Uso de la Transformada de Fourier para el Reconocimiento de voz. Caso de Estudio: Redes Convolucionales

Freddy L. Abad L., Cristian X. Collaguazo M., Esteban D. Vizhñay E.
Facultad de Ingeniería, Universidad de Cuenca
Cuenca, Ecuador
{freddy.abadl, cristian.collaguazo, esteban.vizhñay}@ucuenca.edu.ec

***Resumen*: La voz humana es una cualidad de cada persona que puede ser interpretada de varias maneras, una de ellas es el análisis gráfico mediante ondas; cada voz genera una única onda de acuerdo a las diferentes características que esta posee. El objetivo principal de este documento es describir la metodología para construir un clasificador para reconocimiento de hablante implementando redes neuronales convolucionales (CNN), que son una clase de redes neuronales profundas, mediante la transformación de un audio a una imagen de onda y posteriormente realizar una clasificación según las ondas que se encuentren. El proceso de entrenamiento se realiza con diferentes audios que contienen la lectura de un mismo documento por parte de los autores, pudiendo así generar una onda característica para la voz de cada participante.**

Palabras clave: fourier, transformada, convolución, deep learning, inteligencia artificial, procesamiento, reconocimiento, voz.

I. INTRODUCCIÓN

El reconocimiento de voz se ha convertido en una rama muy importante en el campo de la autenticación biométrica. Dadas sus ventajas únicas de operación remota, la autenticación de identidad basada en la impresión y reconocimiento de voz se ha aceptado y aprobado de forma gradual en los campos como la banca implementada en su sección telefonica, usos aplicados a IoT tal como la domótica, la seguridad de manera de autenticación, entre otros [1]. Para obtener mejores resultados de un reconocimiento de voz, es necesario tener en cuenta que existen muchas dificultades al momento de extraer las características de la voz. Estos problemas vienen dados de tal forma que, una palabra difícilmente será pronunciada de igual manera dos veces, entre los factores de esta problemática tenemos: estado de ánimo, salud, fuerza de pronunciación, tiempo, entonación, entre otros [2].

Las tecnologías tradicionales de reconocimiento de voz se basan en estadísticas y en métodos de procesamiento de señales, en este documento tratadas como "ondas", para extraer las características presentes en el sonido brindado por el hablante, y así tratar de verificar su identidad. Sin embargo, estas tecnologías se ven truncadas por la larga duración de habla que tiene que brindar el hablante, por ejemplo, 30 segundos; en muchos sistemas de interacción por voz es necesario verificar rápidamente la identidad del usuario [1].

Sin importar las dificultades presentadas al momento de realizar un reconocimiento de voz, se han desarrollado varios algoritmos que tratan de eliminar el ruido de un sonido o, procuran eliminar los factores causantes de los problemas, permitiendo así, determinar un mayor nivel de coincidencia entre las pronunciaciones para poder lograr un reconocimiento eficaz [2].

Los diferentes sistemas actuales de reconocimiento de voz hacen uso de algoritmos basados en modelado acústico y lingüístico. El modelado acústico significa presentar una conexión entre las unidades lingüísticas del habla y las señales de audio, y, el modelado del lenguaje se basa en hacer coincidir los sonidos con secuencias de palabras para distinguir entre palabras que suenan similares.

La voz al igual que las huellas dactilares, llevan impregnadas firmas biométricas únicas, es específica del individuo y por esta razón sirve como un método de identificación. Por lo tanto, todo el mundo tiene una huella de voz única que se forma a lo largo del proceso de desarrollo de nuestros órganos vocales. No importa cuán notablemente similar pueda ser la voz imitada a la voz original, sus huellas de voz seguirán siendo diferentes [3].

En el campo de las redes neuronales han aparecido técnicas para reconocimiento de voz, las cuales dos de estas son: Alineamiento dinámico del tiempo (DWT por sus siglas en inglés) y Modelos Ocultos de Markov (HMM por sus siglas en inglés).

En los últimos años, las redes neuronales, especialmente las redes profundas, se utilizan cada vez más en el reconocimiento de voz. A diferencia de los HMM, las redes neuronales no hacen suposiciones sobre las propiedades estadísticas del sonido y tienen varias cualidades que las hacen modelos de reconocimiento atractivos para el reconocimiento de voz.

Un enfoque alternativo o tal vez complementario para solventar los problemas del reconocimiento de voz es usar redes neuronales convolucionales (CNN) [3]. Se sabe que las redes neuronales convolucionales CNN se desempeñan bien en el reconocimiento de imágenes [3].

Por lo tanto, en este trabajo se propone el uso de redes neuronales convolucionales (CNN) para el reconocimiento de voz o de hablante, transformando los audios en imágenes denominadas espectrogramas y tratandolas como conjunto de entrenamiento y de testing. Los aspectos claves de estas propuestas son: i) Transformacion de audios a imagenes (espectrogramas) .ii) Tratar el problema como reconocimiento de imágenes.

El resto del artículo tiene la siguiente estructura. La sección II describe los objetivos del proyecto. En la sección III se presenta un pequeño marco teórico de los temas que se está usando para el desarrollo de los objetivos. La sección IV muestra los métodos que se usó para la construcción del sistema. La sección V se describe los problemas que se encontraron durante la investigación y el desarrollo del proyecto. En la sección VI muestra la discusión y los resultados obtenidos. Y finalmente la sección VII muestra la conclusión.

