PROYECTO 22
5.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO 43
CONCLUSIONES 50
RECOMENDACIONES 52
BIBLIOGRAFÍA 53

LISTA DE TABLAS

Pág

Tabla 1. Arquitectura Modelo 1 36

Tabla 2. Arquitectura Modelo 2 39

Tabla 3. Arquitectura Modelo 3 40

Tabla 4. Validación 42

Tabla 5. Audios por locutor 43

Tabla 6. Comparativa de arquitecturas 45

Tabla 7. Matriz de confusión modelo 1 46

Tabla 8. Matriz de confusión modelo 2 46

Tabla 9. Matriz de confusión modelo 2 47

Tabla 10. Matriz de confusión validación 49

LISTA DE FIGURAS

Pág

Figura 1. Imagen de entrada y kernel 18

Figura 2. Mapa de detención de características 19

Figura 3. Muestreo por Max-Pooling 19

Figura 4. Señal de audio del canal izquierdo (Muestra vs Amplitud) 23

Figura 5. Señal de audio canal izquierdo (Tiempo vs Amplitud) 24

Figura 6. Audio aleatorio 1 (Muestra vs Amplitud) 25

Figura 7. Audio aleatorio 2 (Muestra vs Amplitud) 25

Figura 8. Audio aleatorio 3 (Muestra vs Amplitud) 25

Figura 9. Audio aleatorio 4 (Muestra vs Amplitud) 25

Figura 10. Audio aleatorio 5 (Muestra vs Amplitud) 25

Figura 11. Audio original (azul), audio filtrado por amplitud (naranja) 26

Figura 12. Audio en dominio del tiempo 27

Figura 13. Audio en dominio de la frecuencia 27

Figura 14. Ventana inicial (azul) y ventana final (naranja) 28

Figura 15. Tramo enventanado con Hamming 28

Figura 16. Coeficientes vs. Potencia 29

Figura 17. a) Potencia vs Muestras b) Potencia con 26 filtros de Mel c) Potencia con 26 filtros de Mel (Escala logarítmica) 30

Figura 18. a) Potencia con 26 filtros de Mel (Escala logarítmica) b) Transformada Discreta de Coseno 31

Figura 19. Espectrograma de audio de prueba 31

Figura 20. Espectrograma obtenido con Librosa 32

Figura 21. Espectrograma de coeficientes 32

Figura 22. Espectrograma de potencias 33

Figura 23. Entrada + Convolución 2D, Modelo 1 34

Figura 24. Convolución 2D + Totalmente Conectada ReLU, Modelo 35

Figura 25. TC ReLU + FC Softmax, Modelo 1 35

Figura 26. Topología del Modelo 1 36

Figura 27. Entrada + Convolución 2D, Modelo 2 37

Figura 28. Convolución 2D + MaxPooling, Modelo 2 38

Figura 29. MaxPooling + Totalmente Conectada Tanh, Model 2 38

Figura 30. Totalmente conectada Tanh + Totalmente Conectada Softmax 38

Figura 31. Topología del Modelo 2 39

Figura 32. Topología del Modelo 3 40

Figura 33. Ejecución en consola 42

Figura 34. Espectrograma (Manual) 44

Figura 35. Espectrograma (Librosa) 44

Figura 36. Matriz de confusión modelo 1 45

Figura 37. Matriz de confusión modelo 2 46

Figura 38. Matriz de confusión modelo 3 46

Figura 39. Épocas vs. Exactitud modelo 1 47

Figura 40. Épocas vs. Pérdida modelo 1 47

Figura 41. Épocas vs. Exactitud modelo 2 47

Figura 42. Épocas vs. Pérdida modelo 2 48

Figura 43. Épocas vs. Exactitud modelo 3 48

Figura 44. Épocas vs. Pérdida modelo 3 48

Figura 45. Matriz de confusión validación 49

RESUMEN

Este proyecto tiene como principal objetivo encontrar los métodos que permitan diseñar un sistema de reconocimiento de locutor por voz, basándose en conceptos de inteligencia artificial (IA) y teorías de procesamiento digital de señales (DSP). Para esto se interactúa con distintos procesos relacionados a IA y DSP, que realicen el respectivo filtrado, adecuación de señal y reconocimiento de patrones. El sistema se ejecutará y contará con una interacción simple, lo que corresponde a escuchar al locutor e identificar si este pertenece o no a un registro previo en la base de datos.

PALABRAS CLAVE: procesamiento digital de señales, inteligencia artificial, reconocimiento de voz

ABSTRACT

The aim of this project is to find the methods that drive to design a speaker recognition system by voice, based on artificial intelligence (AI) and digital signals processing (DSP). It is needed to interact with a series of process related to AI and DSP, so the system will do the respective filtering, signal processing and recognition patterns correctly. The system will be executed, and it will have a unique interaction, then it will listen to the speaker and identify if he/she is registered into the database.

PALABRAS CLAVE: digital signals processing, artificial intelligence, voice recognition

1. INTRODUCCIÓN

El ser humano es único e irrepetible. Aunque existan dos personas similares físicamente, aunque unos actúen igual que otros, aunque nazcan gemelos o mellizos, cada ser humano cuenta con características que logran diferenciarlo de los demás.

El rostro y su termograma, las huellas dactilares, la geometría de la mano y sus venas, el iris, los patrones de la retina, la voz y la firma son algunos de los parámetros más conocidos para la caracterización de la dimensión física de cada ser humano. Gracias al avance de la tecnología se ha logrado el aprovechamiento de estas particularidades para ser utilizadas en distintas aplicaciones, de las cuales la seguridad e identificación se convierten en los horizontes más importantes de su estudio.

Es la biometría la parte de la biología que estudia en forma cuantitativa la variabilidad individual de los seres vivos utilizando métodos estadísticos. Con ayuda de los mencionados avances tecnológicos, la biometría hace una serie de medidas de características específicas que permiten la identificación de personas. Para esto, se normaliza el uso de dispositivos electrónicos que las recibe, y así lograr identificar las características específicas de cada persona al comparar con un patrón conocido, que se encuentra almacenado.

Con la convergencia de estos conceptos y apoyándose en teorías de procesamiento digital de señales, acuñando también adelantos en temas de inteligencia artificial, y centrándose específicamente en la subrama denominada aprendizaje de máquina, se encontró a través diferentes métodos de diseño una implementación cercana a un sistema de reconocimiento de locutor por voz, el cual según un estudio biométrico, entiende y categoriza al locutor, permitiendo establecer dinámicas de uso como seguridad por voz, definiendo así un sistema seguro y versátil.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e implementar un sistema de reconocimiento de voz basado en inteligencia artificial y procesamiento digital de señales, permitiendo la identificación del locutor como usuario registrado.

2.2 OBJETIVOS ESPECÍFICOS

- Establecer un método de obtención de datos que permita disminuir agentes generadores de ruido para lograr archivos de audio utilizables.
- Estudiar las teorías de procesamiento digital de señales que permitan extraer las características principales de la voz del locutor.
- Diseñar un modelo de inteligencia artificial en el lenguaje de programación Python que permita identificar al usuario según las características extraídas de sus señales de voz.
- Establecer un flujo de interacción entre los usuarios identificados y el sistema ejecutado desde la terminal de Windows.
- Validar el funcionamiento del sistema mediante pruebas con usuarios registrados y no registrados.

3. PLANTEAMIENTO DEL PROBLEMA

3.1 DEFINICIÓN DEL PROBLEMA

El avance y la proliferación de la información y las nuevas tecnologías ha logrado que aumente el número de personas a su alcance. Esto aporta a la capacidad de manipulación de datos que se puedan presentar por parte de entes o personas que poseen intenciones de generar eventualidades inapropiadas y fraudulentas con su uso, por lo que se planteará un horizonte de trabajo en el cual la seguridad de la información de los usuarios sea la principal necesidad por satisfacer.

Entendiendo esta necesidad, se han realizado aportes en temas de seguridad incursionando en diversos ámbitos como codificación de la información que se transmite a través de internet, contraseñas dinámicas en acceso a portales bancarios y biometría para permitir el ingreso a espacios y sistemas privados, siendo esto último el propósito de este estudio, pues es la voz la señal a utilizar y se considera una característica biométrica importante para sistemas de seguridad. Es necesario asumir la biometría como una práctica basada en los distintos principios de procesamiento de señales, los cuales son esencialmente analógicos y digitales, anotando el procesamiento digital de señales (DSP) como una parte de alta prioridad en este estudio.

Ese mismo aumento de la información ha generado la incursión inminente de alternativas tecnológicas con una alta diversidad de alcances y usos. La inteligencia artificial (IA) es sin duda la tendencia con mayor auge en la actualidad, esto se debe gracias a su versatilidad en cuanto aplicaciones se refiere. La IA cuenta también con una característica de alto impacto, que corresponde a la posibilidad de producir sistemas con comportamientos autónomos y aprendizajes propios. Teniendo esto en cuenta, se plantea la siguiente pregunta de investigación: ¿Es posible desarrollar un sistema de reconocimiento de locutor por voz usando Python, basado en inteligencia artificial y principios del procesamiento digital de señales?

3.2. JUSTIFICACIÓN

Contar con un sistema de seguridad en el que se centralicen las interacciones y la designación de permisos aporta robustez a la aplicación y proporciona además confiabilidad y una alta interacción sistema-usuario, permitiendo también contar con una estructura completamente versátil. Esta es la razón por lo que se buscará diseñar e implementar un sistema de reconocimiento de locutor por voz, el cuál en adelante será llamado “PIA” (Personal Intelligent Assistant), un sistema de seguridad capaz de reconocer las características de la voz de un locutor, concibiendo accesos a los usuarios previamente registrados y denegando a quién no lo esté. PIA es un acercamiento a nuevas tecnologías con el uso de software y lenguajes de programación libres que permite definir los conceptos necesarios para el diseño e implementación de un sistema de reconocimiento por voz.

4. MARCO TEÓRICO

Los asistentes virtuales son una aplicación directa de adelantos en el campo de Inteligencia Artificial (IA)[1]. En la década de 1950, los padres de este campo, Minsky y McCarthy, describieron la inteligencia artificial como cualquier tarea ejecutada por una máquina que pudo haber considerado previamente la inteligencia humana[2], siendo entonces un complemento de múltiples disciplinas, cada una con su respectiva finalidad y procesos internos. Reconocimiento de voz, procesamiento del lenguaje natural, reconocimiento visual, reconocimiento de texto, robótica, domótica, aprendizaje de máquina, aprendizaje profundo e inteligencia cognitiva son algunas de las tecnologías que se encuentra cobijadas bajo el concepto de inteligencia artificial[3].

