Universidad Michoacana de San Nicolás de Hidalgo

Facultad de Ingeniería Eléctrica
División de Estudios de Posgrado

Identificación de Parlantes Independiente del Texto Utilizando Redes Neuronales Convolucionales

TESIS

Que para obtener el grado de
Maestro en Ciencias en Ingeniería Eléctrica

Presenta
Miguel Angel Reynoso Morales

Dr. José Antonio Camarena Ibarrola
Director de Tesis

Morelia, Michoacán Septiembre 2021

Identificación de Parlantes Independiente del Texto Utilizando Redes Neuronales Convolucionales

Los Miembros del Jurado de Examen de Grado aprueban la **Tesis de Maestría en Ciencias en Ingeniería Eléctrica** de ***Miguel Angel Reynoso Morales***

Dr. Félix Calderón Solorio
Presidente del Jurado

Dr. José Antonio Camarena Ibarrola
Director de Tesis

Dr. Juan José Flores Romero
Vocal

Dr. Jaime Cerda Jacobo
Vocal

Dr. Luis Eduardo Gamboa Guzmán
Revisor Externo (FIE UMSNH)

Luis Eduardo Gamboa G.

Dr. Roberto Tapia Sánchez
Jefe de la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica. UMSNH (Por reconocimiento de firmas).

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
Agosto 2021

Este trabajo es dedicado a mi familia, puesto que son las raíces que han sostenido mi existencia.

A mi asesor José Antonio Camarena Ibarrola, por su tiempo y consejos.

A mis maestros, por impulsarme a incursionar en distintos temas, así como por el conocimiento que me han aportado.

A mis compañeros del posgrado, por el tiempo y las experiencias que pasamos.

A la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica, por su excelente trabajo.

Lista de Publicaciones

"Data Augmentation vs Regularization for Time Series Forecasting"
Juan J. Flores, Miguel Reynoso, Josue Gonzalez and Felix Calderon
Publicado en 19^{th} Mexican International Conference of Artificial Intelligence,

MICAI 2020

"Text-Independent Speaker Identification using Formants and Convolutional Neural Networks"
Antonio Camarena-Ibarrola, Miguel Reynoso and Karina Figueroa
Aceptado en 20^{th} Mexican International Conference on Artificial Intelligence,

MICAI 2021

Resumen

La identificación de individuos por medios de su voz abarca dos problemas, la identificación texto-dependiente y la identificación texto-independiente. En la identificación texto-dependiente es necesario que el individuo a identificar pronuncie una frase determinada que cumple con la función de palabra clave. Por otro lado, en la identificación texto-independiente no es necesario que el individuo a identificar pronuncie una frase en específico. Por este motivo la identificación texto-independiente puede utilizarse incluso con una conversación. La flexibilidad de este tipo de identificación también la vuelve más compleja, ya que los sonidos pronunciados no necesariamente son los mismos ni se encuentran en el mismo orden.

En esta tesis se lleva acabo el diseño, la implementación y las pruebas de un método de identificación de parlantes texto-independiente. Este método aprovecha las capacidades que han demostrado tener las redes neuronales convolucionales en el campo de visión por computadora. La propuesta consiste en generar imágenes que representen a los parlantes, para lo cual se usan los formantes de la voz. Estas imágenes posteriormente se emplean en el entrenamiento de a una red neuronal convolucional por medio de aprendizaje supervisado. Los patrones que se generan en las imágenes son analizados por una red neuronal que cuenta con dos etapas. La primer etapa cuenta con capas convolucionales que se encargan de extraer las características y la segunda etapa cuenta con capas densas cuya función es usar las características para identificar al parlante. La implementación se realizó en el lenguaje de programación Python en combinación con TensorFlow y Keras.

La implementación es probada utilizando una base de datos que contiene audios de individuos cuya lengua materna es el español. Las pruebas realizadas consisten en entrenar a la red neuronal utilizando un conjunto de imágenes y probar su comportamiento al utilizar un conjunto de imágenes diferentes a las usadas en el entrenamiento. Los resultados que se obtienen de las pruebas se utilizan para crear gráficas que ayudan en el análisis del comportamiento de la red neuronal, además, se crean matrices de confusión que complementan la información aportada por las gráficas. Los resultados obtenidos fueron buenos ya que se alcanzó un 96 % de accuracy. Por tanto, es posible realizar la identificación de parlantes a través del método propuesto.

Palabras clave: TensorFlow, Keras, español, formantes, imágenes.

Abstract

Speaker identification comprise two problems, text dependent speaker identification and text independent speaker identification. In text dependent speaker identification is necesary that speaker pronounce a certain phrase whose function is that of a keyword. On the other hand, text independent speaker identification does not need that speaker pronounces a specific phrase. For this reason text independent identification can be used with a conversation. Although, its flexibility also makes it more complex, since the words pronounced are not necessarily the same or in the same order.

This thesis presents the design, implementation and tests of a method for text independent speaker identification. This method takes advantage of the capabilities that convolutional neural networks have in the field of computer vision. The proposal consists of generating images that represent the speakers using voice formants. These images are later used to train a convolutional neural network through supervised learning. The paterns generated in the images are analyzed by a network that has two stages. First stage has convolutional layers to extract the characteristics of the images and second stage has dense layer that use the characteristics to identify the speaker. The implementation was done using the combination of Python, TensorFlow and Keras.

The implementation was tested using a database which contains audios of people whose native language is Spanish. The tests consist of training the neural network using a set of images, later evaluate it in terms of accuracy by using a set of images different from those used for training. The results were depicted graphically to help us analyze the behavior of the neural network, besides, confusion matrices were determined to complement the information provided by the graphics. The results obtained were good since 96 % accuracy was achieved. Hence, it is posible to identify the speakers using the proposed method.

Keywords: TensorFlow, Keras, Spanish, formants, images.

Contenido

Lista de Figuras

Lista de Símbolos

$A(z)$	Filtro inverso.
C	Número de clases.
CE	Resultado de la función $cross - entropy$.
E_n	Energía de tiempo corto.
F_s	Frecuencia de muestreo.
G	Ganancia del sistema.
Hz	Unidade de frecuencia del sistema internacional de unidades (Hertz).
$H(z)$	Respuesta a la frecuencia de un filtro
I	Imagen.
$I(z_k)$	Parte imaginaria de la k-ésima raíz.
K	Kernel de convolución.
$R(z_k)$	Parte real de la k-ésima raíz.
$R_n(i)$	i-ésimo coeficiente de Autocorrelación de tiempo corto.
Z_n	Régimen de cruces por cero de tiempo corto.
a	Constante del filtro de pre-énfasis.
b_k	Ancho de banda de la k-ésima raíz.
b	Sesgo de una neurona artificial.
$\boldsymbol{b}$	Vector de sesgos de una capa.
f_k	Frecuencia central de la k-ésima raíz.
s_i	Puntaje de la red a la clase i.
$s[n]$	Señal de voz discreta.
$u[n]$	Excitación del sistema.
$\boldsymbol{w}$	Vector de pesos de una neurona artificial.
$\boldsymbol{W}$	Matriz de pesos.
$w(n)$	Ventana de Hamming.
$\boldsymbol{x}$	Entrada a la red.
y	Salida de una neurona artificial.
z_k	K-ésimo polo complejo conjugado.
α_k	K-ésimo coeficiente del filtro de predicción lineal.

Lista de Abreviaturas

ADN Ácido desoxirribonucleico.

CNN Redes Neuronales Convolucionales (Convolutional Neural Network, por sus siglas en inglés).

CONDUSEF Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros.

ELSDSR Base de datos que contiene muestras de voz en idioma inglés, cuyo nombre significa English Language Speech Database for Speaker Recognition.

FIR Respuesta Finita al Impulso, (Finite Impulse Response, por sus siglas en inglés).

GMM Modelos de Mezclas Gaussianas (Gaussian Mixture Models, por sus siglas en inglés).

HMM Modelos Ocultos de Marcov (Hidden Markov Models, por sus siglas en inglés).

LPC Coeficientes de predicción lineal (Linear Prediction Coefficients, por sus siglas en inglés).

MFCC Coeficientes cepstrales de Mel (Mel Frequency Cepstral Coefficients, por sus siglas en inglés).

MLP Perceptrón Multicapa (Multilayer Perceptron por sus siglas en inglés).

ReLU Función de activación unidad lineal rectificada (Rectified Linear Unit, por sus siglas en inglés).

RGB Es un modelo de color basado en la síntesis aditiva, en el cual se utilizan los colores rojo, verde y azul.

TDSI Identificación de parlantes texto-dependiente (Text Dependent Speaker Identification, por sus siglas en inglés).

TIMIT Es una base de datos que contiene muestras de voz de hablantes de inglés americano de diferentes sexos y dialectos. El significado de TIMIT es Texas Instruments / Massachusetts Institute of Technology.

TISI Identificación texto-independiente (Text Independent Speaker Identification, por sus siglas en inglés).

VQ Cuantificación Vectorial (Vector Quantization, por sus siglas en inglés).

Capítulo 1

Introducción

"Un árbol enorme crece de un tierno retoño. Un camino de mil pasos comienza en un solo paso."

Lao-Tse

1.1. Planteamiento del Problema

La identidad es todo lo que hace que un individuo sea él mismo. En consecuencia, la identidad de un individuo engloba distintos aspectos como el social, el filosófico, el cultural y el biológico [Saks, 1997]. Debido a la complejidad de este concepto es necesario delimitarlo. En el caso de esta tesis se utiliza únicamente el aspecto biológico, por esta razón hemos tomado del ámbito forense su concepción de identidad. En el ámbito forense la identidad se considera como la serie de características que permiten distinguir a un individuo de los demás [Champod and Meuwly, 2000].