II. OBJETIVOS

- Reconocer al hablante mediante la voz de los participantes usando técnicas de Deep Learning, específicamente redes neuronales convolucionales.
- Generar un conjunto de entrenamiento con datos propios de los integrantes del grupo y posteriormente generar varios experimentos para encontrar la configuración de red óptima.
- Extraer las características que una señal de audio las cuales sean adecuadas para la identificación del contenido relevante que ayude para realizar el

reconocimiento de voz. Todo esto mediante el análisis de los coeficientes MFCC (Coeficientes Cepstrales en las Frecuencias de Mel).

- Analizar mediante diferentes pruebas que tan buena es la técnica aplicada para el reconocimiento de voz.

III. MARCO TEÓRICO

En esta sección se presenta a breves rasgos los conceptos sobre las técnicas utilizadas para el reconocimiento de voz.

A. Transformada rápida de Fourier

La Transformación rápida de Fourier (FFT) es un algoritmo matemático que calcula la Transformación discreta de Fourier (DFT) de una secuencia dada. La única diferencia entre FT (Transformada de Fourier) y FFT es que FT considera una señal continua mientras que FFT toma una señal discreta como entrada. DFT convierte una secuencia (señal discreta) en sus componentes de frecuencia al igual que FT para una señal continua.

B. Espectrograma (MFCC)

El espectrograma consiste en la representación gráfica del espectro de frecuencias de la emisión sonora. El espectrograma puede revelar rasgos, como altas frecuencias o modulaciones de amplitud, que no pueden apreciarse incluso aunque estén dentro de los límites de frecuencia del oído humano. Normalmente, un espectrograma representa el tiempo sobre el eje horizontal, la frecuencia sobre el eje vertical y la amplitud de las señales mediante una escala de grises o de colores [7].

Los coeficientes cepstrales simplemente es la información de la tasa de cambio en las bandas espectrales. De forma convencional en análisis de señales en el tiempo de cualquier componente periódico como una grabación que se repite o un eco, aparece como picos agudos en el espectro de frecuencia correspondiente a realizar el cálculo de la Transformada de Fourier Rápida (FFT) a la señal, esto se puede ver en la siguiente figura.[11]

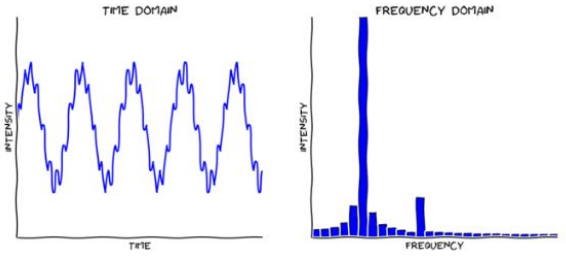

Figura 1. Representación en el dominio de la frecuencia de una señal de audio que representa un eco. **Fuente**: [11]

Al tomar el registro de la magnitud de este espectro de Fourier y luego tomar el espectro de este registro mediante la transformada inversa de fourier (IFT) de la logaritmo de la estimación de la señal espectro. Observamos un pico siempre que haya un elemento periódico en la señal original en el tiempo. Cómo aplicamos una transformación en el espectro de frecuencia en sí, el espectro resultante no está ni en el dominio de la frecuencia ni en el dominio del tiempo y, por lo tanto ha esto se le denomina dominio de quefrency. Y este espectro del registro del espectro de la señal de tiempo se denominó cepstrum.

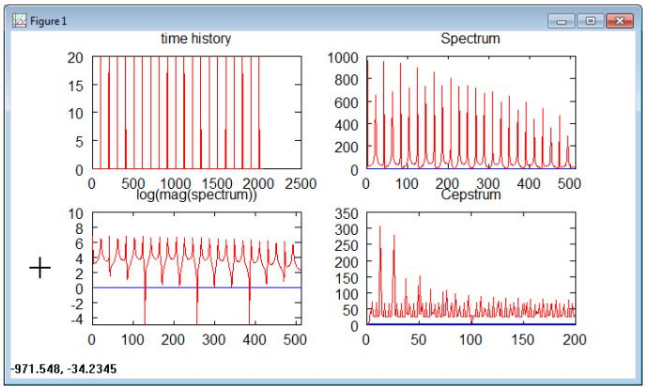

Figura 2. Pasos para formar cepstrum a partir de la historia del tiempo. **Fuente**: [11]

El tono de la voz es una de las características de una señal de voz y se mide como la frecuencia de la señal. La escala de Mel es una escala la cual relaciona la frecuencia percibida de un tono con la frecuencia medida real. Esta escala se ha derivado de un conjunto de experimentos con humanos. [11]

La escala de Mel busca percibir la diferencia que hay entre sonidos de diferente frecuencia. Una frecuencia medida en Hertz (f) se puede convertir a la escala de mel utilizando la siguiente fórmula.