El método de inteligencia artificial óptimo para el reconocimiento de locutor por voz es el aprendizaje profundo que se inspira en el funcionamiento de las redes neuronales del cerebro humano para procesar información. Entre las clasificaciones del aprendizaje profundo se denotan el aprendizaje no supervisado y el aprendizaje supervisado, dónde el primero permite que el sistema pueda determinar conclusiones por sí solo, mientras el último requiere etiquetas que identifiquen la clase de cada observación, se divide a su vez en dos tipos: regresión y clasificación, dónde la clasificación juega un papel importante para el respectivo sistema de

[1] BOTELHO, Bridget. Virtual assistant (AI assistant). Techtarget [sitio web]. Boston; [Consultado: 5 de enero de 2021]. Disponible en: https://searchcustomerexperience.techtarget.com/definition/virtual-assistant-AI-assistant

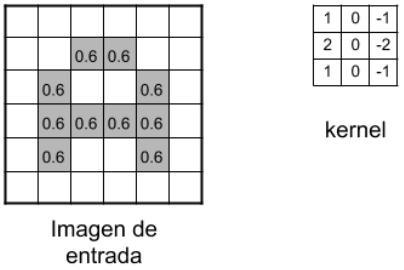
[2] HEATH, Nick. What is Artificial Intelligence (AI)?. zdnet [sitio web]. Londres; [Consultado: 5 de enero de 2021]. Disponible en: https://www.zdnet.com/article/what-is-ai-everything-you-need-to-know-about-artificial-intelligence/

[3] FERNÁNDEZ, Alba. Tecnologías de Inteligencia Artificial y sus categorías [sitio web]. Auraquantic. [Consultado: 5 de enero de 2021]. Disponible en: https://www.auraquantic.com/es/tecnologias-de-inteligencia-artificial-y-sus-categorias/

reconocimiento, pues el algoritmo encuentra diferentes patrones y clasifica los elementos en sus determinados grupos [4].
Para efectos de clasificación existen diferentes tipos de redes neuronales, siendo la Red Neuronal Convolucional uno de las más populares, que recibe su nombre de la operación matemática lineal entre matrices llamada convolución. Una red neuronal convolucional puede contar con múltiples capas, como capas de convolución, de pooling y totalmente conectadas, lo que le permite contar con un excelente desempeño en aplicaciones que se basen en imágenes como data [5].

En una Red Neuronal Convolucional, su proceso distintivo ante otros tipos de redes neuronales es el definido por su nombre, dónde las convoluciones consisten en tomar grupos de pixeles cercanos de una imagen de entrada e ir obteniendo el producto escalar al multiplicarlos con una matriz llamada kernel, ilustrada en la Figura 1, siendo esta el resultando de esta operación la entrada de las siguientes capas ocultas.

Figura 1. Imagen de entrada y kernel



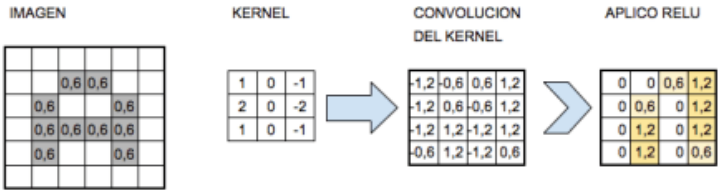
Una vez obtenida la matriz de convolución, se implementa una función de activación cuya principal característica es poder propagar la salida de los nodos de una capa hacia la siguiente capa. Son funciones que producen la activación de la neurona y

[4] ZAMBRANO, Juan. ¿Aprendizaje supervisado o no supervisado? Conoce sus diferencias dentro del Machine Learning y la Automatización Inteligente. Medium. [Consultado: 5 de enero de 2021]. Disponible en: https://medium.com/@juanzambrano/aprendizaje-supervisado-o-no-supervisado-39ccf1fd6e7b

[5] THE INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY 2017 [en línea]. En (3: 21-29 agosto, 2017: Antalya, Turquia). Memorias de International Conference on Engineering and Technology. Antalya. IARES.net. [Consultado: enero 6 de 2021]. Disponible en: https://www.researchgate.net/profile/Saad-Albawi/publication/319253577_Understanding_of_a_Convolutional_Neural_Network/links/5ad26025458515c60f51dbf9/Understanding-of-a-Convolutional-Neural-Network.pdf

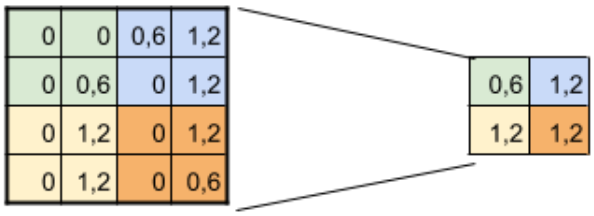
permiten incorporar el modelado de datos de entrada no lineales a la red. La función de activación más utilizada para este tipo de redes neuronales se llama ReLu que cuenta como salida únicamente los valores positivos de entrada, como se identifica en la Figura 2, dónde se identifica el mapa de detención de características. Aunque existe una función de activación de será implementada a lo largo de este proyecto, denominada Tanh, que a comparación de ReLu, sus valores de salida se encuentran en un rango entre -1 y 1.

Figura 2. Mapa de detención de características



Con los mapas de detención de características se puede lograr identificar los patrones que se requieran, pero existe una capa que permite disminuir el tamaño y peso del mapa, a esto se le llama capa de muestreo, dónde se toman las muestras más representativas antes de entrar a la siguiente capa. El muestreo más utilizado es Max-Pooling, en el que se recorre una matriz de tamaño nxm por el mapa de detención de características de izquierda a derecha, de arriba hacia abajo, obteniendo los máximos valores de cada submatriz correspondiente a su tamaño, tal como se puede identificar en la Figura 3.

Figura 3. Muestreo por Max-Pooling



La convolución y el muestreo puede realizarse hasta n veces según se identifique necesario, y una vez obtenida la capa muestreada final se utiliza como entrada para una capa de neuronas totalmente conectadas y finalmente a una capa de salida proveniente de una función llamada softmax, que genera la identificación de pesos por las categorías a clasificas, así una vez ingresado una imagen podrá identificar a cuál pertenece.

Existen conceptos que permiten mejorar el rendimiento de un modelo de inteligencia artificial como las funciones de pérdida y sus respectivos algoritmos de optimización, contando con Perdida de Entropía Categórica Cruzada como la función de pérdida de referencia, la cual calcula la puntuación que penaliza la probabilidad debido a la distancia entre el valor de salida y el valor esperado, dónde el modelo con probabilidades precisas contendrá una pérdida logarítmica o entropía cruzada de 0,0. Esto indica que la pérdida de entropía cruzada es mínima, y los valores más pequeños representarán un buen modelo en lugar de los más grandes.[6] Finalmente, contando con Adam (Adaptative moment estimation) como algoritmo de optimización que permite actualizar los pesos de red de forma iterativa en función de los datos de entrenamiento, logrando converger a un resultante de entrenamiento eficiente en una menor cantidad de épocas.

Lo más cercano al comportamiento del cerebro humano por parte de una máquina se denomina Red Neuronal Profunda, que se caracteriza y a su vez diferencia de una Red Neuronal, por contar además de una capa de entrada y otra de salida, con múltiples capas ocultas, que es dónde ocurren los procesos. Gracias a estas características, una Red Neuronal Profunda puede reconocer comandos de voz, sonido y gráficos, realizar predicción, concebir pensamiento creativo y analizar [7].

Para que un algoritmo de inteligencia artificial de reconocimiento de voz pueda ser entrenado correctamente, los datos a capturar deben ser en formato de imagen; por lo que la fuente, originalmente en formato de audio, debe ser procesada digitalmente para convertirse en un espectrograma, que corresponde en este caso al formato de imagen que es requerido como dato. Para lograrlo, es necesario ejecutar teoremas de Procesamiento Digital de Señales, dónde se manipulan las señales, en este caso de audio, con un fin en específico. Con el uso de diversos métodos y teoremas que permiten trabajar con audio o de vídeo, el Procesamiento Digital de Señales puede a través de algoritmos, detectar qué información de la fuente puede ser separada de cualquier ruido indeseado[8].

---

[6] Pérdida de registro, DataScience [sitio web]; [Consultado: 16 de febrero de 2022]. Disponible en: https://datascience.eu/es/programacion/perdida-de-registro/

[7] KYRYKOVYCH, Anastasia. What is a Deep Neural Network. KDNudggets [sitio web]. Emiratos Árabes Unidos; [Consultado: 6 de enero de 2021]. Disponible en: https://www.kdnuggets.com/2020/02/deep-neural-networks.html

[8] Understanding Audio: What is DSP?. Yamaha [en línea]. Japón; [Consultado: 7 de enero de 2021]. Disponible en: https://uc.yamaha.com/insights/blog/2018/november/understanding-audio-what-is-dsp/

Teniendo en cuenta que las máquinas deben recibir instrucciones claras en cada proceso a realizar, el entendimiento de los audios debe darse en una escala que corresponda a la percepción no lineal por parte del oído humano, lo que se logra al incluir procesamientos de la señal basados en la escala de Mel, al ser más discriminativo a frecuencias bajas y menor a frecuencias altas, logrando esto al hacer uso de una serie de coeficientes, los Coeficientes Cepstrales de Mel, que permiten la representación del habla de forma que una maquina logre percibirla de manera similar a la percepción auditiva humana.

Al hacer uso de las escalas de Mel se implementa un "Bancos de Filtros" que al ser aplicados al espectro de potencias de la señal se puede obtener un gráfico denominado espectrograma, lo que permite analizar de forma visual el comportamiento de la señal. Siendo un espectrograma la imagen del sonido que permite determinar de manera visual las densidades frecuenciales y en qué momento ocurren [9]. Lo que corresponde a las características únicas de la voz de una persona traducidas a la representación que una máquina pueda interpretar, y posteriormente identificar.

Tras contar con los audios procesados como imágenes, el diseño de los algoritmos de inteligencia artificial se realiza usualmente en lenguaje de programación libres, siendo Python el definido por defecto para este ejercicio, utilizando una serie de librerías que le permita hacer manipulación de los datos. Siendo Pyaudio una librería que logra fácilmente reproducir y grabar audios; además de Librosa, que permite el análisis de audio y música. Entre las funciones con las que cuenta Librosa están las operaciones en audio, cálculo de espectrograma y conversión tiempo – frecuencia, contando también con Scikit-learn, definido como "Machine Learning en Python", esta herramienta permite realizar diversas aplicaciones de aprendizaje de máquina como clasificación, regresión, clustering, reducción dimensional, selección de modelos y preprocesamiento. Finalmente, TensorFlow la plataforma de código abierto desarrollada por Google, que cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos, que permite realizar tareas con el propósito principal de entrenar redes neuronales profundas.

[9] Understanding Spectrograms. Izotope [en línea]; [Consultado: 7 de enero de 2021]. Disponible en: https://www.izotope.com/en/learn/understanding-spectrograms.html

5. DESARROLLO DEL PROYECTO

El desarrollo de PIA diversas etapas que comprenden los procesos desde adquisición de la información, hasta la creación y ejecución del script final con el cual el usuario interactuará.

5.1 Adquisición de audios

Con el uso de las librearías PyAudio y Wave de Python, se diseñó una script inicial que permite la grabación de un audio de dos canales en formato .wav de 3 segundos de duración, a una tasa de muestreo de 44.1 kbps, con un formato de 16 bits por muestra, y un tamaño de ventana de 1024 bytes.