Durante la vida de cualquier persona, su identidad juega un papel sumamente importante. Esto se debe a que la identidad es la llave de un individuo para acceder a sus derechos y obligaciones sociales. Aquí es donde radica el problema, ya que se vuelve una necesidad el poder identificar a los individuos. Para satisfacer esta necesidad se han creado diferentes sistemas, aunque no ha sido tarea fácil, puesto que un sistema de identificación de uso cotidiano necesita ser práctico, rápido y que cuente con la aceptación de las personas.

Existen sistemas en los que se opta por una clave personal o una tarjeta para

realizar la identificación. No obstante, estos sistemas tienen problemas como la perdida de la tarjeta de identificación y el olvido de la clave personal, por esto se ha prestado atención a la biometría. La palabra biometría viene del griego bios que significa vida y métron que significa medida. Por tanto, la biometría en el área de la identificación de individuos es la medición de los rasgos del cuerpo humano, así como patrones de comportamiento que hacen a un individuo único [Serratosa, 2008] [Saks, 1997]. A estos rasgos y patrones se les llama rasgos biométricos. Su relevancia radica en el hecho de ser inherentes a las personas, por lo cual no se pierden u olvidan. En la actualidad se puede realizar la identificación utilizando rasgos biométricos como la textura del iris de los ojos [Ma et al., 2003] [Marín et al., 2009], las huellas dactilares [Hong and Jain, 1998], el ADN, el rostro [Shen et al., 2007], e incluso la voz [Rose, 2002].

Para identificar a individuos a través de sus rasgos biométricos es necesario llevar a cabo un proceso en el que se realiza la medición del rasgo biométrico y posteriormente el procesamiento de las mediciones tomadas. Tanto la medición como la forma en que se procesan las medidas pueden variar dependiendo del rasgo biométrico utilizado. No obstante, los sistemas de identificación actuales tienen en común el uso de instrumentos electrónicos para medir los rasgos biométricos (sensores), así como el uso de algoritmos para el procesamiento de la información obtenida.

Los sistemas de identificación necesitan una base de datos cuyo trabajo es almacenar una plantilla por cada individuo registrado en el sistema. Cada una de estas plantillas es generada utilizando las características extraídas al medir el rasgo biométrico, así como algunos datos del individuo. La extracción de características es un proceso para generar una representación compacta tomando información importante del rasgo biométrico medido. Cuando se desea identificar a un individuo, se realiza la medición del rasgo biométrico. Posteriormente, se extraen las características de la medición y se comparan con las de los individuos de la base de datos. En el caso de encontrarse registrado en la base de datos, el sistema indica que se ha detectado un individuo registrado con la suficiente similitud, pero en caso contrario el individuo se considera desconocido [Serratosa, 2008].

Al momento de implementar un sistema de identificación se pueden considerar aspectos como el presupuesto que se tiene, el tiempo que debe demorar el sistema en la

identificación, si el individuo cooperará con el sistema, el nivel de seguridad requerido, etc. Además, es necesario considerar el rasgo biométrico a utilizar ya que cada uno cuenta con ventajas y desventajas. Por ejemplo, puede compararse las huellas dactilares y el ADN, ambos son rasgos biométricos con los que se puede obtener resultados precisos. Sin embargo, el tiempo requerido para la identificación por medio de huellas dactilares es pequeño en comparación con el requerido para identificar utilizando el ADN. Otro aspecto a considerar son los costos donde también es mayor para el ADN, aún así existen situaciones en las que el ADN resulta ser la mejor opción. Una situación común en la que es mejor opción la identificación por ADN se presenta cuando se tiene únicamente algo del material genético del individuo a identificar [Camacho, 2015]. Cada uno de los rasgos biométricos es de utilidad bajo ciertas circunstancias, lo más importante es tener en cuenta las ventajas y desventajas que implica utilizarlos.

En la actualidad, el estudio de la identificación por medio de los rasgos biométricos se sigue profundizando, esto se debe a varios factores como la creación de nuevas formas para burlar los sistemas de identificación. Por ejemplo, Téllez Ortiz explica como es posible burlar un sistema de identificación de huellas dactilares al utilizar algunos materiales para simular una huella dactilar humana [Téllez Ortiz et al., 2020]. En el articulo se agrega un nuevo parámetro a medir cuando se identifica por medio de huellas dactilares, esto para evitar las intrusiones de seguridad. Aunque existen casos de rasgos biométricos que se siguen estudiando debido a que todavía no se encuentra un método que sea considerado lo suficientemente confiable para ser utilizado al identificar individuos.

Esta tesis aborda la identificación de individuos utilizando la voz de los individuos como rasgo biométrico, por esta razón a partir de este momento a los individuos se les llamará parlantes. Los humanos podemos generar sonidos complejos para comunicarnos, estos sonidos son creados al utilizar el aparato fonador[Gallardo, 2013]. Las características de los órganos del aparato fonador como lo son sus dimensiones y su forma hacen única a la voz de cada persona, además con la experiencia los humanos adquirimos patrones de comportamiento al hablar. Por lo que en el momento de realizar la identificación se busca reconocer los patrones producidos por estas características.

Para realizar la identificación de una persona utilizando su voz, es necesario tomar

muestras que en realidad son grabaciones de audio. Por este motivo la identificación por medio de voz es muy práctica, ya que el sensor encargado de capturar la voz es un micrófono. En la actualidad los micrófonos se encuentran en todas partes, puesto que son componentes fundamentales de los teléfonos celulares. Además, no es necesario que el individuo entre en contacto directo con el sensor biométrico (micrófono), lo que hace posible la identificación a distancia. No obstante, la identificación por medio de la voz se enfrenta a problemas como el ruido acústico, que puede enmascarar las características que se están buscando durante el proceso de identificación.

La tarea de identificar parlantes consiste en clasificar muestras de voz que no han sido etiquetadas como pertenecientes a uno de N parlantes de referencia que se encuentran en la base de datos [Chen et al., 1996]. Adicionalmente la identificación de parlantes abarca dos problemas, la identificación de parlantes texto-dependiente (Text Dependent Speaker Identification, TDSI por sus siglas en inglés) y la identificación texto-independiente (Text Independent Speaker Identification, TISI por sus siglas en inglés).

La identificación de parlantes dependiente del texto se caracteriza por la necesidad de que el parlante pronuncie una palabra o frase determinada. Esta característica es importante, ya que las técnicas empleadas en este tipo de identificación toman en cuenta los sonidos que se han de pronunciar en la palabra o frase utilizada, así como el orden en el que se pronuncian los sonidos. De esta manera se tiene la posibilidad de comparar dos elocuciones de la misma palabra para encontrar similitudes que ayuden a identificar al parlante. Por otro lado, se tiene a la identificación de parlantes independiente del texto, cuya principal característica radica en que no es necesario utilizar una palabra o frase en específico. Al no depender de la estructura de la frase, este tipo de identificación puede utilizarse a pesar de que el individuo a identificar no este cooperando. Esto es muy útil, sin embargo, también es el motivo de que sea un tipo de identificación mucho más compleja, ya que no se sabe que sonidos serán pronunciados ni el orden de estos.

En la identificación de parlantes las técnicas de extracción de características comúnmente se basan en el funcionamiento del aparato fonador, aunque en algunos casos se utilizan técnicas que combinan el funcionamiento del aparato fonador y el oído humano. Estas técnicas se utilizan con la intensión obtener información a partir de la voz que sea de utilidad

para la identificación de los parlantes. En el caso de esta tesis se crea un método de identificación de parlantes independiente del texto, por lo se tuvo que abordar la problemática de carecer de una estructura determinada en la oración producida por el parlante. Esto fue muy importante ya que se buscó una técnica de extracción de características que ayudara a compensar este problema.

Las características extraídas son comparadas para obtener la identidad del parlante. Este proceso se puede abordar a través del uso de técnicas como las distancias, las técnicas de agrupación, los modelos probabilistas, las redes neuronales, etc. En el caso de la TISI los modelos probabilistas han obtenido muy buenos resultados. No obstante, en esta tesis se opta por utilizar las redes neuronales artificiales, ya que en los últimos años se han obtenido buenos resultados en distintos campos como la visión por computadora, el procesamiento de lenguaje natural, etc. El método de identificación TISI que se desarrollado busca obtener buenos resultados utilizando una red neuronal artificial.

Las redes neuronales empleadas en esta tesis son las redes neuronales convolucionales. Este tipo de redes se han destacado en el campo de la visión por computadora debido a sus capacidades para extraer información importante de las imágenes. No obstante, las señales de voz son de naturaleza muy distinta, por lo que es necesario desarrollar un método para que a pesar de trabajar con señales de voz, las redes convolucionales trabajen en el área que han demostrado tener buenas capacidades.

1.2. Antecedentes

La identificación de parlantes se ha estudiado desde hace varias décadas, dando como resultado una mayor comprensión de la producción de voz y los factores que hacen a la voz humana un rasgo biométrico. Los trabajos publicados en esta área se centran sus esfuerzos en la extracción de características, así como en los métodos que se pueden emplear para comparar las características.