$$Mel(f) = 2595\, log(1 + \frac{f}{700})$$

Cualquier sonido generado por humanos se determina mediante una serie de características únicas. Si esta forma se puede determinar correctamente, cualquier sonido producida se puede representar con precisión. La envolvente del espectro de potencia temporal de la señal de voz es representativa del tracto vocal y el MFCC (que no es más que los coeficientes que componen el cepstrum de frecuencia Mel) representa con precisión esta envolvente. [11]

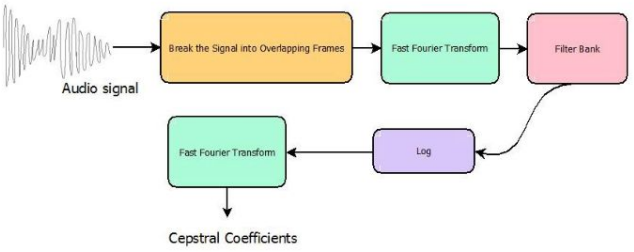

Figura 3. Diagrama de bloques que representa el resumen paso a paso de cómo llegamos a los MFCC. **Fuente**: [11]

C. Deep Learning

El Deep Learning (aprendizaje profundo), ha tenido varios logros en relación al reconocimiento de patrones y Machine Learning. No obstante, Deep Learning es un subcampo de Machine Learning, y su principio fundamental es aprender abstracciones de alto nivel de datos mediante el uso de arquitecturas jerárquicas [3].

Los métodos que se aplican en Deep Learning, son métodos de de aprendizaje con múltiples niveles de representación, obtenidos mediante la composición de módulos simples pero no lineales que transforman la representación en un bajo nivel (comenzando por la entrada en su forma más particular) en una representación a un nivel más alto [4]. Los métodos utilizados en el Deep Learning, han mejorado drásticamente el estado de la técnica en reconocimiento de voz, reconocimiento de objetos visuales, detección de objetos y entre otros dominios relacionados [4].

La implementación del Deep Learning está generando grandes e importantes avances en la solución de problemas que se

presentaban para la comunidad de la inteligencia artificial durante muchos años [4].

D. Métodos utilizados de Deep Learning para reconocimiento de voz

En los últimos años, el principal enfoque para el desarrollo del Deep Learning ha sido el campo de la visión por computador y, debido a esto, se ha generando una amplia gama de enfoques relacionados. De acuerdo al método básico del Deep Learning se puede dividir en cuatro categorías: Redes Neuronales Convolucionales (CNN), Máquinas Boltzmann Restringidas (RBM), Autoencoder y Sparse Coding [3]. Con respecto al reconocimiento de voz, las técnicas utilizadas implican CNN, Unidades de Memoria a largo plazo (LSTM por sus siglas en inglés) que permiten clasificar, procesar y hacer clasificaciones basadas en series de tiempo y los Coeficientes Cepstrales en las Frecuencias de Mel (MFCC por sus siglas en inglés).

Con el fin de centrar el estudio en redes neuronales capaces de identificar voces, se ha puesto en consideración las siguientes categorías usadas de Deep Learning:

E. Redes Neuronales Convolucionales.

Las Redes Neuronales Convolucionales (CNN) o redes de convolución, es uno de los enfoques del Deep Learning más notable, pues aquí se entrenan las capas múltiples de manera robusta. Su aplicación ha demostrado que es altamente efectivo y es utilizado en el procesamiento de imágenes.

En general, una CNN consta de varias capas, entre estas podemos encontrar tres capas neuronales principales; una capa de entrada que se encarga de recibir los píxeles de una imágen. Las capas ocultas que es donde se encuentra la información importante, dentro de estas capas existen dos operaciones importantes:

- *Pooling o agrupación*: operación usada para reducir el tamaño de la imágen, de modo que cuando el proceso avance a las siguientes capas, se procese una imágen de menor tamaño. Este proceso se muestra en la figura 4

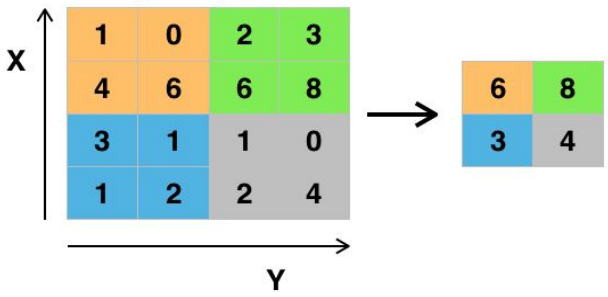

Figura 4. Reducción de una matriz aplicando Max-Pooling. En la reducción se toma en cuenta los valores más significativos. **Fuente**: [4]

- *Convoluciones*: esta operación consiste en incluir filtros en las imágenes para detectar patrones relevantes [3] [4]. Este proceso se puede observar en la figura 5.