El formato .wav tiene como fundamento la conservación de información que podría perderse al utilizar formatos de compresión como .mp3[10]. Por otro lado, 44.1 kHz es definida como la tasa de muestreo por excelencia para la grabación de audios al cumplir con el teorema de Nyquist[11], obteniendo los posibles 22.05 kHz que cubren todas las frecuencias percibidas por una persona común.

Una vez se logró grabar y crear un audio, se definió un segundo script que permite, mediante el mismo procedimiento del script inicial, grabar una serie de 'n' audios, pudiendo definir el nombre del locutor, duración del audio, cantidad de audios a grabar y la ruta en la que será guardada la lista final de audios por locutor.

[10] What is a WAV file?. Fileformat [en línea]; [Consultado: 7 de enero de 2021]. Disponible en: https://docs.fileformat.com/audio/wav/

[11] Nyquist and Shannon's Sampling Theorems. National Instruments [en línea]. [Consultado: 7 de enero de 2021]. Disponible en: https://zone.ni.com/reference/en-XX/help/370524V-01/siggenhelp/fund_nyquist_and_shannon_theorems/

5.2. Filtrado y adaptación de audios

En la etapa de estudio del procesamiento y manipulación de los datos, se utilizó el entorno gratuito de Google, Google Colaboratory, obteniendo como primera instancia las características de las unidades de procesamiento brindadas por la plataforma: Tesla P100-PCIE-16GB (unidad de procesamiento gráfico).

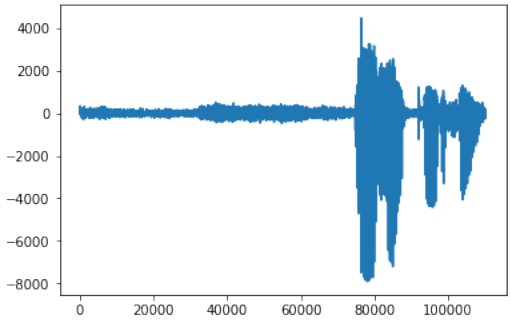
Es importante el uso de librerías que permitan la lectura de archivos de audio, realizar cálculos numéricos, graficar y procesar datos, por lo cual se utilizó Scipy.io.wavfile (en adelante Wave), para la lectura de audios .wav; Numpy, para los respectivos cálculos; Matplotlib, para realizar gráficos; OS, para navegar entre directorios del sistema operativo; Pandas, para el procesamiento de datos.

Un análisis temporal y frecuencial en los audios es necesario para poder aprovechar correctamente los datos recolectados y generar la estructura para obtener resultados fiables, lo que corresponde a un sistema de reconocimiento con un alto porcentaje de acierto, con un uso de recursos reducido.

5.2.1 Análisis temporal

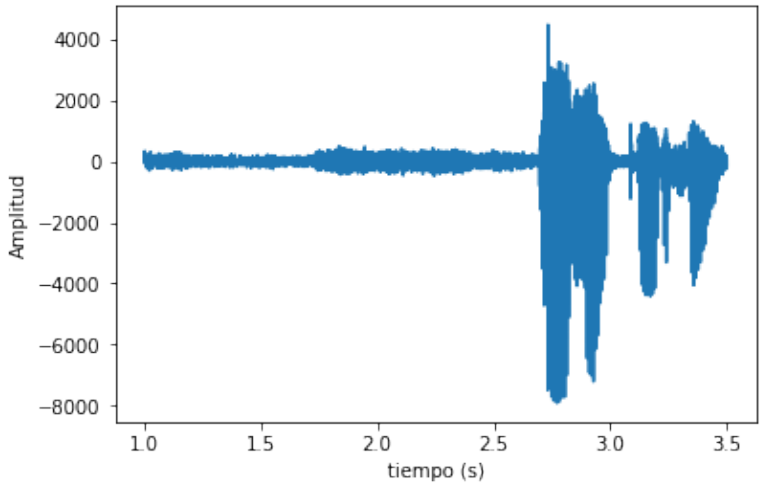
Con el uso de la librería Wave, se trató un audio escogido de forma aleatoria y se obtuvo un arreglo de dos dimensiones con sus muestras y 44100 Hz como tasa de muestreo, correspondiente a la definida en su grabación. Se entiende el arreglo de muestras de dos dimensiones como la representación de un audio estéreo, dónde cada dimensión corresponde a un canal de los 2 definidos en un audio basado en estereofonía. Para procesar de manera práctica el audio se separa uno de los canales, como se muestra en la Figura 4.

Figura 4. Señal de audio del canal izquierdo (Muestra vs Amplitud)



Conociendo la tasa de muestreo, se puede calcular la representación en función del tiempo, dónde el diferencial del tiempo corresponde a la relación 1/muestreo, como se muestra en la Figura 5.

Figura 5. Señal de audio canal izquierdo (Tiempo vs Amplitud)



Una vez entendida la estructura de un audio y obtenida su representación gráfica, se diseñó una función que permite acumular en un par de arreglos los audios que se encuentran almacenados en un directorio de Google Drive en específico. Dónde el primer arreglo posee lo que será denominado como “Etiquetas”, que corresponde al nombre de la subcarpeta que a su vez es definido por el nombre del respectivo locutor. El segundo arreglo contará con los audios obtenidos mediante la implementación del método *.read* de la librería Wave, que permite obtener la información del audio definida en arreglo de dos dimensiones con sus muestras y 44100.

Para efectos de prueba y entendimiento de la información recolectada, se construyó un script que permite obtener un audio aleatorio por cada locutor registrado hasta el momento, representados en las figuras Figura 6 a Figura 10. Este proceso se realiza con el fin de observar las características del audio según los ambientes en que fueron grabados.

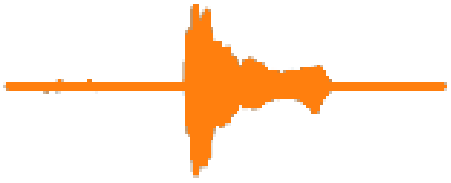
Figura 6. Audio aleatorio 1 (155232, 44100) (Muestra vs Amplitud)

Figura 7. Audio aleatorio 2 (132480, 44100) (Muestra vs Amplitud)

Figura 8. Audio aleatorio 3 (161176, 48000) (Muestra vs Amplitud)

Figura 9. Audio aleatorio 4 (148480, 44100) (Muestra vs Amplitud)

Figura 10. Audio aleatorio 5 (156672, 44100) (Muestra vs Amplitud)

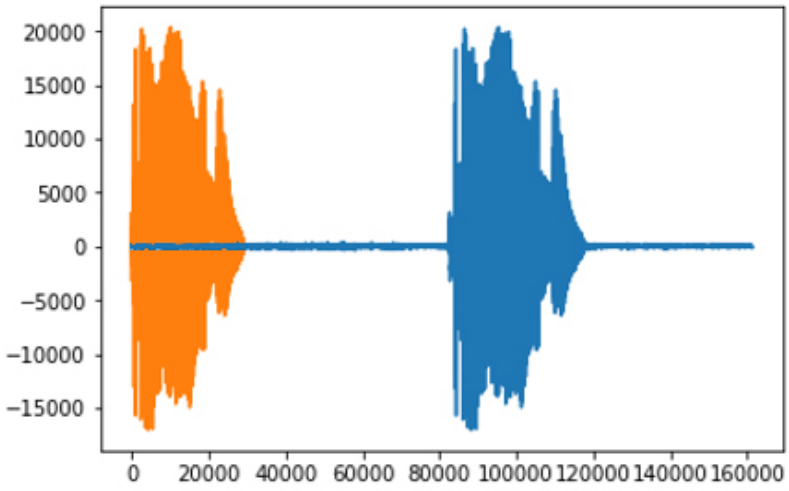


La forma y tamaño de los audios corresponde a la libertad con la que contaron los locutores a la hora de grabarlos. Cada locutor obtuvo sus audios en distintos lugares y pronunciaron la frase “Hola PIA” con una longitud temporal indefinida, por lo que se cuenta con datos obtenidos en ambientes no controlados.

Posteriormente, se agregan los datos de los canales por separado a un dataframe que posee una estructura definida por 5 columnas: Etiquetas, Amplitudes, Muestreo, Canal 1, Canal 2; siendo cada fila un audio. Una vez agregados todos los parámetros al dataframe, se exporta en formato .csv para su posterior uso.

Una vez todos los audios son separados en función de sus canales, se reconoció la existencia en común de silencios y posible información correspondiente a ruido, por lo que es necesario hacer un filtrado en amplitud. Para esto se tomó como umbral inicial |10|, denotando que este rango aún permite el paso de ruidos, por lo que de manera iterativa, aumentando en una ración de 10 por iteración, se comprobó que es necesario eliminar todas las muestras con amplitudes menores a 500 y mayores a -500, así mismo permitiendo conservar la mayor información posible, como se evidencia en la Figura 11.

Figura 1. Audio original (azul), audio filtrado (naranja) (Muestra vs Amplitud)



Obteniendo de esta manera, información con aproximadamente 25% de la cantidad de muestras de las que se contaba originalmente. Generando así una representación fiel, con una disminución de tamaño y peso importante a la hora de hacer el posterior tratamiento de datos en los modelos de reconocimiento.

5.2.2 Análisis frecuencial

Entender el comportamiento de una señal en el dominio del tiempo no es suficiente al tratarse de audio, por lo que se estudió los efectos que surgen a partir un análisis frecuencial haciendo uso de una serie de teoremas y transformadas.

Se escogió un audio de manera aleatoria para observar su estructura frecuencial, en primera instancia se observa en el dominio temporal en la Figura 12, con ayuda de la librería scipy se obtuvo la Transformada Rápida de Fourier, y así poder ver las amplitudes de los coeficientes frecuenciales del audio en función de sus muestras representado en la Figura 13.

Figura 12. Audio en dominio del tiempo (Muestra vs Amplitud)

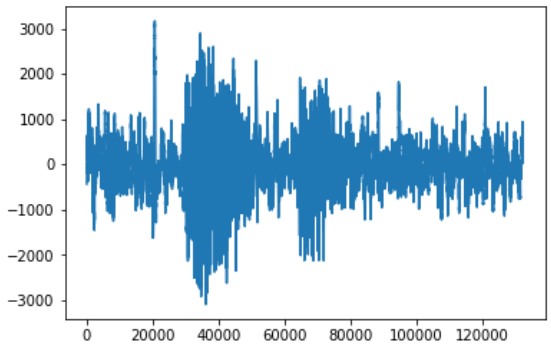
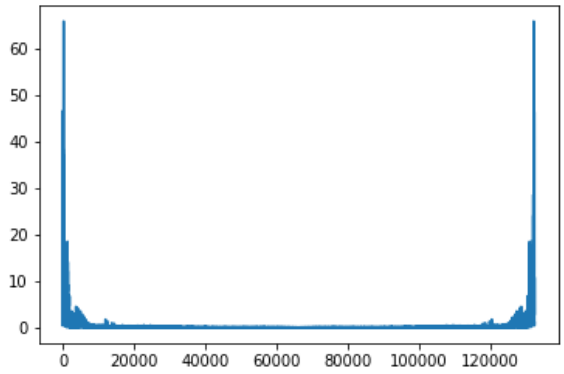


Figura 13. Audio en dominio de la frecuencia (Muestra vs Frecuencia)



Cuando la señal medida es periódica y un número entero de periodos llena el intervalo de tiempo de la adquisición, la Transformada Rápida de Fourier (FFT) resulta bien. Por lo que, para señales de propiedades aleatorias, no periódicas y con un número no entero de periodos, es necesario hacer una aproximación tomando pequeños trazos de la señal como producto de una función denominada: ventana[12].