La extracción de características ha sido abordada desde distintas perspectivas, ya sea tratando de modelar el tracto vocal, ó utilizando el conocimiento acerca del oído humano para obtener información del espectro de frecuencia de la señal de voz. Entre las técnicas que han tenido una mayor relevancia se encuentran la codificación lineal predictiva (Linear Prediction Coding, LPC por sus siglas en inglés) [Makhoul and Wolf, 1972]. Esta técnica surgió de la necesidad de transmitir la señal de voz de por medio de un canal de comunicación sin demandar mucho ancho de banda. Una de sus principales características radica en que se encuentra sumamente relacionada con la síntesis de voz [Rabiner and Schafer, 1978]. Otra de las técnicas que ha sido ampliamente utilizada son los coeficientes cepstrales de Mel (Mel Frequency Cepstral Coefficients, MFCC por sus siglas en inglés), los cuales han tenido una gran relevancia a través de los años y han sido utilizados tanto en la identificación de parlantes como en el reconocimiento de voz [Davis and Mermelstein, 1980]. Además de las anteriormente mencionadas también se ha propuesto el modelado del pulso glotal de las personas [Plumpe et al., 1999], e incluso utilizar los formantes de las vocales [Almaadeed et al., 2016].

En el proceso de comparación de características, se toman las características extraídas de la señal de voz del parlante a identificar y se busca encontrar similitudes con las características utilizadas para crear las plantillas de la base de datos del sistema. Las soluciones más simples utilizan las distancias para comparar características, donde las distancias más comúnmente utilizadas se encuentra la distancia Manhattan y distancia Euclidiana, además de la medida de divergencia coseno. Con el paso del tiempo se abordó el problema de la comparación de características utilizando técnicas más complejas como la Cuantificación Vectorial (Vector Quantization, VQ por sus siglas en inglés), que es una técnica de aprendizaje no supervisado y fue de las primeras empleadas en la TISI [Soong et al., 1987]. Otro de los enfoques que ha sido probado son los métodos estocásticos, dentro de los que se puede encontrar a los Modelos Ocultos de Marcov (Hidden Markov Models, HMM por sus siglas en inglés) [Zheng and Yuan, 1988], [Matsui and Furui, 1994]. Posteriormente en la década de los 90s se introdujeron los Modelos de Mezclas Gaussianas (Gaussian Mixture Models, GMM por sus siglas en inglés) [Reynolds, 1995], que es el algoritmo estocástico más exitoso en el área de identificación de parlantes. En las últimas décadas también se

pueden observar trabajos en los que se usa algún tipo de redes neuronales, esto se debe principalmente a su resurgimiento. Las redes neuronales fueron dejadas de lado durante un tiempo, sin embargo, con el mejoramiento de los algoritmos, el hardware y el incremento en la información disponible para entrenar a las redes neuronales se han convertido en el centro de atención [Jahangir et al., 2020], [Ashar et al., 2020].

En la Universidad Michoacana de San Nicolás de Hidalgo se han realizado trabajos en el área de la identificación de parlante. En donde se pueden encontrar publicaciones de artículos de investigación y tesis. En años recientes se ha publicado una tesis en la que se realiza la identificación de parlantes con un enfoque independiente del texto [Castro, 2019]. En ésta se utiliza los primeros formantes de las vocales para crear polígonos que representen a los parlantes y por medio de la distancia Manhattan se realiza la comparación entre los polígonos. En cuanto a las publicaciones de artículos de investigación, se pueden observar trabajos en los que se utiliza la entropía por bandas para realizar la identificación de parlantes [Camarena-Ibarrola et al., 2017]. Además, dentro de las publicaciones más recientes se encuentra un trabajo en el que se realiza la identificación de parlantes combinando el uso de los entropigramas con las redes neuronales convolucionales [Camarena-Ibarrola et al., 2020].

1.3. Justificación

La identificación de individuos cada vez toma más relevancia en las actividades de la vida cotidiana. Esto se torna evidente cuando notamos que actividades comunes como desbloquear un teléfono celular se llevan a cabo utilizando rasgos biométricos. Los sistemas de identificación son utilizados como parte de sistemas de seguridad, para identificación en tramites oficiales y últimamente en el comercio electrónico. En México el comercio electrónico se está incrementando, lamentablemente a la par se ha incrementado el porcentaje de fraudes cibernéticos. Los datos de la CONDUSEF indican que se incremento la proporción de fraudes cibernéticos de 33 % en el año 2016 a 69 % en el año 2020 y el monto de reclamaciones ascendió a $12,280 millones de pesos [CONDUSEF, 2021].

La identificación de parlantes texto-independiente puede aportar en el proceso de identificar a un individuo, al trabajar en conjunto con otros métodos de identificación o de

manera independiente cuando las situaciones así lo requieran. Como en situaciones en las que se cuenta únicamente con una grabación de audio y es necesario identificar la identidad del individuo de la grabación, ya sea con fines judiciales o en el caso de que se este buscando a una persona perdida. Además, la TISI es un campo de investigación abierto, por lo que sigue en estudio.

1.4. Hipótesis

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) han dado buenos resultados en la tarea de reconocimiento de patrones en imágenes. No obstante, una imagen es una señal con dimensiones espaciales, mientras que una señal de voz es una señal que depende del tiempo. En una señal de voz la información que se puede obtener para caracterizar a los parlantes se encuentra dispersa en una cantidad gigantesca de muestras. Adicionalmente, obtener una buena caracterización de un parlante depende de la duración de la señal y la cantidad de sonidos distintos que son pronunciados por el parlante. Por tanto, se propone generar imágenes que contengan la información más relevante de los audios y utilizarlas para entrenar a una red neuronal convolucional que se encargue de identificar a los parlantes.

En el caso de la TISI no es necesario que el parlante a identificar pronuncie una frase en específico, por lo que se propone obtener información del parlante al tomar los sonidos vocalizados de la señal de voz para extraer los formantes y usarlos como características. Los formantes de la voz son las frecuencias de resonancia de las cavidades del tracto vocal, por lo que pueden ser utilizados para generar representaciones de los parlantes. Por tanto, en este proyecto se propone crear las imágenes utilizando los formantes de la voz, para que de esta manera las imágenes se vuelvan representaciones del tracto vocal.

1.5. Metodología de Investigación

La implementación se lleva a cabo en el lenguaje de programación Python, ya que es uno de los lenguajes de programación más utilizados en las ciencias de datos. El

sistema consta de dos partes, en la primera parte las funciones creadas se encargan de realizar el pre-procesamiento de la señal de voz, la identificación de las secciones con sonido vocalizado, la extracción de los formantes y la generación de las imágenes. Aunque, en el pre-procesamiento no se contempla la limpieza del ruido en las señales de voz. Por otro lado, la segunda parte del sistema es la implementación de la red neuronal, para la que se utiliza la librería de Machine Learning TensorFlow en combinación con Keras.

En las pruebas realizadas a la implementación se utilizó una base de datos que cuenta con 2856 archivos de audio. Los archivos pertenecientes a la base de datos son del tipo Wave que fueron muestreados a una frecuencia de 8000Hz. La base de datos fue creada utilizando a 21 personas, seis mujeres y quince hombres. Cada persona pronunció 34 palabras los números del cero al nueve y las letras del alfabeto griego. Además, cada una de las palabras es pronunciada cuatro veces. Esta base de datos fue creada por el M.C. José Francisco Rico Andrade [Andrade and Ibarrola,], y se puede encontrar en el link: `http://dep.fie.umich.mx/~camarena/dsp/elocuciones21.tar.gz`.

Los resultados obtenidos de la red neuronal son medidos utilizando la métrica accuracy, que consiste en contar la cantidad de imágenes clasificadas de manera correcta por la red y dividirla entre la cantidad de imágenes que clasificó la red. La métrica accuracy toma un valor entre cero y uno, aunque en el caso de este trabajo esta cantidad es mostrada como un porcentaje. Además, se utilizan matrices de confusión para observar de una mejor manera el comportamiento de la red neuronal ante las imágenes de entrada.

1.6. Objetivos de la Tesis

1.6.1. Objetivo general

Desarrollar e implementar un método texto-independiente para identificación de parlantes. El método utiliza los formantes de la voz para generar imágenes que caractericen a los parlantes, así como redes neuronales convolucionales para reconocer los patrones en las imágenes generadas.

1.6.2. Objetivos particulares

- Implementar el módulo encargado de preprocesar los audios.
- Crear el módulo encargado de segmentar los audios en marcos, así como seleccionar los marcos que contienen sonido vocalizado.
- Hacer el módulo que lleve a cabo la tarea de extraer los formantes y sus anchos de banda de los marcos que contienen sonido vocalizado.
- Crear el módulo que a partir de los formantes y sus respectivos anchos de banda genere imágenes.
- Diseñar la arquitectura de la red neuronal.
- Implementar la red neuronal.
- Entrenar la red neuronal.
- Realizar las pruebas que nos permitan evaluar el desempeño del sistema.

1.7. Descripción de Capítulos

- Capítulo 2, se presenta una breve explicación de la señal de voz, así como el método utilizado para llevar a cabo la extracción de los formantes de la voz.
- Capítulo 3, se realiza una breve introducción a las redes neuronales, explicando conceptos básicos, fortalezas, debilidades y los tipos de redes neuronales utilizados en este trabajo.
- Capítulo 4, explica el diseño del sistema que ha sido implementado, detallando la extracción de formantes y la conversión de formantes a imágenes.
- Capítulo 5, describe las pruebas realizadas al sistema y muestra los resultados obtenidos a partir de las pruebas.
- Capítulo 6, plasma las conclusiones a las cuales se ha llegado después de realizar las pruebas y se explican los trabajos futuros.