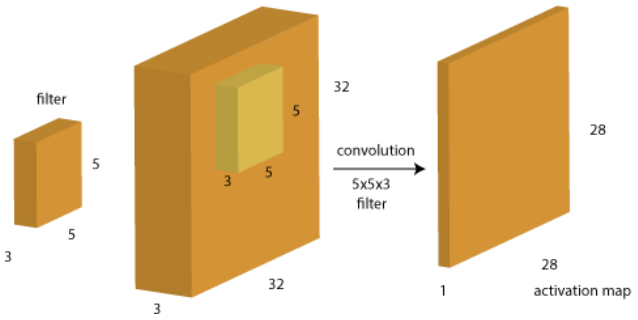

Figura 5. Filtrado de imágenes mediante convoluciones. **Fuente**: [3]

En el trabajo realizado por Y. Guo et al. [3] se presenta un estudio de las diferentes capas que conforman una CNN. Por motivos que se expanden del tema. Se deja a voluntad del lector el adquirir conocimiento sobre estas capas y su diferente forma de clasificación.

IV. METODOLOGÍA

A. Trabajo relacionado.

Este trabajo se baso en en un aporte de Yicong Liu, el cual el principal objetivo es la construcción de un clasificador para reconocer un número hablado entre el 0 y el 9 [7]. Dicho sistema permite reconocer el número hablado y mostrarlo como salida en formato de texto. Este paper se basa en la problemática de reconocer números escritos (Figura 3), en el cual, a pesar de diferir su forma de representar, (manera de escribir cada número difiere en cada persona). al tener un gran dataset se puede entrenar un modelo suficientemente bueno para el reconocimiento de números.

Figura 6. Representación de dataset usado para reconocimiento de números mediante CNN **Fuente**: [8]

Mediante este concepto se hizo uso de las ideas principales de este trabajo, en la cual se hizo uso de MFCC (espectrogramas) como características para representar imágenes, posteriormente estas imágenes se han usado para el entrenamiento de la red neuronal convolucional.

De esta manera, el problema de reconocimiento de voz se transfiere a un problema de reconocimiento de imagen. Basándose en el trabajo anterior se ha construido un clasificador para el conjunto de datos con capas de convolución y de max-pooling con el objetivo de identificar al hablante, sin importar el lo que el hablador diga en ese momento.

B. Método utilizado para obtener conjunto de datos.

De un total de cuatro audios que corresponden a los hablantes de una duración aproximada de 20 minutos cada uno en un formato de grabación .m4a, se dividieron los audios en clips muy cortos (1, 2, 3 y 5 segundos), a partir del cual la pronunciación característica de cada hablante se identifica mediante la comparación de cientos de diferentes modelos de frecuencia de voz.

Desarrollando el proyecto mediante el uso de librerías de Linux para el pre tratamiento de los datos, se consigue recortar los audios en múltiples cuadros de voz consecutivos para encontrar características propias de la voz de cada participante, tal como se muestra en la figura 7. Toda la información que se encuentra que los cuadros de voz se procesa para tratar de eliminar sonidos que no aporten información del habla. Como resultado, se reduce la longitud total del audio a una imagen que contiene las características principales de la voz de un hablante. El resultado del espectro de las características finales de la voz se puede apreciar en la figura 8.

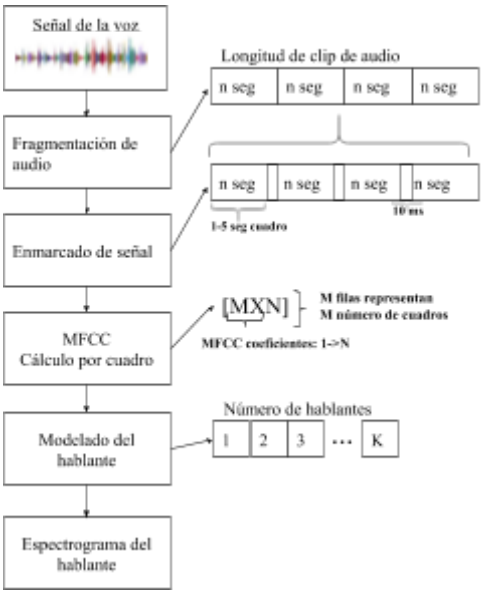

Figura 7. Transformación de una señal de audio a un espectrograma. Fuente: [9]

En la Figura 8 se puede apreciar el resultado obtenido al aplicar el proceso de transformación de audio a un espectrograma MFCC, la imagen contiene las características más importantes de la voz de cada participante. Posteriormente cada espectrograma será utilizado para realizar el entrenamiento de la red neuronal.

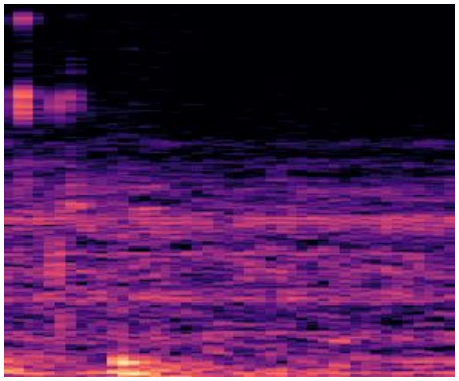

Figura 8. Espectrograma de Mel que contiene las características de la voz.

C. Red Neuronal Convolucional para la clasificación de espectrogramas que representan sonidos

El objetivo es generar un modelo de clasificación que permita determinar la identidad del hablante del audio. Las redes neuronales convolucionales tienen su aplicación más fuerte en el reconocimiento de imágenes. Su estructura cuenta con una capa de entrada, las capas ocultas intermedias que a su vez cuentan con dos operaciones importantes: Pooling (agrupación), operación usada para reducir el tamaño de las imágenes y la otra operación son las convoluciones (filtros), para detectar patrones relevantes en las imágenes.