[12] KAISER, Gerald. Windowed Fourier Transforms. Springer [sitio web]. Boston; [Consultado: 6 de enero de 2021]. Disponible en: https://link.springer.com/chapter/10.1007/978-0-8176-8111-1_2

Debido a que las señales obtenidas de los audios incluyen interferencia a causa de factores como el ruido ambiental, se hace uso de una ventana con un rango alto de caída del lóbulo lateral, lo que corresponde a una ventana Hamming.

Para simular lo que será la estructura de las ventanas a implementar, se crearon dos arreglos a partir de un audio filtrado por amplitud, dónde el primer arreglo corresponde a una ventana con punto de partida en la muestra 0, y una segunda ventana arrancando en la muestra 512, ambas con un tamaño de 1024 como se muestra en la Figura 14.

En la Figura 15 se muestra cómo se aplica una ventana Hamming al primer tramo de 1024 muestras.

Figura 14. Ventana inicial (azul) y ventana final (naranja) (Muestra vs Amplitud)

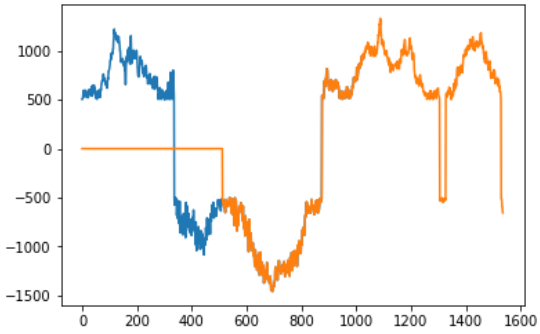
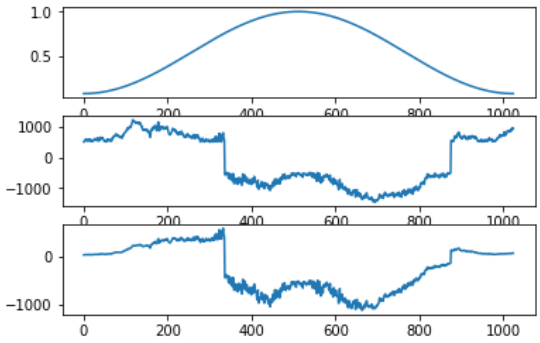
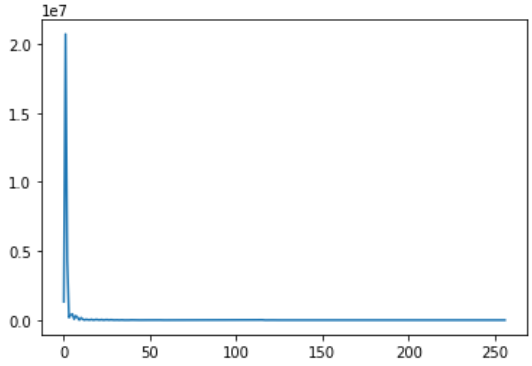


Figura 15. a) Ventana de Hamming b) Tramo de la señal de 0 a 1024 muestras c) tramo enventanado con Hamming



Una vez verificando la suavizado de la señal después de aplicar la ventana Hamming se adquiere su Transformada Rápida de Fourier (STFT) para posteriormente obtener el arreglo de potencias por coeficiente, que se presenta en la Figura 16, lo que será utilizado para la etapa correspondiente a filtrados mediante los bancos de filtros de Mel, pues el uso de STFT permite localizar eventos puntuales en el determinado momento en que ocurrió, teniendo entonces información detallada que permita caracterizar la voz de un locutor, pues se podrán entender patrones en cuanto a los niveles de amplitudes y frecuencias de su voz, y en qué tiempos tienen ocurrencia, para así correlacionar y dar detalle al algoritmo de inteligencia artificial.

Figura 16. Coeficientes vs. Potencia



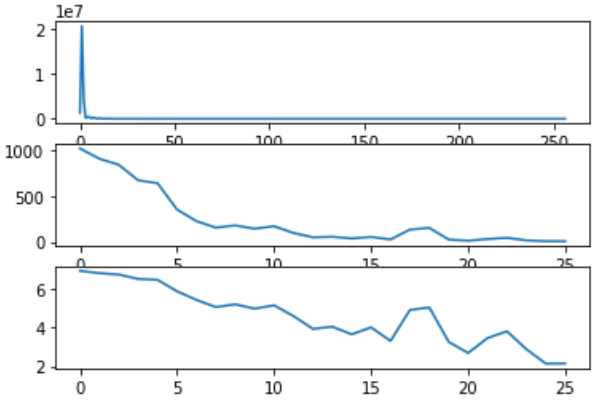
5.2.2.1 Filtrado

Para el filtrado de la señal, se optó por utilizar los bancos de filtros de Mel, que permiten acercar de manera asertiva un rango de frecuencias que serán reconocidas por las máquinas tal como un humano. Para esto, se cuenta con una serie de funciones que permiten tanto transformar una frecuencia de muestreo en escala de frecuencias de Mel, como viceversa. Una tercera función, que retorna un banco de filtros de Mel a partir de parámetros como la cantidad de filtros a crear, las frecuencias máximas y mínimas (7500 y 300 respectivamente, basado en la frecuencia de habla humana), el número de coeficientes a filtrar y la frecuencia de muestreo, evidenciado en la Figura 17.

Es usual el uso de la Transformada Discreta de Coseno (DCT) en algoritmos de tratamiento de imagen y audio, pues permite de manera directa la recuperación de la totalidad de la información, al ser repres9entada únicamente en una suma de cosenos, eliminando así el factor imaginario que aporta el uso de la Transformada

Discreta de Fourier, adicionando la escala de frecuencias de Mel con lo que se consideran las frecuencias sensibles para los sistemas sensoriales humanos. Como se muestra en la Figura 1, la señal filtrada se representa en escala logarítmica con la ayuda del método dct del paquete fft de la librería scipy, se obtiene su DCT[13] (Direct Cosine Transform), almacenando los 13 primeros coeficientes correspondientes a los Coeficientes Cepstrales de las Frecuencias de Mel[14], de los cuales se identificó mayor cantidad de características importantes de la voz del locutor, lo que permitirá generar su espectrograma, para lo que será necesario entender que el proceso de análisis en frecuencia y filtrado realizado hasta el momento corresponde a la ventana inicial de un único audio escogido de manera aleatoria, por lo que se diseñó un ciclo que permite recorrer todo el audio con ventanas Hamming de 512 muestras de ancho, lo que generará un arreglo bidimensional con *n*x*m,* dónde *n* corresponde a la cantidad de ventanas recorridas según la duración del audio y *m* los 13 Coeficientes Cepstrales de las Frecuencias de Mel, tal como se representa en la Figura 19.

Figura 17. a) Muestras vs Potencia 2) Potencia con 26 filtros de Mel 3) Potencia con 26 filtros de Mel (Escala logarítmica)



[13] Discrete Cosine Transform [en línea]. Paperswithcode. [Consultado: 7 de enero de 2021]. Disponible en: https://paperswithcode.com/method/discrete-cosine-transform

[14] FAYEK, Haytham. Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between [sitio web]. Seattle; [Consultado: 8 de enero de 2021]. Disponible en: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

Figura 18. a) Potencia con 26 filtros de Mel (Escala logarítmica) b) Transformada Discreta de Coseno

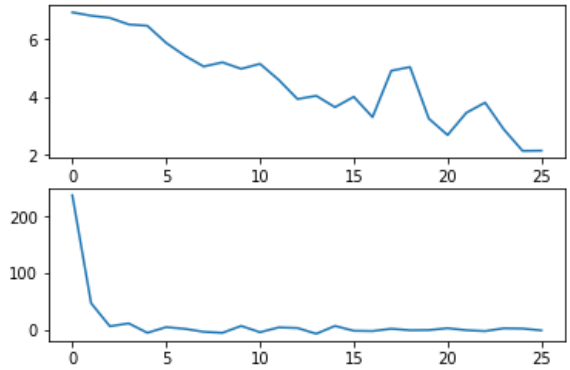
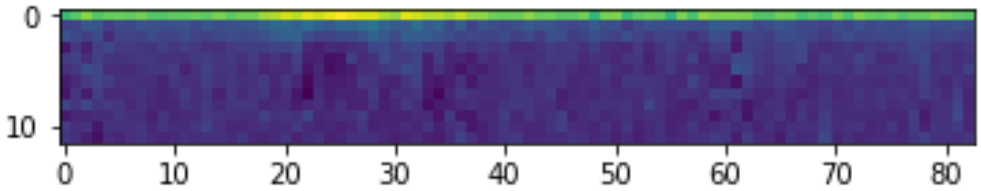


Figura 19. Espectrograma de audio de prueba



5.2.2.2 Análisis frecuencial - Síntesis

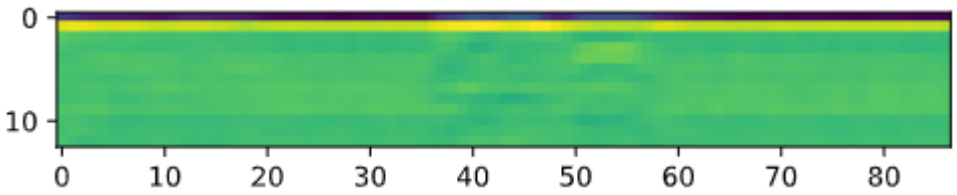
Existen librerías que aportan procesos y resultados previamente comprobados y verificados. Para el manejo de audios en Python, la librería *Librosa* cumple con estas características, pues reúne en sus métodos lo que previamente se realizó de manera independiente y experimental[15], su uso permitirá hacer la respectiva comparativa entre el proceso diseñado desde la grabación de los audios, hasta la obtención de su respectivo espectrograma, así determinar qué método se continuará utilizando, entendiendo el consumo computacional y la fidelidad de la información, como factores importantes a tener en cuenta.

Para generar una estructura de prueba, se aplicó a un audio aleatorio el método *.load* de Librosa que permite cargarlo y obtener la misma información que arroja la librería *wave* con su método *.read*, un arreglo con sus respectivas amplitudes normalizadas de -1 a 1, y la frecuencia de muestreo del audio.