Capítulo 2

Estimación de los Formantes

"Habla para que yo te conozca."

Sócrates

2.1. Producción de Voz

Los humanos podemos comunicarnos a través de distintos medios, pero uno de los más importantes es el habla. Al comunicarnos por medio del habla transmitimos información a través de sonidos. En una conversación se pronuncian palabras, las cuales llevan el mensaje que se quiere transmitir. No obstante, parte importante del mensaje es expresado a través de detalles como la manera en la que se pronuncian las palabras o incluso la entonación usada. Además, los sonidos producidos por una persona al hablar también aportan información que permite reconocer su identidad, su estado de ánimo y su sexo, esto es posible debido a la complejidad del aparato fonador [Scivetti, 2007].

Para producir sonidos los humanos utilizamos el aparato fonador, que está compuesto por el aparato respiratorio, el aparato laríngeo y el aparato resonador [Gallardo, 2013]. En la Figura 2.1 se muestra el aparato fonador dividido en tres secciones. Al hablar los pulmones con la ayuda del diafragma generan un flujo de aire que pasa por la traquea y llega hasta la laringe. En la laringe se encuentran las cuerdas vocales, que son pliegues flexibles que vibran al obstruir el paso del flujo de aire. Posteriormente, las cavidades supraglóticas se encargan de darle forma a las vibraciones producidas por las cuerdas vocales [Torres, 2007].

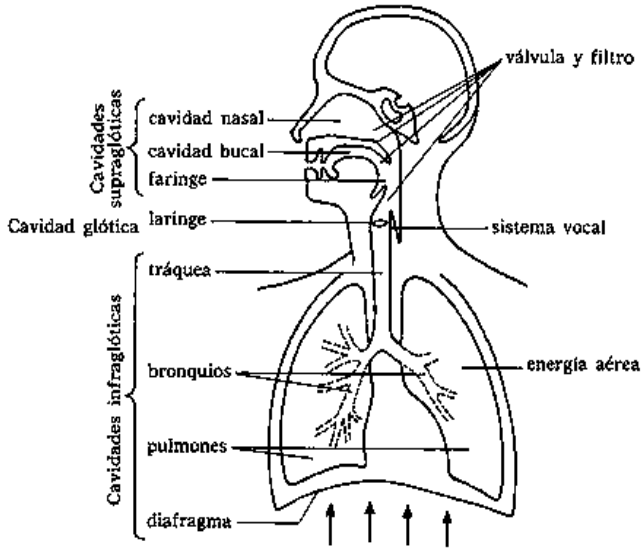


Figura 2.1: Aparato Fonador (tomada de [Martínez Celdrán, 1984]).

La mayoría de los idiomas pueden ser descritos en términos de un conjunto de sonidos distintivos conocidos como fonemas [Rabiner and Schafer, 2010]. Los fonemas cumplen con el trabajo de proveer un enlace entre el lenguaje escrito y la señal de voz correspondiente [Rabiner and Schafer, 2010]. Los sonidos del habla pueden clasificarse como sonidos consonánticos y sonidos vocálicos. Los sonidos consonánticos se producen cuando se realiza una obstrucción al flujo de aire de los pulmones, mientras que durante la producción de los sonidos vocálicos el flujo de aire pasa libremente por la cavidad bucal [Zhang, 2016]. En el idioma español sólo tenemos cinco sonidos vocálicos distintos y menos de 20 sonidos consonánticos [Hualde et al., 2010], en tanto que en el inglés de los Estados Unidos existen entre 39 y 48 fonemas incluyendo vocales, diptongos, semivocales y consonantes [Rabiner and Schafer, 2010].

Los sonidos consonánticos se clasifican de acuerdo al punto de articulación, el modo de articulación y la actividad de las cuerdas vocales [Hualde et al., 2010]. Cuando una consonante es clasificada de acuerdo a su punto de articulación, se debe de tomar en cuenta el articulador activo y el articulador pasivo. Por ejemplo, la consonante p es clasificada como bilabial, esto se debe a que para pronunciarla el labio inferior toca al labio

superior y por un instante se obstruye el paso de aire. En el caso anterior el labio inferior es el articulador activo y el labio superior es el articulador pasivo. Por otro lado, cuando la consonante es clasificada de acuerdo al modo de articulación, se considera la forma en la que se produce el sonido. Por ejemplo, las consonantes oclusivas se caracterizan por un bloqueo total del paso de aire seguido de una liberación repentina (consonantes p, t, k), mientras que las consonantes fricativas se caracterizan por el acercamiento del articulador activo con el articulador pasivo sin que se interrumpa el flujo de aire (consonantes s, f, x). Por último, cuando se clasifican las consonantes por la actividad de las cuerdas vocales únicamente se tiene que considerar si las cuerdas vocales vibran durante la producción de la consonante. Las consonantes en las que participan las cuerdas vocales se les llama sonoras (consonantes b, d, g), por otra parte cuando esto no sucede se les llama sordas (consonantes p, t, k).

En cuanto a las vocales, estas se clasifican a través de tres parámetros. Los primeros dos están relacionados con la posición de la lengua dentro de la boca. La altura y el desplazamiento hacia la parte anterior o posterior. Mientras que el tercero está relacionado con la posición de los labios. Por ejemplo, de acuerdo a la altura se tienen vocales altas (i, u), medias (e, o) y una vocal baja (a), de acuerdo a su desplazamiento se tienen vocales anteriores (i, e), una vocal central (a) y vocales posteriores (o, u). Mientras que cuando se considerar la posición de los labios, se tienen vocales redondas (o, u) y no redondas (i, e, a) [Hualde et al., 2010].

2.2. Formantes

Cuando las vibraciones provenientes de las cuerdas vocales atraviesan el tracto vocal algunas secciones del espectro de frecuencia se acentúan, mientras otras secciones se atenúan. Esto se debe a los efectos de la resonancia acústica dentro del tracto vocal [Aalto et al., 2018]. Los efectos de este proceso se pueden apreciar como la formación de picos en el espectro de frecuencias. En la Figura 2.2 se muestra el espectro de frecuencia cubierto por la envolvente espectral, donde los picos de la envolvente se les llama formantes. El nombre de los picos es dado debido a que estas resonancias son las que se encargan de darle forma a la voz, en particular le dan color y forma a las vocales [Rabiner and Schafer, 2010].

Los formantes se identifican de acuerdo a su frecuencia, siendo el primer formante el de menor frecuencia y el quinto el de mayor frecuencia. Sin embargo, no todos los picos de la envolvente espectral de la voz están relacionados con las frecuencias de resonancia del tracto vocal. Algunos de los picos pueden ser provocados por las propiedades acústicas fuera del tracto vocal [Aalto et al., 2018].

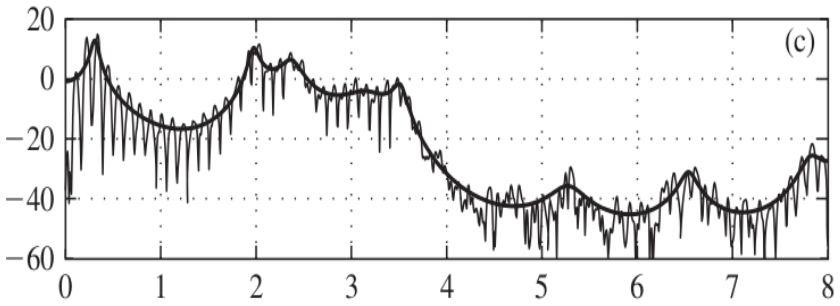


Figura 2.2: Cuando se suaviza el espectro de frecuencia se obtiene la envolvente espectral, donde resaltan los picos más importantes del espectro de frecuencia (tomada de [Rabiner and Schafer, 2010]).

Las frecuencias de los formantes dependen de factores como la frecuencia a la que vibran las cuerdas vocales, la posición de la mandíbula, así como las dimensiones y forma de las cavidades de resonancia [Aalto et al., 2018] [Latorre et al., 2009]. Esto implica que la frecuencia de los formantes varía incluso cuando se pronuncia el mismo fonema, lo que se puede notar al analizar las vocales. La identificación de las vocales puede llevarse a cabo utilizando sus primeros dos formantes, aunque, al identificarlas se considera que cada vocal es representada por un dominio con limites amplios [Celdrán et al., 1995]. Esto se puede observar en la Figura 2.3 [Castro, 2019] donde se muestran las regiones vocálicas del español. En la figura se utilizan los primeros dos formantes para identificar las vocales pronunciadas, y se muestra como cada vocal no es representada por un punto, en su lugar es una zona del espacio la que la representa.