El sistema de reconocimiento desarrollado permite la clasificación de imágenes (espectrogramas). Para el entrenamiento, el conjunto de datos utilizado consta con espectrogramas que representan los audios de los hablantes. Para el entrenamiento, es necesario tener los archivos ordenados en un directorios como se muestra en la figura 9. La estructura contiene dos directorios, uno de entrenamiento y uno de validación; los dos directorios contienen los espectrogramas de los audios de cada uno de los hablantes.

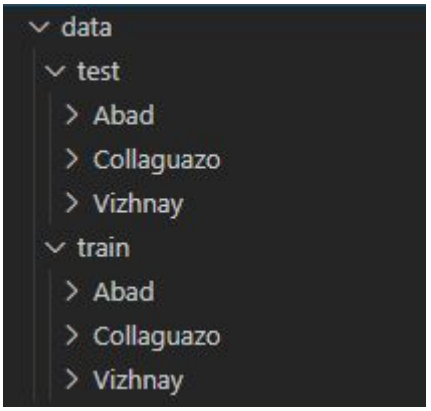

Figura 9. Estructura de directorios empleada para el entrenamiento y la validación de la red neuronal.

Se ha desarrollado la aplicación con el uso de Tensorflow[1] usado como Backend, además de la API de Keras[2], para la red neuronal convolucional se ha hecho uso de parámetros como:

- *Épocas*: número de veces que se itera en el conjunto de datos durante el entrenamiento.
- *Altura y longitud*: tamaño al cual se van a procesar las imágenes (espectrogramas).
- *Batch Size*: es el número de imágenes a procesar en cada iteración.
- *Iteraciones*: es el número de veces que se va a procesar la información en cada una de las épocas.
- *Iteraciones de validación*: variable que indica que al final de cada época se ejecutarán iteraciones con el conjunto de validación.
- *Filtros de convoluciones*: permite cambiar la profundidad de la imágen, para los experimentos se aplicarán 2 filtros en las convoluciones.
- *Tamaño de filtros*: indica altura y longitud para los filtros en las convoluciones.
- *Tamaño de pool*: tamaño del filtro que se usará durante la técnica de MaxPooling.
- *Número de clases*: indica la cantidad de clases para la clasificación. Para los experimentos, el número de hablantes.
- *Learning Grade*: que tan grandes serán los ajustes realizados por la red neuronal para acercarse a una solución óptima (normalmente es un número pequeño).

1. Procesamiento de las imágenes:

[1] Biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, permite construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones.

[2] Keras es una API de redes neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow. Fue desarrollado con un enfoque en permitir la experimentación rápida.

Se hace uso de un generador para las imágenes tanto para el conjunto de datos como para el conjunto de validación. Dicho generador realiza operaciones como:

- *Reescalar los píxeles de la imágen*: cada uno de los píxeles de la imagen perteneces a un rango entre 0 y 255, por las grabilla RGB y, al hacer este reescalado, los valores de los píxeles se encuentren en un rango de 0 a 1. Se aplica esta operación con el objetivo de hacer más eficiente el entrenamiento.
- *Shear Range (inclinar imágenes)*: permite inclinar las imágenes (espectrogramas) para que el algoritmo "aprenda" de mejor manera.
- *Zoom Range*: permite hacer zoom en los espectrogramas de manera aleatoria con el objetivo de identificar partes incompletas en los espectrogramas.
- *Direccionalidad de imágenes*: permite invertir una imagen (espectrograma); con el objetivo de distinguir direccionalidad y mejore el aprendizaje.

2. Creación de la red convolucional:

La red neuronal convolucional, que se va a generar es secuencial; es decir, varias capas apiladas entre ellas. A continuación se muestra el proceso de creación:

- Fijar la primera capa de la red como una convolución con parámetros como el número de filtros, tamaño de los filtros y una función de activación.
- Posteriormente se agrega una capa de Max Pooling con un tamaño de pool definido.
- Ahora se fija una siguiente capa convolucional.
- Se añade otra capa de Max Pooling con un tamaño de pool definido.

Es importante mencionar que se entrenó distintos modelos que permiten revisar cuantas iteraciones y épocas se debe emplear para un modelo único de reconocimiento de voz, lo cual se hizo dos intentos. Tomando en cuenta, los recursos de hardware que se emplearon para la ejecución de esta, se entrenó modelos con audios 1 seg de duración de los 3 individuos. Se hizo modelos con 500 iteraciones por cada una de las 10 épocas. Y un segundo modelo con audios de 1000 iteraciones por cada una de las 20 epocas. Los resultados se muestran en la Tabla 1. El resultado final del proceso será una red neuronal que contiene una capa de convolución, seguida por una capa de Max Pooling, luego otra capa de convolución seguida por otra capa de Max Pooling. Una vez que una imágen haya pasado por todas las capas mencionadas anteriormente tendremos como resultado una imágen con un tamaño bastante pequeño y con gran profundidad que es necesario que sea plana (es decir, una dimensión con toda la información de la red neuronal). Cada uno de las diferentes capas utilizadas para el reconocimiento de voz aplicados a este trabajo se presenta en la Figura 10. Posteriormente se realiza un proceso denominado "apagado de neuronal", esta operación es aplicada con el objetivo de evitar el sobreajuste ya que si todo el tiempo las neuronas permanecen activadas puede que la red neuronal aprende algo en específico. Entonces si de manera aleatoria en cada paso solo activa ciertas neuronas, la red aprenderá de todas las maneras posibles con un modelo que se adaptará de mejor manera a la información nueva. Como último paso se crea una última capa que contendrá las salidas dependiendo del número de clases a clasificar (para los experimentos el número de hablantes) junto con una función de activación que indicará el valor de porcentaje más alto que pertenece a la clase correcta.