[15] Librosa. Librosa [en línea]; [Consultado: 8 de enero de 2021]. Disponible en: https://librosa.org/

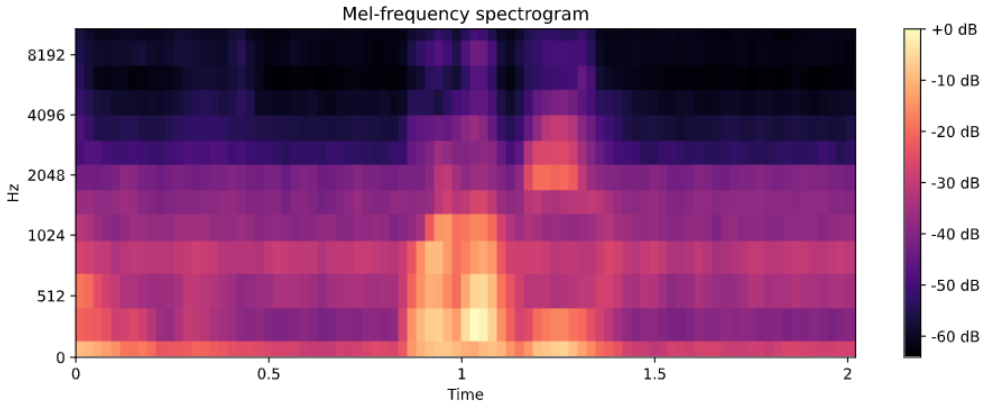
Los procesos de recorrer un audio por ventanas, obtener las potencias de los coeficientes de la Transformada Rápida de Fouirer ,aplicar filtros de Mel y con Transformada Directa de Coseno obtener los 13 primeros Coeficientes Cepstrales de las Frecuencias de Mel por cada ventana, se condensan en la función mfcc del método features de Librosa, teniendo como resultado un arreglo bidimensional con *nxm,* dónde *n* corresponde a la cantidad de ventanas recorridas según la duración del audio y *m* los 13 Coeficientes Cepstrales de las Frecuencias de Mel, como se representa en la Figura 20.

Figura 20. Espectrograma obtenido con Librosa



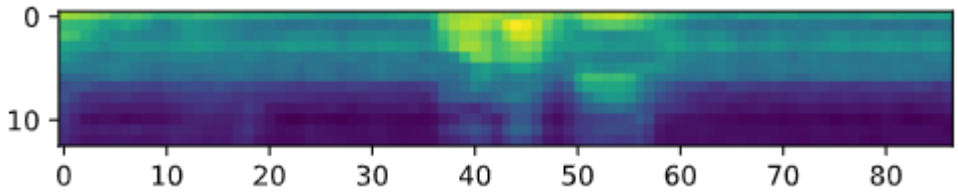
En el análisis frecuencial realizado de manera experimental, se aplicó el banco de filtros de Mel a la señal del audio seleccionado en función de sus potencias, por lo que previo a la generación del espectrograma de la Figura 17, se aplicó el método *stft* de Librosa se generó la Transformada de Fourier de Tiempo Reducido (STFT), y se obtuvo como resultado las potencias mostradas en la Figura 21.

Figura 21. Espectrograma de coeficientes



Definiendo que la representación a interpretar en adelante está en función del mapeo de potencias a decibeles, lo que se logra con el método *power_to_db* de Librosa para obtener un arreglo bidimensional con *n*x*m,* dónde *n* corresponde a la cantidad de ventanas recorridas según la duración del audio y *m* los 13 Coeficientes Cepstrales de las Frecuencias de Mel, tal cómo se representa en la Figura 22.

Figura 22. Espectrograma de potencias



Una vez estructurado el análisis frecuencial y filtrado de un único audio utilizando Librosa, se diseñó un script que recorre todos los audios cargados y les aplica cada una de las etapas anteriormente mencionadas, a su vez se etiqueta cada audio con su respectivo locutor, dónde un cada locutor corresponde a un número entero de 0 a 2, siendo 0 y 1 los locutores con acceso y 2 los locutores no autorizados, lo que facilitará su reconocimiento por parte de los modelos a aplicar.

5.2.3 Diseño de modelos

Con el dataset de imágenes se dividió en conjunto de entrenamiento y prueba, dando como resultado dos conjuntos de datos, dónde el 70% de la cantidad total de audios serán utilizados para entrenar el modelo y el 30% restante para probarlo.

TensorFlow utiliza un grafo computacional para realizar las operaciones referentes al modelo implementado, por lo que es necesario reiniciarlo para que este se encuentre libre de cualquier característica referente a un modelo anteriormente ejecutado. Para esto se hizo uso del método *ops* de la librería *TensorFlow*[16]. En el proceso de diseño de modelos de inteligencia artificial, es importante entender cada una de sus etapas, el por qué se incluyen y cuándo debe ocurrir. Para esto, una secuencia iterativa permite avanzar eliminando errores y aumentar de manera sustancial el parámetro de satisfacción que será la exactitud.

[16] Por qué TensorFlow [en línea]. Tensorflow; [Consultado: 9 de enero de 2021]. Disponible en: https://www.tensorflow.org/

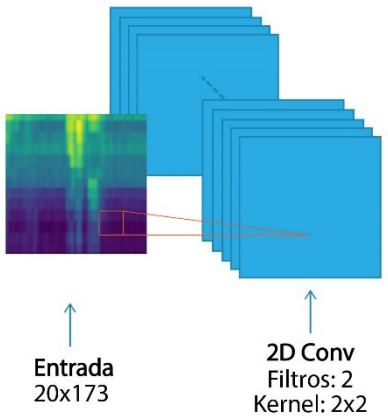
La librería *keras* con su método *Sequential()* permite instanciar un objeto tipo secuencial, al que se irán agregando las distintas capas a utilizar.

En una secuencia de capas convolucionales, de MaxPooling, de aplanado y densas con diversos tipos de activadores, se logrará encontrar un modelo que satisfaga la necesidad de clasificar los audios en los 3 tipos de locutores previamente establecidos, por lo que se implementarán una serie de 3 modelos de inteligencia artificial que permitan ser comparados y finalmente seleccionar el modelo que mejor satisfaga el objetivo de este proyecto.

5.2.3.1 Modelo 1.

El modelo 1 cuenta con una capa de entrada convolucional en 2D con 2 kernels(filtros) de tamaño 2x2 y una función 'relu', en esta capa se define el tamaño de las imágenes con las que se alimentará el modelo, representado en la Figura 23.

Figura 2. Imagen de entrada + Convolución 2D del Modelo 1



Entendiendo que una capa convolución en 2D[17] es una operación entre un kernel y una imagen, dónde el kernel que es una matriz de pesos, y se desplaza a través de la imagen, ejecutando una multiplicación con la parte de la imagen en la que se encuentra, sumando al final los resultados en una salida que será un píxel único.

[17] Intuitively Understanding Convolutions for Deep Learning. TowardsDataScience. [Consultado: 8 de enero de 2021]. Disponible en: https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

Una vez filtrada cada imagen, se agregó una capa de aplanado (Flatten), que permite tomar dicho mapa y convertirlo en un vector unidimensional que podrá ser consumido por la siguiente capa que será una capa totalmente conectada de 5 neuronas con función de activación ‘relu’, esta capa, denominada capa oculta se muestra en la Figura 24, a la cual le procede otra capa totalmente conectada, en esta ocasión con 3 neuronas y una función de activación ’softmax’, la que permite hacer la respectiva clasificación entre las 3 categorías, dónde cada neurona representa cada uno de los locutores, como se ilustra en la Figura 25.

Figura 24. Convolución 2D + Totalmente Conectada ReLU, Modelo 1

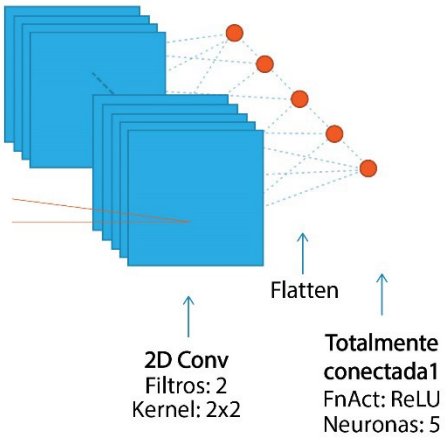
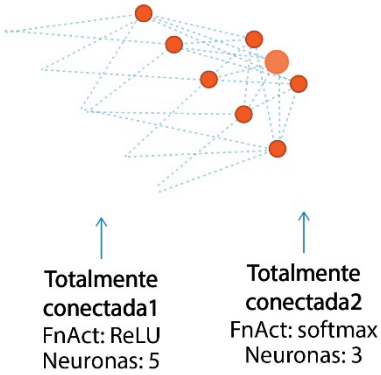


Figura 25. Totalmente conectada ReLU + Totalmente Conectada Softmax, Modelo 1



Finalmente se usa el optimizador ‘adam’ con el que se entrenará la red con una tasa de aprendizaje de 0.0001, dónde un optimizador es la herramienta que actualiza los parámetros de peso para minimizar la función de pérdida, que en este caso es Entropía Categórica Cruzada. Dónde una función de pérdida actúa como guía, indicando al optimizador si este se dirige correctamente al mínimo global.

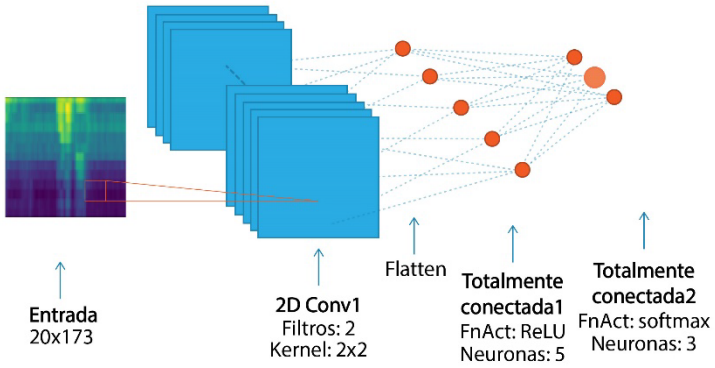
Una vez estructurado el modelo, se estableció exactitud como métrica de evaluación. Con el fin de poder evaluar los resultados de la red a medida que se entrena.

La Tabla 1 permite identificar la arquitectura del modelo 1 que se representa en la Figura 26, teniendo en cuenta cada una de sus capas y la cantidad de parámetros utilizados por cada una de ellas. Con un total de 32.713 parámetros, de los cuales el 100% son parámetros entrenables.

Tabla 1. Arquitectura Modelo 1

Capa	**Forma de salida**	**Número de parámetros**
Convolución 2D	(19, 172 ,2)	10
Aplanado	6536	0
Densa 1	5	32685
Densa 2	3	18

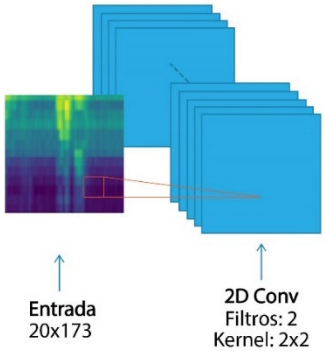
Figura 26. Topología del Modelo 1



5.2.3.2 Modelo 2.

Para el modelo 2 se usó una capa de convolución en 2D con 16 kernels(filtros) de tamaño 2x2 y junto a la función ‘tanh’ como función de activación serán la capa de entrada, dónde se define el tamaño de las imágenes con las que se alimentará el modelo, representado en la Figura 27.