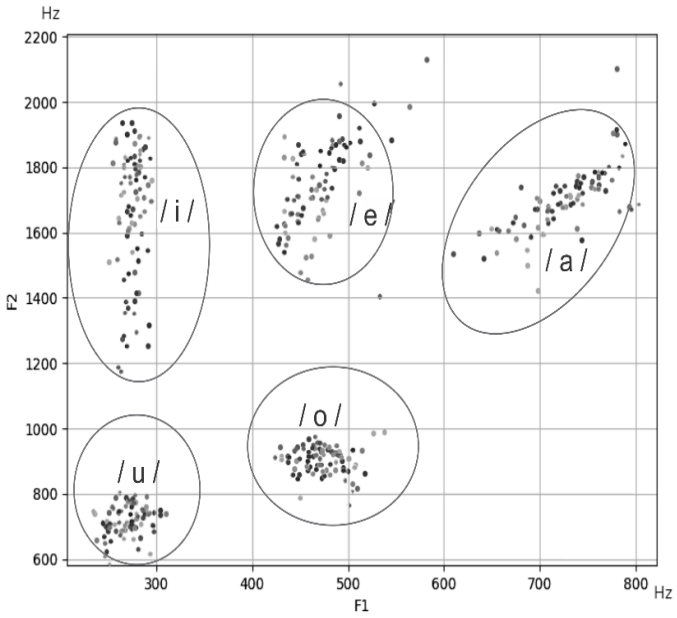


Figura 2.3: Cada vocal del idioma español es representada por una región en el espacio formado por los primeros dos formantes (tomada de [Castro, 2019]).

Existen varios métodos para estimar los formantes, aunque, en el caso de este trabajo se estiman utilizando la codificación lineal predictiva. El proceso que se lleva a cabo consiste de cuatro etapas. En la primera se realiza el preprocesamiento de la señal de voz, que consiste en tratar la señal de audio y dividirla en pequeños fragmentos llamados marcos. Posteriormente, en la segunda etapa se identifican los marcos que contienen sonido vocalizado. En la tercera, los marcos que contienen sonido vocalizado son utilizados para obtener los coeficientes LPC. Por último, se utilizan los coeficientes LPC para identificar los formantes, por lo que se calcular su ancho de banda y su frecuencia.

2.3. Preprocesamiento de la señal de voz

En el caso de este trabajo la señal de voz se encuentra almacenada en archivos de audio en formato wave, en este formato se almacena una versión digitalizada de la señal de voz. La señal digitalizada consiste de un conjunto de muestras tomadas de la señal de voz periódicamente. Las señales digitalizadas pasan por un preprocesamiento antes de llegar a

ser utilizadas para la estimación de los formantes. Por lo que durante el preprocesamiento se busca el inicio y el final de la señal de voz en el archivo. En la Figura 2.4 se muestra la gráfica de un audio en la que se puede notar que la señal de voz ocupa sólo una parte. En el proceso para identificar el inicio y el final de la sección con voz se utilizan dos herramientas, el régimen de cruces por cero de tiempo corto y la energía de tiempo corto. Una vez que se tiene identificado el inicio y el final de la elocución se procede a aplicar el filtro de pre-énfasis y posteriormente la señal filtrada es dividida en marcos.

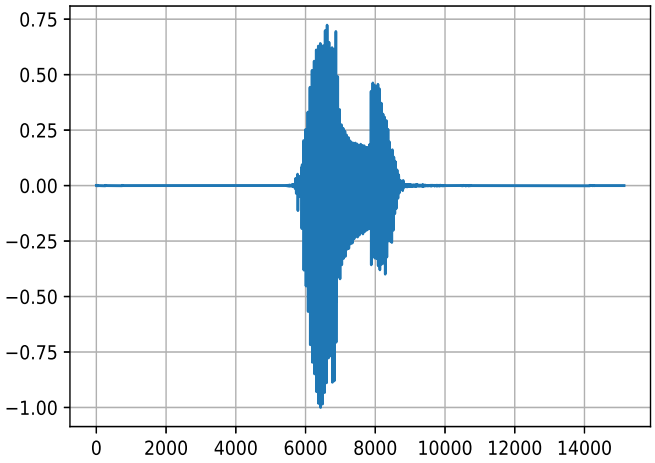


Figura 2.4: En la gráfica se muestra la señal de un archivo de audio donde solo una sección entre la muestra 5800 y la muestra 9000 contiene voz.

2.3.1. Régimen de cruces por cero de tiempo corto

Los cruces por cero ocurren cuando muestras sucesivas de la señal de audio tienen signo distinto [Rabiner and Schafer, 2010]. El régimen de cruces por cero de tiempo corto es una medida de frecuencia en una señal. Las secciones de audio que contienen voz normalmente tiene un régimen inferior que las secciones del audio que no contienen voz [Rabiner and Schafer, 2010]. Por tanto, el régimen de cruces por cero puede ser utilizado para diferenciarlas. El proceso consiste en tomar pequeños fragmentos del audio y calcular su régimen de cruces por cero, que es comparado con un umbral para identificar si la sección contiene voz. En la Ecuación 2.1 se muestra como se calcula el régimen de cruces por cero

de una sección del audio de N muestras.

$$Z = \frac{1}{2N} \sum_{i=1}^{N} |\,\text{signo}(x[i]) - \text{signo}(x[i-1])| \tag{2.1}$$

En la Ecuación 2.1 Z se le conoce como régimen de cruces por cero de tiempo corto. Por su parte en la Ecuación 2.2 se describe la función signo, que recibe una muestra y si es mayor o igual que 0 regresa 1, en caso contrario regresa -1. De manera que en la Ecuación 2.1 cuando las muestras tienen el mismo signo el resultado es cero, pero en caso contrario el resultado es dos, por este motivo el resultado del sumatorio es divido entre dos.

$$\text{signo}\,(x[i]) = \begin{cases} 1 & x[i] \geq 0 \\ -1 & x[i] < 0 \end{cases} \tag{2.2}$$

2.3.2. Energía de tiempo corto

La energía de tiempo corto (E) es la energía de un fragmento de una señal. En la Ecuación 2.3 se muestra como para calcularla es necesario sumar el cuadrado de cada muestra de un fragmento de tamaño N [Rabiner and Schafer, 2010]. La energía de tiempo corto es de ayuda en este caso para diferenciar los fragmentos que contienen voz, ya que los fragmentos que contienen voz tienen más energía que los fragmentos que no la contienen.

$$E = \sum_{i=1}^{N} x^2[i] \tag{2.3}$$

2.3.3. Filtro de pre-énfasis

El filtro de pre-énfasis es un filtro del tipo FIR (Finite Impulse Response, FIR por sus siglas en inglés), que solo cuenta con un cero. Este filtro se aplica en caso de que durante el modelado del tracto vocal en el análisis de predicción lineal no se considere la influencia de los labios en la producción de voz [Markel and Gray, 2013]. Cuando este filtro es aplicado a la señal de voz se realiza una acentuación de las altas frecuencias. Por su parte, la Ecuación 2.4 describe la respuesta en frecuencia del filtro, donde a la constante a puede tomar un valor entre 0.9 y 1.0. En el caso de este proyecto se opto por asignar a la constante a un valor de 0.98.

$$H(z) = 1 - az^{-1} \tag{2.4}$$

En la ecuación $H(z)$ representa la respuesta del filtro a la frecuencia, y z es la variable compleja de la transformada z.

2.3.4. División del audio en marcos

Posterior al filtrado de pre-énfasis la señal es dividida en marcos, esto significa que se toman pequeños fragmentos de la señal de voz. En esta tesis se toman fragmentos de 30ms de audio, que equivale a 240 muestras de un audio muestreado a 8000Hz. La división del audio en marcos se lleva a cabo al aplicar una ventana a la señal, tomando todo lo que se encuentre dentro de la ventana y descartando todo lo que se encuentre fuera de ella [Rabiner and Schafer, 2010]. Esta ventana es desplazada por toda la señal de audio. El desplazamiento suele ser de un tercio del tamaño de la ventana, a 10ms de audio que equivale a 80 muestras. Esto se realiza con la finalidad de aumentar la resolución en tiempo sin sacrificar resolución en frecuencia.

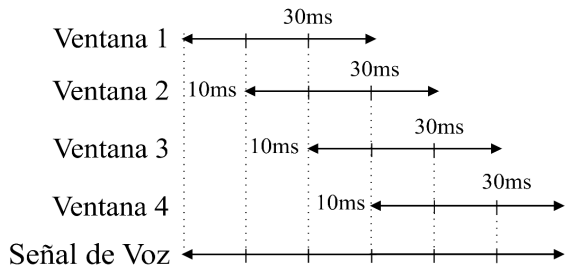
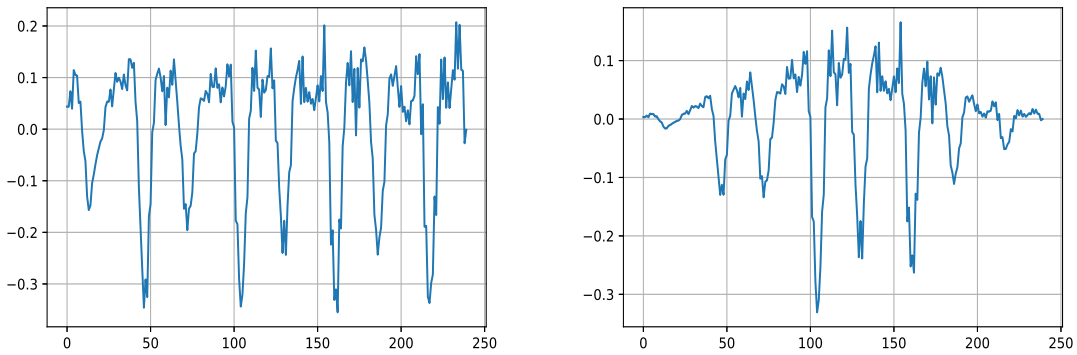


Figura 2.5: La señal de voz es dividida utilizando una ventana que toma fragmentos de 30ms y es desplazada por toda la señal cada 10ms (tomado de [Castro, 2019]).