Finalmente, se almacena la estructura del modelo y los pesos en cada una de las capas entrenadas con la configuración establecida en archivos de extensión .h5, esto con el objetivo de no entrenar nuevamente el modelo cada vez que se requiera hacer una clasificación. Todo este esquema arquitectónico se muestra en la figura 10 que representa la arquitectura del sistema utilizado para el reconocimiento de voz de un hablante.

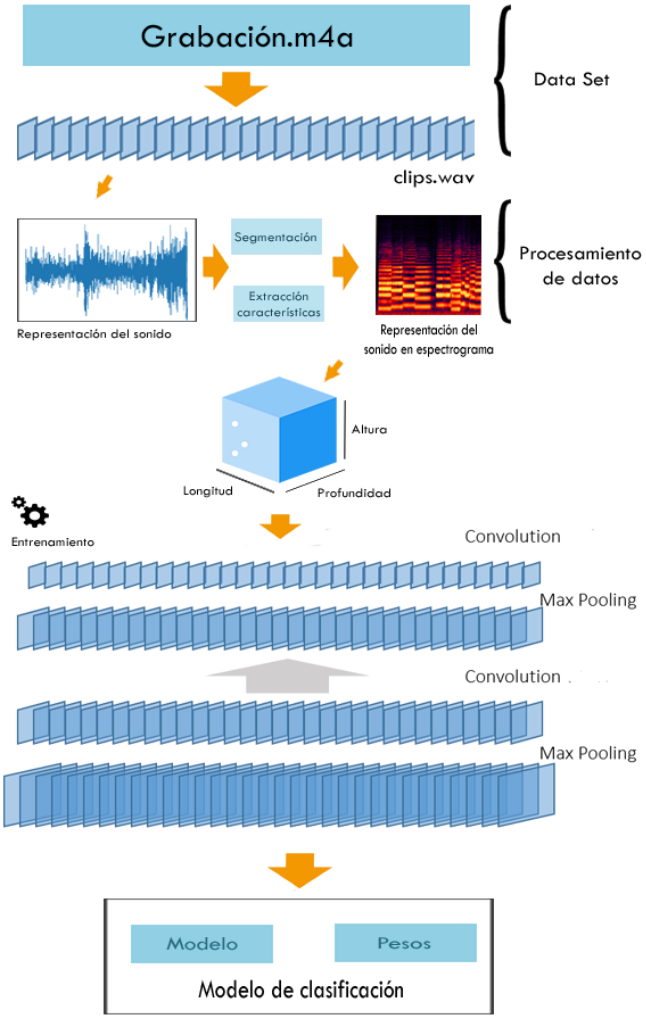

Figura 10. Arquitectura del sistema utilizado para el reconocimiento de voz usando redes neuronales convolucionales.

3. Reconocimiento de voz:

Para hacer el reconocimiento de la voz, se ha hecho uso de grabaciones de prueba (de cada uno de los hablantes y evitando un ambiente ruidoso, debido a que afecta a la generación de los espectrogramas). Además, son necesarios los archivos de modelo y pesos generados durante la creación y entrenamiento de la red neuronal. Los audios de prueba son transformados a espectrogramas para analizar y clasificar la voz del hablante. Para el proceso de clasificación, la imágen (espectrograma) es procesada como un arreglo de valores que la represente; dicho arreglo de valores es enviado al modelo para realizar la clasificación y el modelo devuelve un arreglo con valores de probabilidades de cada una de las clases.

Finalmente, se obtiene el valor de probabilidad más alto que es el que corresponderá a una clase definida y por ende, a la voz del hablante. Y se indicará a quién pertenece dicha voz. Todo este proceso, se puede apreciar en la figura 11.