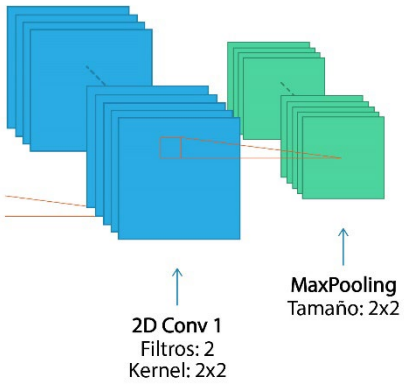
Figura 27. Entrada + Convolución 2D, Modelo 2



Con Tanh como función activación, se tiene una función con un rango de acción acotado entre -1 y 1, lo que permite tener un mayor espectro de apreciación de los espectrogramas, pues se considera la normalización de las amplitudes de cada audio en este mismo rango.

Una tercera capa de convolución 2D con 32 kernels de tamaño 3x3 y función de activación ‘tanh’, seguida por una capa de MaxPooling 2D, que permite reducir la dimensionalidad del mapa de características obtenido en la capa anterior. La capa de MaxPooling funciona haciendo pasar una ventana por la matriz 2D, tomando únicamente el valor máximo que se encuentra en dicha ventana. En esta ocasión estableciendo un tamaño de ventana (pool_size) de 2x2, y un stride de 2. Lo que significa que una ventana de 2x2 se moverá de 2 en 2 a través de la imagen, etapa ilustrada en la Figura 2.

Figura 28. Convolución 2D + MaxPooling, Modelo 2



Finalmente, la Figura 29 muestra una capa de aplanado, la cual será conectada a una capa totalmente conectada de 50 neuronas con función de activación ‘tanh’, a la cual le procede otra capa totalmente conectada con 3 neuronas y una función de activación ’softmax’, para clasificación, como se ve en la Figura 30.

Figura 29. MaxPooling + Totalmente Conectada Tanh, Modelo 2

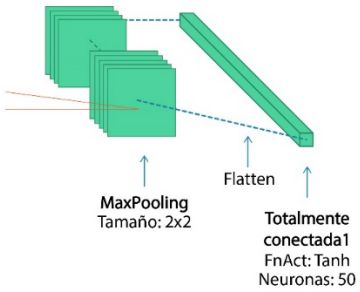
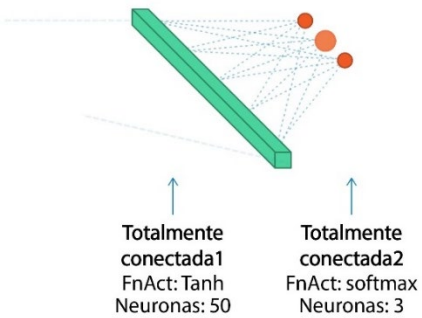


Figura 30. Totalmente conectada Tanh + Totalmente Conectada Softmax



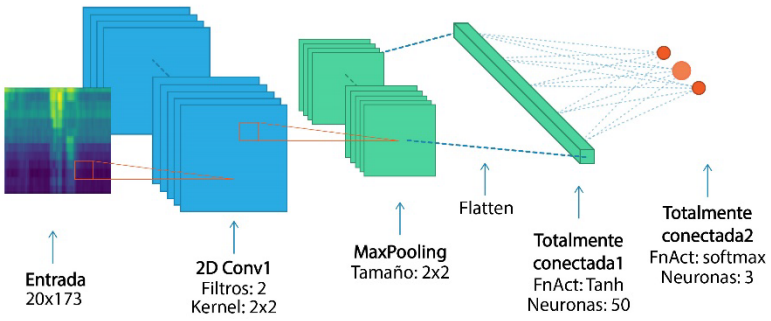
Por último, es agregado un optimizador ‘adam’ con el que se entrenará la red, contando con una taza de aprendizaje de 0.0001. Se estableció exactitud como métrica de evaluación

La Tabla 2 representa la arquitectura del modelo 2 que se muestra en la Figura 3, con un total de 1’092.923 parámetros, de los cuales el 100% son parámetros entrenables.

Tabla 2. Arquitectura Modelo 2

Capa	**Forma de salida**	**Número de parámetros**
Convolución 2D	(19, 172 ,16)	80
Convolución 2D	(17, 170 ,32)	4640
MaxPooling	(8, 85, 32)	0
Aplanado	6536	0
Densa 1	50	1088050
Densa 2	3	18

Figura 31. Topología del Modelo 2



5.2.3.3 Modelo 3.

El tercer modelo inicia su arquitectura con una capa de convolución en 2D con 32 kernels de tamaño 2x2 y una función ‘tanh’.

Una siguiente capa de convolución 2D con 64 kernels de tamaño 2x2 y función de activación ‘tanh’, seguida por una capa de MaxPooling 2D. Posteriormente una nueva capa de convolución 2D con 128 kernels de tamaño 2x2, la cual se conecta

a una capa de aplanado, que servirá de entrada para una siguiente capa, la cuál será densa totalmente conectada de 500 neuronas y función de activación ‘tanh’.

Finalmente, se agregó una serie capas con 300, 200, 100, 50 y 25 neuronas respectivamente, todas con función de activación ‘tanh’, siendo la capa de 25 neuronas procedida por una última capa totalmente conectada con 3 neuronas y una función de activación ’softmax’, para clasificación.

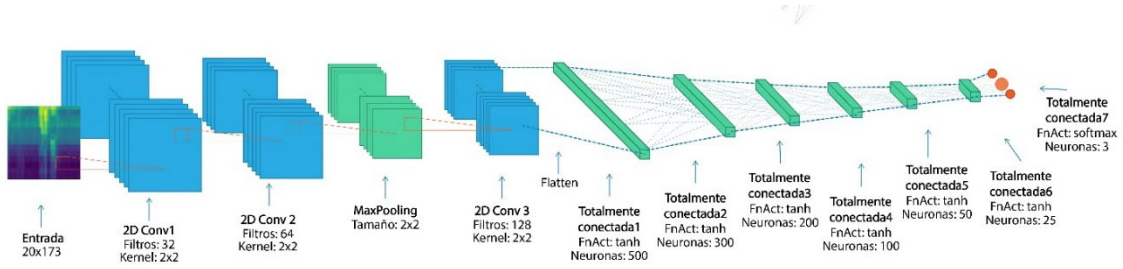
Teniendo a ‘adam’ como optimizador, contando con una taza de aprendizaje de 0.0001. Se estableció exactitud como métrica de evaluación

La Tabla 3 representa la arquitectura del modelo 3 visto en la Figura 32, con un total de 37’921.055 parámetros, de los cuales el 100% son parámetros entrenables.

Tabla 3. Arquitectura Modelo 3

Capa	**Forma de salida**	**Número de parámetros**
Convolución 2D	(19, 172 ,32)	160
Convolución 2D	(17, 170 ,64)	18496
MaxPooling	(8, 85, 64)	0
Convolución 2D	(7, 84 ,128)	32896
Aplanado	75264	0
Densa 1	500	37632500
Densa 2	300	150300
Densa 3	200	60200
Densa 4	100	20100
Densa 5	50	5050
Densa 6	25	1275
Densa 7	3	78

Figura 32. Topología del Modelo 3



5.2.3 Script final y ejecución

Un modelo de inteligencia artificia se ejecuta con la articulación de su arquitectura y sus pesos, almacenados en archivos de extensión .yaml y .h5, respectivamente. Para lograr dar vida a PIA, se diseñó un script que ejecute el modelo.
Importando las librerías de grabación de audio, de análisis, y de ejecución de los modelos, se estructuró una primera función basada en la que fue previamente utilizada para la obtención de los audios:

guardar_n_audios(nombre,duracion, cantidad, ruta, indice_inicio)

dónde:

Nombre: “saludo”
Duración: 2 (en segundos)
Cantidad: 1
Ruta: se especifica según el dispositivo en que se ejecute
Índice_inicio: 0

Permitiendo así guardar y sobrescribir un audio llamado “saludo” en la ruta especificada cada vez que se ejecute el script, en el que se establecen parámetros para las librerías de codificación de audios:

Canales: 2, Muestreo: 44100Hz, Tramos: 1024

Será recomendado decir la frase “Hola PIA”, para que esta logre una mejor interacción.

Una segunda función fue diseñada para la lectura y procesamiento del audio previamente grabado:

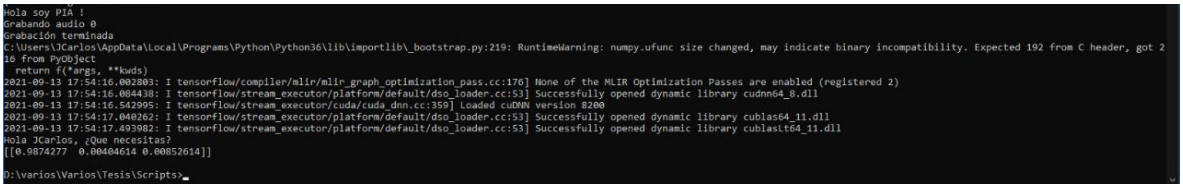
nuevo_audio(ruta)

dónde ruta corresponde a la ubicación y el nombre del audio, que tendrá como base “…/saludo0.wav”.

El audio será guardado en una variable, la cual servirá como parámetro para la ejecución del modelo, utilizando la función .predict de keras. Al contar con un valor de predicción específico, se estructuró un condicional, dónde se determina que en caso de que la predicción retorne un “0”, PIA dirá “Hola Carlos, ¿Qué necesitas?”,

por el contrario, si el resultante es un “1” PIA dirá “Hola Jesús, ¿Qué necesitas?”, y “Hola, no te conozco”, para el caso de tener un “2” como respuesta. Evidenciando la ejecución de todos los proceso en la Figura 33.

Figura 33. Ejecución en consola



5.2.4 Validación

Las validaciones se realizaron con el apoyo de 8 locutores desconocidos y los 2 previamente registrados. Cada locutor realizó 5 grabaciones en ambientes no controlados y 5 en ambientes controlados, definiendo un ambiente controlado como aquél que presenta reducción de ruido externo, sin eco notorio y un micrófono con cancelación de ruido integrado, con lo cual se obtuvo los resultados mostrados en la Tabla 4.

Tabla 4. Validación

Locutor	**Aciertos**	
	Ambiente controlado	Ambiente no controlado
Juan Carlos	5	5
Jesús	5	5
Locutor 3	5	4
Locutor 4	5	5
Locutor 5	4	4
Locutor 6	5	5
Locutor 7	4	4
Locutor 8	5	4
Locutor 9	5	5
Locutor 10	5	5
Porcentaje de acierto	96%	92%

5.3 ANÁLISIS DEL DESARROLLO DEL PROYECTO

Para lograr un correcto desarrollo del proyecto fue importante precisar los métodos de adquisición de lo que serían los datos por tratar y posteriormente utilizar en el diseño de los algoritmos de inteligencia artificial. Razón por la cual, a medida que avanzaban los procesos fue necesario contar con una mayor cantidad de locutores y audios por locutor, tal como se muestra en la tabla 4.