En el caso de este trabajo se aplica la ventana de Hamming (Ecuación 2.5). El motivo por el que se utiliza la ventana de Hamming es para atenuar las discontinuidades en los extremos del marco, que se crean al segmentar la señal de audio. En la Ecuación 2.5 n representa al índice de la ventana y N es el tamaño de la ventana en muestras.

$$w[n] = 0.54 + 0.46 \ \cos\left(\frac{2\pi n}{N}\right) \tag{2.5}$$

En la Figura 2.6 a) se muestra un marco de la señal antes de aplicar la ventana de Hamming y en la Figura 2.6 b) se muestra el resultado de aplicar la ventana de Hamming al marco. Como se puede observar las discontinuidades en los extremos de la señal fueron atenuadas.



(a) ***Marco sin ventana de Hamming*** **(b)** ***Marco con ventana de Hamming***

Figura 2.6: En la imagen a) se muestra como se generan discontinuidades en los extremos al tomar un marco sin utilizar la ventana de Hamming. Mientras que en la imagen b) se muestra como al aplicar la ventana de Hamming las discontinuidades son atenuadas

2.4. Identificación de sonido vocalizado

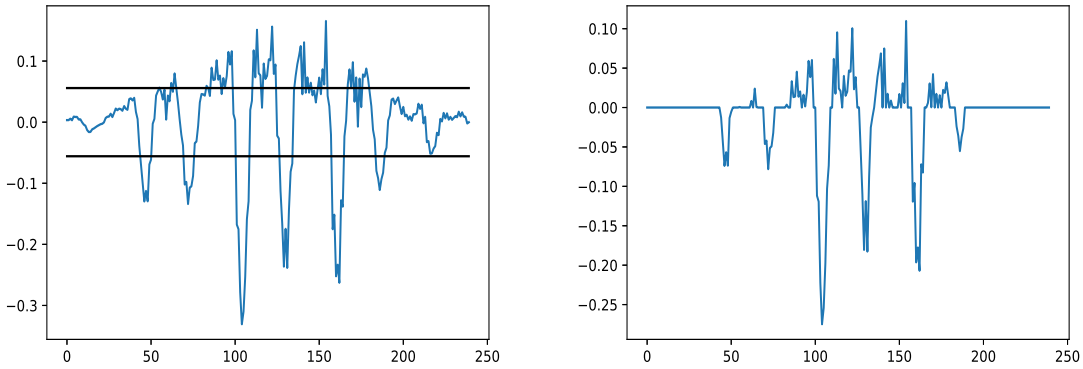
Los formantes son extraídos de los marcos que contiene sonido vocalizado, por lo que es necesario identificar este tipo de marcos entre todos los que se han obtenido. Los sonidos vocalizados son cuasi-periódicos, teniendo como periodo el tono de la voz, mientras que los sonidos no vocalizados carecen de esta propiedad [Rabiner and Schafer, 2010]. Por tanto, para identificar los marcos con sonido vocalizado se busca obtener el tono de la señal de voz contenida en los marcos, en caso de que el marco carezca de sonido vocalizado no se encontrará periodicidad.

En esta tesis para obtener el tono se utiliza la autocorrelación, el proceso consiste en calcular la autocorrelación de un marco y buscar el pico más grande, en el caso de que el tamaño del pico supere un umbral establecido puede considerarse que el marco contiene sonido vocalizado. Sin embargo, en ocasiones se pueden generar picos de autocorrelación

más grandes que el pico del tono, por lo que también se utiliza un aplanador de espectro que ayuda atenuando los picos que no corresponden al tono.

2.4.1. Aplanador de espectro

Un aplanador de espectro hace más prominente la periodicidad mientras que se encarga de suprimir otras características de la señal [Rabiner and Schafer, 2010]. Existen varios aplanadores de espectro, pero en este trabajo se utiliza el Center Clipping. Como se muestra en la Figura 2.7, el Center Clippling realiza un recorte justo al centro de la señal de audio dejando únicamente los picos relevantes.



(a) ***Marco sin recorte*** **(b)** ***Marco con recorte***

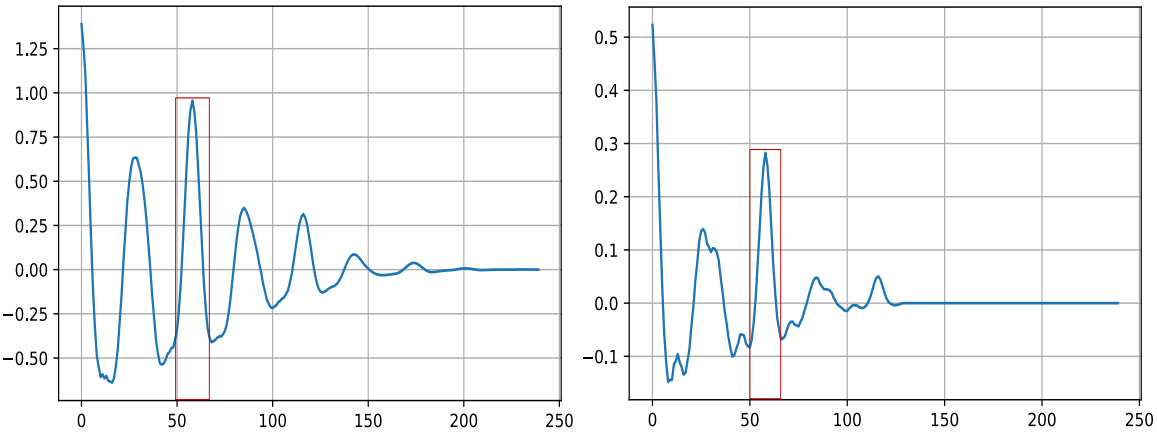
Figura 2.7: En la figura a) se muestra la señal de un marco con líneas que marcan la sección donde se realizará el recorte al centro. En la Figura b) se muestra el resultado de aplicar el recorte al centro.

2.4.2. Autocorrelación

Después de aplicar el recorte al centro a la señal de audio se procede a obtener la autocorrelación, que es definida en la Ecuación 2.6. A través de la autocorrelación se pueden observar propiedades de la señal como la energía total de ésta, aunque la más importante es la periodicidad [Rabiner and Schafer, 2010].

$$\phi[k] = \sum_{m=-\infty}^{\infty} x[m]x[m+k] \tag{2.6}$$

En la Figura 2.8 a) se muestra la autocorrelación de un marco al que no se le aplicó el recorte al centro, y como se puede observar existen picos que se acercan en tamaño al pico del tono. Mientras que en la Figura 2.8 b) si se aplico el recorte al centro y se puede observar como resalta más el pico correspondiente al tono.



(a) ***Autocorrelación del marco sin recorte*** **(b)** ***Autocorrelación del marco con recorte***

Figura 2.8: Al observar las figuras se puede comparar el resultado de la autocorrelación al utilizar el recorte al centro. En la figura a) se muestra el resultado de calcular la autocorrelación sin el recorte al centro. Mientras que en b) se puede observar como el pico del tono ubicado en la muestra 56 es acentuado y los demás picos atenuados.

2.5. Análisis de Predicción Lineal

El análisis de predicción lineal es una de las técnicas más utilizadas en el procesamiento de señales de voz [Rabiner and Schafer, 2010]. Esta técnica se basa en el modelado del proceso que se lleva a cabo durante la producción de voz. La voz es producida como resultado de la excitación del tracto vocal, el cual cambia de forma a través del tiempo [Makhoul and Wolf, 1972]. Debido a las variaciones del tracto vocal es necesario tomar fragmentos pequeños de la señal de voz, en los que la forma del tracto vocal se encuentra

estática, esto permite que el tracto vocal sea modelado por un filtro lineal. En la Figura 2.9 se esquematiza de forma simple el proceso de producción de voz. En el esquema, el filtro $H(z)$ realiza el trabajo del tracto vocal, por lo que sus parámetros α cambian con el tiempo. Además, dependiendo de sí el sonido es vocalizado o no, el filtro es excitado por una entrada $u[n]$ que puede ser un tren de impulsos con el periodo del tono de voz o una señal de ruido aleatorio, ambos tipos de excitación son amplificadas por una ganancia G antes de entrar al filtro. Por su parte, la salida s representa la señal de voz producida. En el análisis de predicción lineal se busca obtener los parámetros del modelo de producción de voz a partir de la señal de voz. Esta técnica fue originalmente creada para transmitir voz con un bajo costo, ya que en lugar de enviar la señal completa se puede enviar solo los parámetros [Rabiner and Schafer, 2010]. No obstante, debido a que se modela el tracto vocal, también puede ser utilizada para extraer características como los formantes de la voz [Markel and Gray, 2013].