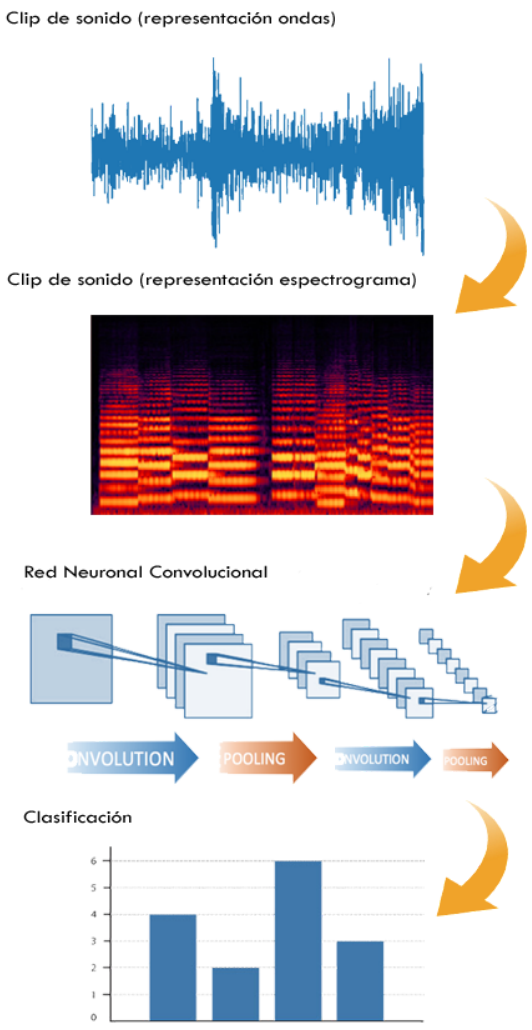

Figura 11. Proceso de extracción del espectrograma el audio, aplicación de la red neuronal y resultado de clasificación mediante el algoritmo entrenado.

V. PROBLEMAS ENCONTRADOS

Las problemáticas en el desarrollo de este proyecto fueron distintas en distintos campos:

1. Problemáticas con el uso de las herramientas proveídas por Python para Deep Learning (Tensorflow, Keras). Ya que originalmente el proyecto se basó en el uso de Theano como Backend de Keras, lo cual trajo consigo múltiples problemas al momento de entrenar.
2. Tratamiento del audio, debido a que un audio por lo general se graba con sonidos externos, que afectan los datos importantes. Es por eso que se proceso el audio para eliminar sonidos que afecten el entrenamiento del modelo.
3. Los resultados obtenidos por los modelos inicialmente utilizados se pudieron mejorar utilizando modelos donde se hagan mayores iteraciones y mayores épocas. Evidentemente se mejoró los resultados, pero los tiempos de entrenamiento se extendieron.
4. Para obtener mejores clasificaciones, el tratamiento de los audios de ingreso al modelo, es decir al momento de predecir, se generó problemas por situaciones externas (ruido). Dado estas problemáticas, se trato el audio con Scipy [10], para la reducción de los ruidos externos. Sin embargo, la clasificación no es confiable al 100%.

VI. DISCUSIÓN Y RESULTADOS

Se realizaron varias pruebas al sistema de manera que en cada prueba se modifica la cantidad de capas ocultas, a manera de tratar de encontrar un mejor resultado. A continuación se presenta los resultados usando cuatro modelos. Para la configuración de la red neuronal se ha hecho uso de los parámetros mencionados en la sección de Metodología, estos parámetros se pueden observar en la Tabla I. El resultado obtenido al aplicar el modelo de la red neuronal a los audios de entrenamiento se presentan en la Tabla II.

TABLA I

PARÁMETROS UTILIZADOS PARA LA CREACIÓN DE LA RED NEURONAL CONVOLUCIONAL

Parámetro	Valor
Épocas	20
Longitud, Altura	150, 150
Batch Size	32
Pasos	1000
Pasos de validación	300
Filtros de convolución 1	32
Filtros de convolución 2	64
Tamaño filtro 1	(3,3)
Tamaño filtro 2	(2,2)
Número de clases	4

La TABLA I representa los valores de los parámetros parámetros explicados en la sección IV de Metodología en el apartado C, empleados para la creación de la red neuronal convolucional.