Tabla 5. Audios por locutor

Locutor	**Grabaciones**				**Duración (s)**	**Ambiente**
	Etapa 1.	Etapa 2.	Etapa 2.	Total		
Juan Carlos	20	50	150	170	2	Controlado
Jesús	20	50	150	170	2	Controlado
Locutor 3	20	30	0	50	2 - 3	No Controlado
Locutor 4	20	30	0	50	2 - 3	No Controlado
Locutor 5	30	0	0	30	2	No Controlado
Locutor 6	30	50	0	80	2	No Controlado
Locutor 7	0	0	90	90	2	Controlado
Locutor 8	0	0	100	100	2	Controlado
Locutor 9	0	0	50	50	2	Controlado
Locutor 10	0	0	30	30	2	Controlador

El hecho de contar con audios de tan variables características requirió un tratamiento exhaustivo, iniciando por un análisis temporal hasta llegar a entender sus componentes frecuenciales, reconociendo entonces las ventajas y desventajas que presenta cada procedimiento:

- Análisis temporal:
  - Beneficios:
    - Eliminación de silencios.
    - Disminución de duración.
    - Menor cantidad de operaciones.
  - Desventajas:
    - Datos vistos como un vector unidimensional.
    - Imposibilidad de utilizar algoritmos de extracción de características en imágenes.

- Análisis frecuencial:
    - Beneficios:
        - Permite extracción de características por medio de transformadas.
        - Contar con transformadas permite la generación de espectrogramas.
        - El uso de espectrogramas habilita el uso de algoritmos de análisis de imágenes en extracción de características.
    - Desventajas:
        - Mayor cantidad de operaciones.
        - Mayor gasto computacional.

Una vez entendidos los comportamientos de los audios tanto en el dominio temporal como el dominio frecuencial, se logró comprobar que los resultantes correspondieran a procesos previamente testeados con el uso de herramientas como librerías y teoremas. El resultado obtenido al aplicar la transformada de Fourier, banco de Filtros de Mel y transformada de Coseno de manera manual y hacerlo con el uso de Librosa como librería que compacta estos procesos en un único método se evidencia un comportamiento cercano a un patrón, dónde la mayor información a lo largo de las 86 ventanas se encuentra alrededor del coeficiente 0, lo que permite enunciar que es la información de la frecuencia principal de la voz de un respectivo locutor, a continuación se ilustra en la Figura 3 el espectrograma obtenido de manera manual y en la Figura 35 el resultante de usar Librosa.

Figura 34. Espectrograma (Manual)

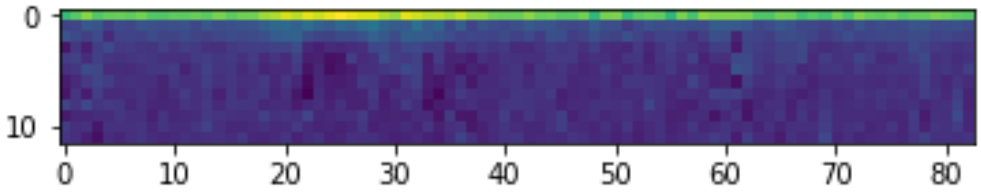
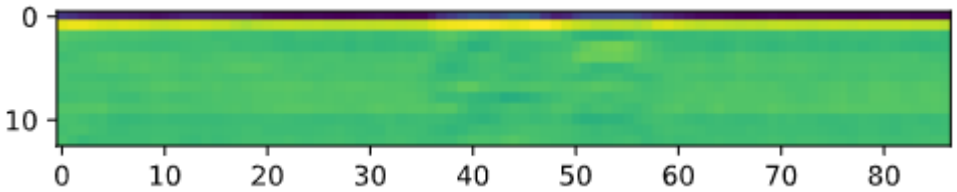


Figura 35. Espectrograma (Librosa)



Tras definir los procesos de generación de espectrogramas, al diseñar la arquitectura de un modelo de inteligencia artificial que permita realizar la respectiva clasificación fue necesario relacionar distintas cantidades y tipos de capas. Al reconocer que los datos que una vez fueron audios en formato .wav, ahora se representan mediante imágenes espectrográficas, el uso de capas de convolución 2D se entendió como la base de los modelos a generar, pues su finalidad es permitir hacer uso de datos de tipo imagen, extrayendo de ellas las características de mayor relevancia.

Tabla 6. Comparativa de arquitecturas

Modelo	**# Capas**				**Número de parámetros**
	Convolución 2D	**MaxPooling**	**Aplanado**	**Densa**	
Modelo 1	2	0	1	2	32.713
Modelo 2	2	1	1	2	1’092.923
Modelo 3	3	1	1	6	37’921.055

Para relacionar los comportamientos de los modelos y establecer métricas comparativas, se utilizó las matrices de confusión, en las que se distribuyen los aciertos y errores cometidos por el modelo al ser ejecutado, distribuyendo los resultados como se muestran en las Figuras 36 a 39 y las Tablas 7 a 9.

Figura 36. Matriz de confusión modelo 1.

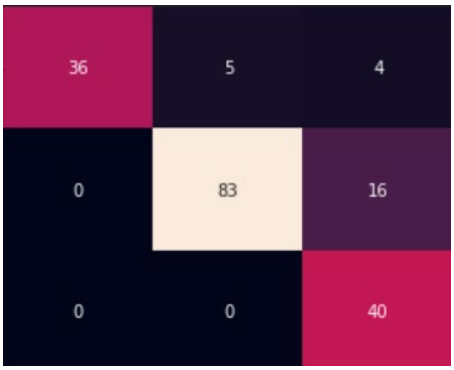


Tabla 7. Matriz de confusión modelo 1

		Categoría			% Aciertos	% Fallos
		0	**1**	**2**		
Categoría	**0**	36	5	4	80	20
	1	0	83	16	86.86	13.13
	2	0	0	40	100	0

Figura 37. Matriz de confusión modelo 2.

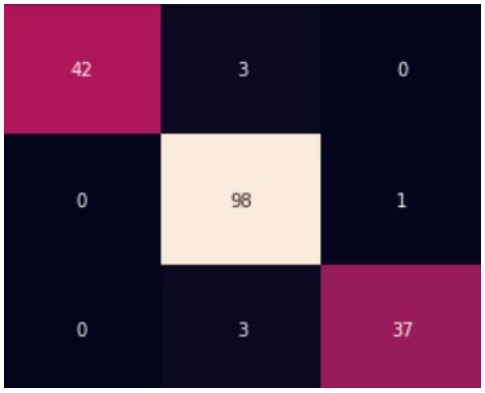


Tabla 8. Matriz de confusión modelo 2

		Categoría			% Aciertos	% Fallos
		0	**1**	**2**		
Categoría	**0**	42	3	0	93.33	6.66
	1	0	98	1	98.98	1.02
	2	0	3	37	92.5	7.5

Figura 38. Matriz de confusión modelo 3.

Tabla 9. Matriz de confusión modelo 2

		Categoría			% Aciertos	% Fallos
		0	**1**	**2**		
Categoría	**0**	42	3	0	93.33	6.66
	1	0	84	1	98.82	1.18
	2	0	8	46	85.18	14.82

Una de las formas en las que se puede medir la eficiencia de entrenamiento de un modelo es obteniendo las relaciones época-exactitud y época-loss. Entendiendo así cuánto tarda cada modelo en lograr la mayor cantidad de precisión con la menor pérdida, como se ilustra en las Figuras 39 a 44.

Figura 39. Épocas vs. Exactitud modelo 1.

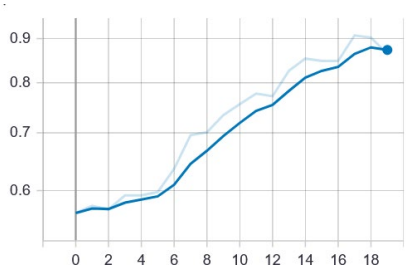


Figura 40. Épocas vs. Pérdida modelo 1.

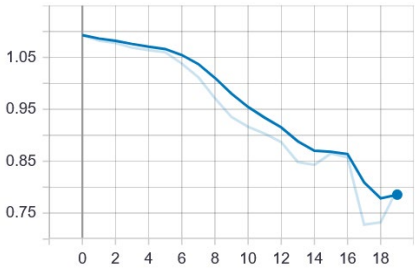


Figura 41. Épocas vs. Exactitud modelo 2.

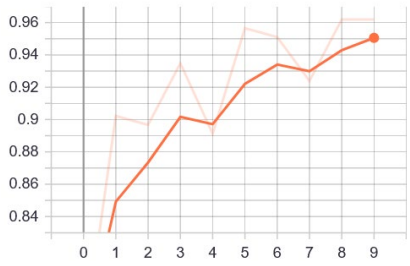


Figura 42. Épocas vs. Pérdida modelo 2.

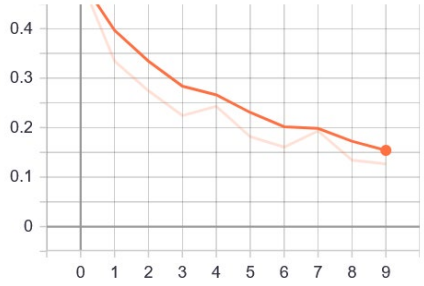


Figura 43. Épocas vs. Exactitud modelo 3.

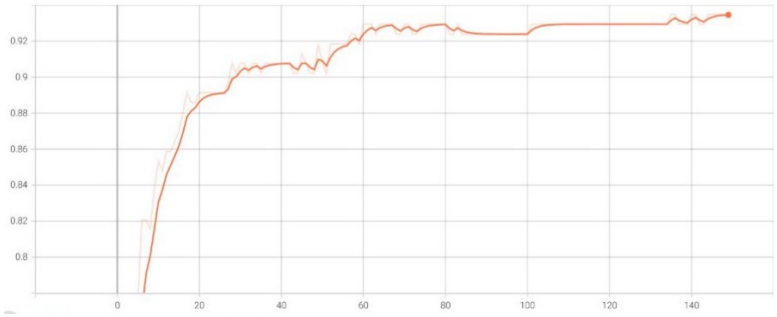
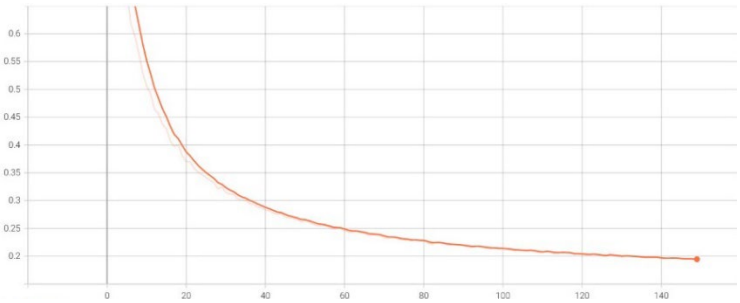


Figura 44. Épocas vs. Pérdida modelo 3.



De la Figura 43 y Figura 44 se evidencia que a medida que disminuye la pérdida del modelo, aumenta su exactitud, pudiendo suponer que al llegar a 60 épocas lograría ser suficiente para el mejoramiento del modelo en total, pero se evidencia que es necesario una serie adicional de pérdidas, pues aunque se estabiliza la curva de exactitud, las pérdidas continúan disminuyendo, por lo que tener la comparativa entre ambos comportamientos genera un mayor espectro en la decisión de entrenamiento.