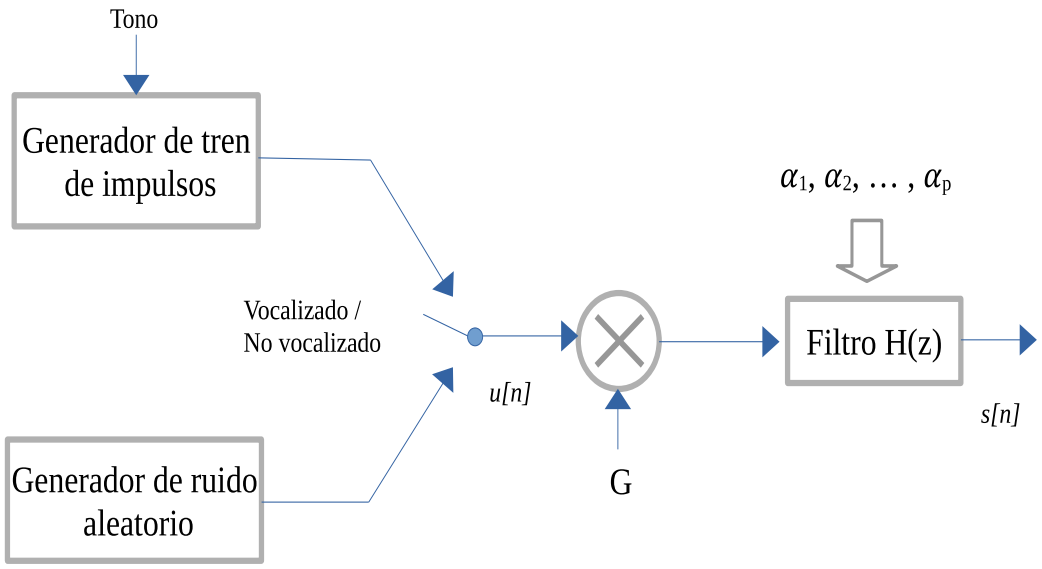


Figura 2.9: En el esquema se muestra una simplificación del proceso de producción de voz, donde dependiendo si se va a producir un sonido vocalizado se utiliza un tren de impulsos o en caso contrario ruido aleatorio. En el esquema $H(z)$ cumple con el papel del tracto vocal y $s[n]$ es la señal de voz resultante.

El filtro $H(z)$ es un filtro de puros polos, cuya respuesta en frecuencia está dada por la Ecuación 2.7. En esta ecuación p es el orden del filtro, z es la variable compleja de la transformada z y α_k es el k-ésimo coeficiente del filtro.

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{2.7}$$

El funcionamiento del sistema que se muestra en el esquema de la Figura 2.9 es descrito por la Ecuación 2.8, que es la ecuación en diferencias que se obtiene cuando se aplicar la transformada z inversa al sistema. En esta ecuación las muestras de la señal se generan utilizando p muestras anteriores, donde p también es el orden del filtro predictor $H(z)$.

$$s[n] = \sum_{k=1}^{p} \alpha_k s[n-k] + Gu[n] \tag{2.8}$$

A los coeficientes del filtro $H(z)$ se les llama coeficientes de predicción lineal (LPC, por sus siglas en inglés) y parte importante del análisis de predicción lineal consiste en el cálculo de estos. En este trabajo se utiliza la autocorrelación para calcular los coeficientes LPC, lo que es posible debido a que la autocorrelación contiene gran cantidad de información de la señal de voz [Rabiner and Schafer, 2010]. Para calcular los coeficientes LPC se recurre a la Ecuación 2.9, que se obtiene al minimizar la diferencia entre las muestras generadas por el filtro y las muestras del audio [Markel and Gray, 2013]. En esta ecuación R_n representa a la autocorrelación de tiempo corto, ya que se calcula a partir de pequeños fragmentos de la señal, por lo que el subíndice n hace referencia al número de marco del que ha sido extraída. La ecuación expresa que el coeficiente i de autocorrelación es igual al sumatorio de la multiplicación de los coeficientes α_k por los coeficientes $|i-k|$ de la autocorrelación.

$$\sum_{k=1}^{p} \alpha_k \; R_n[|i-k|] = R_n[i] \qquad i = 1, 2, 3, \ldots, p \tag{2.9}$$

La Ecuación 2.9 también puede ser expresada en forma matricial (Ecuación 2.10), lo importante de esta forma es que se puede observar que el sistema de ecuaciones a resolver forma una matriz del tipo Toeplitz. Este tipo de matrices se caracterizan por ser matrices simétricas cuyos valores a lo largo de cualquier diagonal son iguales. Los sistemas de ecuaciones que forman matrices Toeplitz pueden ser resueltos de manera mas rápida a través del algoritmo de Levinson-Durbin [Rabiner and Schafer, 2010].

$$\begin{bmatrix} R_n[0] & R_n[1] & R_n[2] & \ldots & R_n[p-1] \\ R_n[1] & R_n[0] & R_n[1] & \ldots & R_n[p-2] \\ R_n[2] & R_n[1] & R_n[0] & \ldots & R_n[p-3] \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ R_n[p-1] & R_n[p-2] & R_n[p-3] & \ldots & R_n[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n[1] \\ R_n[2] \\ R_n[3] \\ \vdots \\ R_n[p] \end{bmatrix} \tag{2.10}$$

El algoritmo de Levinson-Durvin se caracteriza por resolver un sistema de ecuaciones que forma una matriz de Toeplitz de forma recursiva al realizar la resolución de sistemas más pequeños [Rabiner and Schafer, 2010]. El algoritmo recibe el vector con los valores de autocorrelación (R_n), así como el orden del filtro predictor (p). y se compone por tres partes. En la linea 3 se lleva a cabo la primera, en ésta se calcula el coeficiente k. La segunda parte abarca de la linea 3 a la linea 8, donde se calculan los coeficientes α para el sistema de orden i. Mientras que en la tercera parte (linea 9) se calcula el error que ha de ser utilizado en la siguiente iteración del ciclo. Al finalizar el algoritmo regresa los valores de α para el orden p.

Algoritmo 1 Levinson-Durbin

Entrada: R_n, p
Salida: α

$E^{(0)} \leftarrow R_n[0]$
para $i \leftarrow 1, p$ **hacer**
  $k_i \leftarrow \left(R_n[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_n[i-j] \right) / E^{(i-1)}$
  $\alpha_i^{(i)} \leftarrow k_i$
  si $i > 1$ **entonces**
    para $j \leftarrow 1, i-1$ **hacer**
      $\alpha_j^{(i)} \leftarrow \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$
    fin para
    $E^{(i)} \leftarrow (1 - k_i^2) E^{(i-1)}$
  fin si
fin para
$\alpha \leftarrow \alpha_j^{(p)} \;\; j = 1, 2, \ldots, p$
devolver α

2.6. Estimación de los Formantes

Los coeficientes del filtro de predicción lineal $H(z)$ son los coeficientes LPC, que se obtienen a partir de la señal de voz. Por lo que los parámetros del filtro contienen información de la respuesta en frecuencia del tracto vocal, lo que nos permite identificar información como los máximos en el espectro de frecuencias. En la Ecuación 2.11 se puede observar $A(z)$ que representa al polinomio conformado por los coeficientes LPC del filtro de predicción. Para estimar los formantes es necesario calcular las raíces del polinomio $A(z)$, por lo que se procede a igualar a cero y buscar sus raíces [Makhoul and Wolf, 1972].

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \tag{2.11}$$

Debido a que los coeficientes del polinomio son reales se pueden obtener algunas raíces reales y el resto son pares complejos conjugados [Makhoul and Wolf, 1972]. Las raíces del polinomio que se generan por los formantes son polos complejos conjugados, de manera que pueden ser representadas como se muestra en la Ecuación 2.12. Sin embargo, no todos

los polos complejos conjugados son generados por los formantes, esto se debe a que algunas raíces captan otros aspectos del espectro de frecuencia [Makhoul and Wolf, 1972].

$$z_k = e^{-\sigma_k T} e^{\pm j2\pi F_k T} \tag{2.12}$$

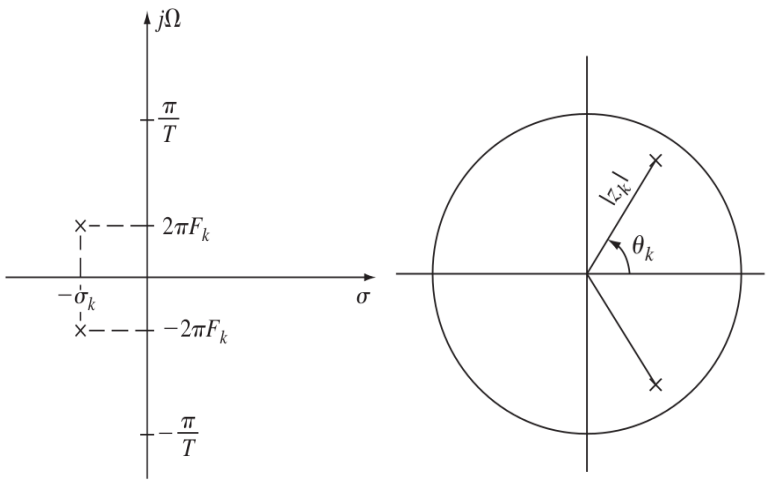
donde:

z_k	k-ésima raíz compleja conjugada
σ	factor de amortiguamiento
F_k	frecuencia de la k-ésima raíz
T	periodo de muestreo

Las raíces del polinomio $A(z)$ que pueden ser consideradas como formantes suelen tener rasgos que las diferencian [Makhoul and Wolf, 1972], como los siguientes:

- Un formante es representado por un par de polos complejos conjugados.
- Los formantes comúnmente tienen anchos de banda pequeños con respecto a sus frecuencias centrales. Por lo que los polos conjugados que tienen anchos de banda amplios se consideran como aportaciones al espectro en general.
- Los rangos de frecuencia de un formante en particular son conocidos.
- Entre marcos continuos los valores de los formantes suelen mantenerse. Aunque, debe tomarse en cuenta que los formantes pueden realizar transiciones rápidas.