TABLA II
RESUMEN DEL MODELO APLICADO A CNN

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 150, 150, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 75, 75, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 75, 75, 64)        8256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 37, 37, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 87616)             0
_________________________________________________________________
dense (Dense)                (None, 256)               22429952
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 1028
=================================================================
Total params: 22,440,132
Trainable params: 22,440,132
Non-trainable params: 0
_________________________________________________________________
```

La TABLA II representa el resumen del modelo de red neuronal aplicando los parámetros de la TABLA I.

TABLA III
RESULTADOS DE LA CLASIFICACIÓN DE LA RED NEURONAL CONVOLUCIONAL

Modelo	Input	Output	Clasificado Correcto
Modelo de entrenamiento audio de 1 segundos	2	2	SI
	2	2	SI
	1	0	NO
	0	1	NO
Modelo de entrenamiento audio de 3 segundos	2	2	SI
	2	2	SI
	1	1	SI
	0	1	NO
Modelo de entrenamiento audio de 5 segundos	2	2	SI
	0	0	SI
	1	1	SI
	0	0	SI

Los parámetros de la TABLA III son: *Input* significa el audio de entrada o de prueba que se envía a la red neuronal, cada participante tiene asignado un número que es único y lo identifica de los demás. *Output* significa la salida de la red neuronal, *Clasificado Correcto* esta columna significa si la predicción es correcta o nó en base a Input y Output, si estos coinciden, la predicción es correcta; en caso contrario, la predicción será incorrecta.

Como recomendación se puede usar en el entrenamiento de datos, todas los espectrogramas obtenidos, es decir las imágenes de los audios de uno, dos, tres, cuatro y cinco segundos. Analizar los resultados obtenidos y comparar estos con los resultados

VII. CONCLUSIONES

Este trabajo a sido diseñado con el fin de encontrar una red capaz de identificar la voz de un hablante. Para conseguir el objetivo de este proyecto, reconocer al hablante mediante la voz de los participantes usando técnicas de Deep Learning, se realizó un estudio, con sus posteriores experimentos, sobre cómo transformar un audio a una imagen que contiene las características principales de la voz (espectrograma) y luego pasar el problema de reconocimiento de voz a un problema de reconocimiento de imagen. Para ello se a seguido una serie de pasos.

En un principio, se realizaron distintos experimentos variando los parámetros para el entrenamiento, obteniendo modelos con mejores resultados al clasificar cuando se tiene mayores iteraciones y épocas por cada red convolucional utilizada.

Obtenidos los distintos resultados se pudo concluir que es mejor entrenar modelos con 1000 iteraciones y 20 épocas. Por último se usó uno de los varios modelos obtenidos con el entrenamiento para predecir la voz de un determinado participante. El audio de entrada para la clasificación se lo realizó mediante un micrófono.

Asi se consiguio el objetivo de este proyecto, pudiendo concluir que para el reconocimiento de voz no es muy recomendable, las técnicas de reconocimiento de imágenes, debido a la afectación que causan agentes externos como el ruido; lo que causa una variación en las imágenes para el entrenamiento, obteniendo modelos poco eficientes.

Para un modelo más eficiente, se debe incrementar los datos por cada clase (persona), además, de incrementar el número de clases y el número de iteraciones. Esto evidenciará modelos más óptimos.

Una desventaja de esta metodología para el reconocimiento de voz, es que al utilizar estos modelos entrenados con clases (personas) que no se incluyeron en el entrenamiento de este, se obtiene resultados incorrectos, es decir, reconoce voces incorrectamente. Con esto, se puede concluir que un modelo de reconocimiento de voz usando CNN, no es recomendable para sistemas de seguridad, ya que daría falsos positivos (es decir confusiones en la clasificación).

Evidentemente una de las desventajas para el reconocimiento de voz mediante redes neuronales convolucionales es el ruido que se puede presentar. Resulta muy difícil realizar el reconocimiento de voz si el audio no es muy claro o tiene demasiado ruido de fondo.

Los métodos de reconocimiento de voz usando redes neuronales convolucionales son muy eficientes si se usa conjuntamente con imagen que contiene las características principales de la voz (espectrograma).

REFERENCIAS

[1] Fujitsu (2017). *Deep Learning-based Voiceprint Authentication from Very Short Speeches - Fujitsu China*. [online] Fujitsu.com. Available at: http://www.fujitsu.com/cn/en/about/resources/news/press-releases/2017/frdc-0309.html

[2] De Luna Ortega, C., Martínez Romo, J. and Mora González, M. (2006). *Reconocimiento de Voz con Redes Neuronales, DTW y Modelos Ocultos de Markov*. [online] https://www.redalyc.org/articulo.oa?id=94403203. Available at:https://www.researchgate.net/profile/Carlos_Luna-Ortega/publication/235741495_Reconocimiento_de_Voz_con_Redes_Neuronales_DTW_y_Modelos_Ocultos_de_Markov/links/09e4151303d2cb3298000000/Reconocimiento-de-Voz-con-Redes-Neuronales-DTW-y-Modelos-Ocultos-de-Markov.pdf

[3] Camacho Costumero, C. (2019). *Desarrollo de un sistema de reconocimiento de habla natural basado en redes neuronales profundas*. [online] Repositorio.uam.es. Available at: https://repositorio.uam.es/handle/10486/674094

[4] Graham, B. (2014). *Fractional Max-Pooling*. [online] arXiv.org. Available at: https://arxiv.org/abs/1412.6071

[5] Ricco, J. (2017). *What is max pooling in convolutional neural networks?*. [online] https://www.quora.com. Available

at:https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks

[6] Briega, R. (2016). *Redes neuronales convolucionales con TensorFlow*. [online] Relopezbriega.github.io. Available at: https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/

[7] Martínez Mascorro, G. and Aguilar Torres, G. (2013). [online] Ingenius.ups.edu.ec. Available at: https://ingenius.ups.edu.ec/index.php/ingenius/article/view/10.2013.02/211

[8] Lerch, D. (2018). *Introducción al Deep Learning con Keras – Neuron4 – Medium*. [online] Medium. Available at: https://medium.com/neuron4/introducci%C3%B3n-al-deep-learning-con-keras-b51c47560565

[9] Boles, A. and Rad, P. (2017). *Voice biometrics: Deep learning-based voiceprint authentication system - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/7994971/

[10] The SciPy community (2018). *Signal Processing (scipy.signal) — SciPy v1.2.0 Reference Guide*. [online] Docs.scipy.org. Available at: https://docs.scipy.org/doc/scipy/reference/tutorial/signal.html

[11] "The dummy's guide to MFCC", Medium, 2018. [Online]. Available: https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd

[12] [2]"Understanding Audio data, Fourier Transform, FFT, Spectrogram and Speech Recognition", Medium, 2020. [Online]. Available: https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520