Realizando una comparativa entre los tres modelos propuestos, teniendo como base sus matrices de confusión, distribución de exactitud y de pérdidas, se entendió que, a pesar de recibir un porcentaje inferior de aciertos en la matriz de confusión, el contar con una mayor exactitud se decidió que el modelo 3 será el encargado de ejecutar la clasificación de locutor. Para comprobar se realizó una etapa de evaluación al modelo obteniendo la matriz de confusión relacionada a audios ajenos a los que fueron utilizados en su entrenamiento y la clasificación realizada por el modelo, lo que se evidencia en la Figura 45 y la Tabla 10.

Figura 45. Matriz de confusión validación

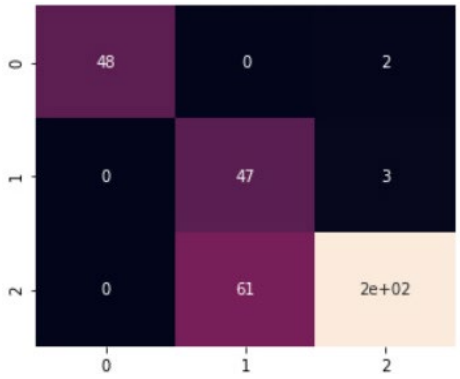


Tabla 10. Matriz de confusión validación

		Categoría			% Aciertos	% Fallos
		0	**1**	**2**		
Categoría	**0**	48	0	2	96	4
	1	0	47	3	94	6
	2	0	61	200	76.62	23.38

Recordando las condiciones en las que fueron grabados los audios de entrenamiento y validación, un resultado como el representado es esperado, teniendo en cuenta que estos últimos fueron capturados por dispositivos ajenos al micrófono con el que se obtuvieron los audios de los locutores a reconocer, siendo así un resultado coherente y acertado, pues un porcentaje de acierto superior a 94% para cada uno de los locutores previamente registrados representa un alto grado de acierto, lo que permite tener como oportunidad de mejora la categorización de los locutores desconocidos o no registrados. De esta manera permitiendo resaltar y reafirmar directamente los resultantes de validaciones en dónde se alcanzó un 96% y 94% en ambientes controlados y no controlados, respectivamente.

CONCLUSIONES

Los datos son la base de los modelos de inteligencia artificial, por lo que contar con un ecosistema controlado al momento de obtenerlos es la opción recomendada para tener conocimiento de la mayor cantidad de parámetros por permitir y prevenir en su tratamiento. Para el caso de audios en un sistema de reconocimiento, es útil realizar su grabación con el uso de un micrófono en específico y un ambiente determinado. No contar con lo propuesto no representa un obstáculo que imposibilita alcanzar el objetivo, pero lograrlo genera mejores resultados relacionados al porcentaje de aciertos y disminución del consumo computacional de los modelos diseñados.

Clasificar un audio decidiendo quién es su locutor resulta aún más alcanzable al entender los procesos de tratamiento de audio que mejor encajan para lograrlo. Por lo que un análisis frecuencial mediante lo que se conoce como MFCC o Coeficientes Cepstrales de Mel es la opción por excelencia, pues permite obtener la información más relevante de un audio de voz humana al ser estos la representación del habla basados en la percepción auditiva humana.

El conjunto de MFCC acoge los conceptos de separación por ventana, Transformada de Fourier discreta, bancos de filtros de Mel y Transformada de Coseno discreta, siendo estos importantes conocimientos en el Procesamiento Digital de Señales (DSP).

Los modelos de inteligencia artificial optimizados para la clasificación de imágenes se componen por capas de Convolución 2D, que permite extraer las características principales de cada imagen mediante filtros. El uso de mayor o menor cantidad de capas responde a la complejidad del modelo, que se resume en costo computacional tanto en su entrenamiento, como en su ejecución. Existen situaciones que indican la necesidad de sacrificar complejidad por mejoría en sus resultados, esto se debe a la naturalidad de los datos con los que es entrenado el modelo, decisión que toma para el beneficio de los resultados del proyecto, pues la falta de uniformidad de los datos exige mayor complejidad en la arquitectura del modelo diseñado.

Otro parámetro para tener en cuenta en el diseño de los modelos de inteligencia artificial son las funciones de activación con las que se relacionarán cada una de las capas, en este caso, de Convolución 2D y totalmente conectadas. Tras implementar modelos utilizando las funciones de activación ‘relu’ y ‘tanh’, fue esta última la que encajaba con las características de los espectrogramas generadas por audio, pues su cardinalidad comprende los números enteros, desde negativos hasta positivos, dónde la tangente hiperbólica se define como una función que se acerca infinitamente a -1 y a 1, pasando por el 0 en su origen.

Un modelo más complejo sin la cantidad y calidad necesaria de datos no necesariamente genera mejores resultados, por lo que se converge en que aumentar significativamente cantidad de audios de entrenamiento permitirá una considerable disminución de complejidad del modelo, a su vez una mejora en su capacidad de acertar.

Es necesario también, definir un ecosistema de obtención de datos uniforme, en el que se considere al menos uno de los siguientes aspectos: dispositivo de grabación y/o el entorno. Un espacio silencioso, sin ruidos considerables, un micrófono en específico serían las características que permitan un aumento en la capacidad de acertar por parte del modelo. Aunque sin contar con un ambiente completamente controlado se obtuvo un resultado apreciable, que este exista aporta también a disminuir la complejidad del modelo.

Por todo lo anteriormente desarrollado, analizado y comprendido, se puede determinar que es posible realizar un sistema de reconocimiento de voz basado en inteligencia artificial (AI) y procesamiento digital de señales (DSP), siento estos dos conceptos la base para el tratamiento y clasificación de los datos, debido a que, con DSP, se logra convertir de un formato audio a una imagen que será posteriormente entendida e interpretada por un modelo de inteligencia artificial, logrando así cumplir con el objetivo principal del proyecto.

RECOMENDACIONES

Siguiendo el flujo de análisis de los datos y diseño de los modelos de inteligencia artificial propuesto, un ecosistema completamente controlado para la obtención de audios, tanto de entrenamiento como de validación resulta necesario, además de aumentar considerablemente la cantidad de locutores y audios per cápita.

Una vez logrado un mejor resultado en el sistema de clasificación, se permite la creación de un sistema visual en el que se pueda embeber PIA, así mejorando la experiencia del usuario mediante una interfaz sencilla y a su vez dinámica.

Esto permitiría ampliar el abanico de oportunidades, generando la posibilidad de incluir al sistema la característica de generar audio a través de texto, y así PIA pueda tener una conversación corta, pero concreta en el sistema de seguridad en que se encuentre aplicado.

BIBLIOGRAFÍA

BOTELHO, Bridget. Virtual assistant (AI assistant). Techtarget [sitio web]. Boston; [Consultado: 5 de enero de 2021]. Disponible en: https://searchcustomerexperience.techtarget.com/definition/virtual-assistant-AI-assistant

HEATH, Nick. What is Artificial Intelligence (AI)?. zdnet [sitio web]. Londres; [Consultado: 5 de enero de 2021]. Disponible en: https://www.zdnet.com/article/what-is-ai-everything-you-need-to-know-about-artificial-intelligence/

Tecnologías de Inteligencia Artificial y sus categorías [en línea]. Auraquantic. [Consultado: 5 de enero de 2021]. Disponible en: https://www.auraquantic.com/es/tecnologias-de-inteligencia-artificial-y-sus-categorias/

ZAMBRANO, Juan. ¿Aprendizaje supervisado o no supervisado? Conoce sus diferencias dentro del Machine Learning y la Automatización Inteligente. Medium. [Consultado: 5 de enero de 2021]. Disponible en: https://medium.com/@juanzambrano/aprendizaje-supervisado-o-no-supervisado-39ccf1fd6e7b

THE INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY 2017 [en línea]. En (3: 21-29 agosto, 2017: Antalya, Turquia). Memorias de International Conference on Engineering and Technology. Antalya. IARES.net. [Consultado: enero 6 de 2021]. Disponible en: https://www.researchgate.net/profile/Saad-Albawi/publication/319253577_Understanding_of_a_Convolutional_Neural_Network/links/5ad26025458515c60f51dbf9/Understanding-of-a-Convolutional-Neural-Network.pdf

Pérdida de registro. DataScience [sitio web]; [Consultado: 16 de febrero de 2022]. Disponible en: https://datascience.eu/es/programacion/perdida-de-registro/

KYRYKOVYCH, Anastasia. What is a Deep Neural Network. KDNudggets [sitio web]. Emiratos Árabes Unidos; [Consultado: 6 de enero de 2021]. Disponible en: https://www.kdnuggets.com/2020/02/deep-neural-networks.html

Understanding Audio: What is DSP?. Yamaha [en línea]. Japón; [Consultado: 7 de enero de 2021]. Disponible en: https://uc.yamaha.com/insights/blog/2018/november/understanding-audio-what-is-dsp/

Understanding Spectrograms. Izotope [en línea]; [Consultado: 7 de enero de 2021]. Disponible en: https://www.izotope.com/en/learn/understanding-spectrograms.html

What is a WAV file?. Fileformat [en línea]; [Consultado: 7 de enero de 2021]. Disponible en: https://docs.fileformat.com/audio/wav/

Nyquist and Shannon's Sampling Theorems. National Instruments [en línea]. [Consultado: 7 de enero de 2021]. Disponible en: https://zone.ni.com/reference/en-XX/help/370524V-01/siggenhelp/fund_nyquist_and_shannon_theorems/

KAISER, Gerald. Windowed Fourier Transforms. Springer [sitio web]. Boston; [Consultado: 6 de enero de 2021]. Disponible en: https://link.springer.com/chapter/10.1007/978-0-8176-8111-1_2

Discrete Cosine Transform [en línea]. Paperswithcode. [Consultado: 7 de enero de 2021]. Disponible en: https://paperswithcode.com/method/discrete-cosine-transform

What is a Spectogram? [en línea]. PNSN. [Consultado: 7 de enero de 2021]. Disponible en: https://pnsn.org/spectrograms/what-is-a-spectrogram

FAYEK, Haytham. Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between [sitio web]. Seattle; [Consultado: 8 de enero de 2021]. Disponible en: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

¿Qué es el software libre? [en línea]. GNU. [Consultado: 8 de enero de 2021]. Disponible en: https://www.gnu.org/philosophy/free-sw.es.html

PHAM, Hubert. PyAudio Documentación [sitio web]. California; [Consultado: 8 de enero de 2021]. Disponible en: https://people.csail.mit.edu/hubert/pyaudio/docs/

Librosa. Librosa [en línea]; [Consultado: 8 de enero de 2021]. Disponible en: https://librosa.org/

Por qué TensorFlow [en línea]. Tensorflow; [Consultado: 9 de enero de 2021]. Disponible en: https://www.tensorflow.org/

Intuitively Understanding Convolutions for Deep Learning. TowardsDataScience. [Consultado: 8 de enero de 2021]. Disponible en: https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1