Los formantes son representados por polos complejos conjugados esto se debe a que cada formante es un pico en el espectro de frecuencias y por tanto un máximo. La magnitud de los máximos está íntimamente relacionada con el factor de amortiguamiento, ya que este factor rige la participación de la raíz en la formación del espectro. De acuerdo con Rabiner el ancho de banda en el contexto analógico (plano-s) es 2σ [Rabiner and Schafer, 2010]. Por lo que al trasladarlo al contexto discreto (plano-z) se llega a que el ancho de banda está dado por el radio desde el origen al polo, lo que se puede observar en la Figura 2.10.



(a) *plano-s* **(b) *plano-z***

Figura 2.10: En la figura a) se muestra la ubicación en el plano s de un par de polos complejos conjugados que representan a una resonancia del tracto vocal. Mientras que en la figura b) se muestra su equivalente en el plano complejo z, (imágenes tomadas de [Rabiner and Schafer, 2010]).

La Ecuación 2.13 es utilizada para calcular el ancho de banda. En esta ecuación $|z_k|$ es la magnitud de la raíz, b_k es el ancho de banda de la k-ésima raíz, además el resultado del logaritmo natural se multiplica por la división entre la frecuencia de muestreo F_s y π, esto se lleva a cabo para obtener el resultado en Hertz [Vallabha and Tuller, 2002].

$$b_k = -\left(\frac{F_s}{\pi}\right)\ln(|z_k|) \tag{2.13}$$

Por otro lado, la frecuencia central en el contexto analógico está dada por la posición de la raíz con respecto al eje imaginario, por lo que en el contexto discreto está dada por el ángulo θ_k, que se forma por el eje real y el radio de la raíz (Figura 2.10). De esta manera para calcular la frecuencia central se utiliza la Ecuación 2.14 [Vallabha and Tuller, 2002], donde $I(z_k)$ es la parte imaginaria de la raíz y $R(z_k)$ representa la parte real de la raíz.

$$f_k = \left(\frac{F_s}{2\pi}\right) \tan^{-1}\left(\frac{I(z_k)}{R(z_k)}\right) \tag{2.14}$$

Una vez que se han obtenido el ancho de banda y la frecuencia central de las raíces, se procede a identificar cuales representan a los formantes. Para su identificación utilizamos una propiedad de las raíces que representan a los formantes, ésta indica que los formantes comúnmente tienen anchos de banda pequeños con respecto a sus frecuencias centrales. Por tanto, se calcula la razón entre el ancho de banda y su frecuencia central, después el resultado pasa por un umbral. Por lo que todas las raíces que tengan una razón menor que 0.5 son consideradas como formantes.

2.7. Conclusiones del capítulo

Los formantes de la voz están íntimamente relacionados con las características del tracto vocal, así como el proceso de producción de voz, por lo que aportan información tanto de la estructura del tracto vocal como de los patrones de comportamiento de las personas al hablar. Esto permite obtener información suficiente para realizar la identificación de parlantes independiente del texto.

Capítulo 3

Nociones de Redes Neuronales

"Aprender es olvidar los detalles tanto como recordar las partes importantes."

Pedro Domingos

El capítulo comienza con una breve explicación de uno de los órganos más fascinantes del cuerpo humano, el cerebro. Este órgano ha sido la fuente de inspiración para el desarrollo de las redes neuronales artificiales. El primer modelo neuronal contemplaba únicamente a una neurona artificial, aunque con el paso del tiempo se han desarrollado algoritmos en los que el trabajo en conjunto de cantidades enormes de neuronas artificiales han aportado buenos resultados en distintos campos. Esto ha desencadenado el estudio del aprendizaje profundo. No obstante, parte fundamental en el campo del aprendizaje profundo es la estructura en la que se organizan y trabajan las neuronas. De aquí es donde surgen algoritmos como las redes neuronales convolucionales cuyas aportaciones en el campo de visión por computadora son notables. Por último, se aborda el proceso de aprendizaje de las redes neuronales, donde se hace mención de los paradigmas, así como las consideraciones que deben de ser tomadas en cuenta cuando se trabaja con esta familia de algoritmos.

3.1. Neurona artificial

El cerebro humano nos permite tener la capacidad de respirar, realizar movimientos, leer y pensar, esto es posible debido a su increíble complejidad. Un cerebro humano está compuesto por una enorme cantidad de neuronas que interactúan entre ellas a través de una intrincada red de conexiones. Aproximadamente un cerebro tiene 10^{11} neuronas y cada una de ellas a su vez tiene aproximadamente 10^4 conexiones [Hagan et al., 2014]. En la Figura 3.1 se muestra un esquema con un par de neuronas biológicas. En él cada neurona está compuesta por tres partes, las dendritas, el cuerpo de la neurona y el axón. Las dendritas son ramificaciones de redes de fibras nerviosas pertenecientes a la neurona, las cuales se encargan de recibir señales eléctricas del axón de otra neurona y llevarlas al cuerpo de la célula. El cuerpo de la célula se encarga de sumar las señales de entrada y aplicarles un umbral de activación. Mientras que el axón es una fibra larga que lleva la seña de salida desde el cuerpo de la neurona a las dendritas de otras neuronas. A la conexión entre el axón de una neurona y la dendrita de otra se le llama sinapsis.

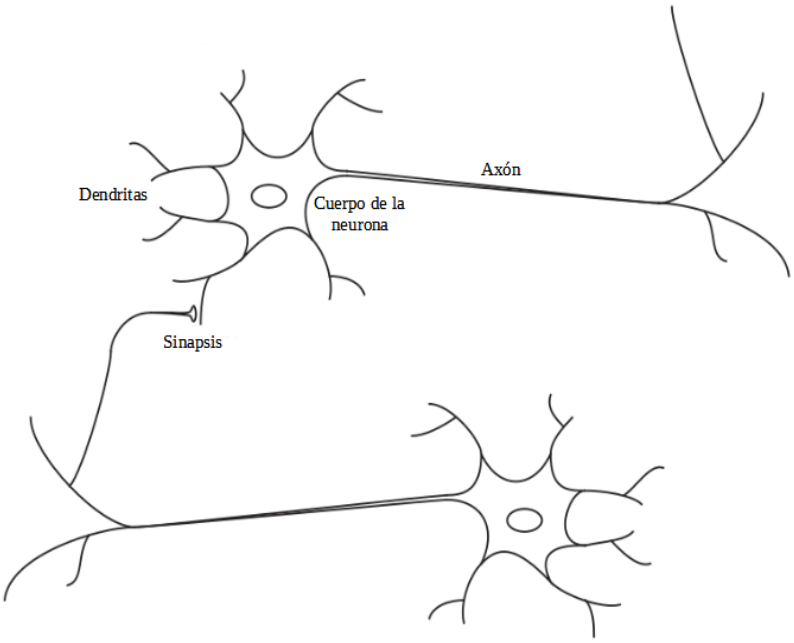


Figura 3.1: En el esquema se muestran dos neuronas biológicas que tienen una sinapsis (tomada de [Hagan et al., 2014]).

Las neuronas artificiales son abstracciones muy simples de lo que son las neuronas biológicas. En la Figura 3.2 se puede observar el esquema de una neurona artificial que cuenta con múltiples entradas. La neurona está compuesta por los pesos ($\boldsymbol{w}$) que ponderan a las entradas ($\boldsymbol{x}$), así como el bias (b) y la función de activación (f). La neurona multiplica las entradas $\boldsymbol{x}$ por los pesos $\boldsymbol{w}$ para posteriormente agregar el bias b. El resultado de esta suma ponderada es tomado como entrada para la función de activación que regresa y. Al hacer una analogía con la neurona biológica los pesos de la neurona artificial corresponden a la fuerza de las sinapsis. Mientras que la suma, el bias y la función de activación corresponden al cuerpo de una neurona biológica, por su parte la salida a corresponde a la señal que pasa por el axón.

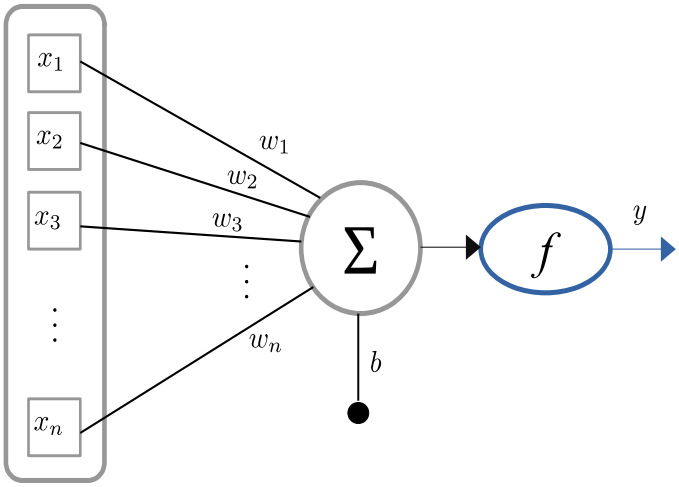


Figura 3.2: Esquema de una neurona artificial que cuenta con n entradas.

La neurona artificial es una abstracción matemática representada por medio de la Ecuación 3.1 [Hagan et al., 2014]. En ésta se puede observar como el vector de entrada $\boldsymbol{x}$ es multiplicado por los pesos $\boldsymbol{w}$, a la multiplicación se le suma el bias b y el resultado es la entrada de la función de activación. Esta abstracción matemática aprovecha los conceptos del álgebra lineal por lo cual todas las operaciones relacionadas con las redes neuronales son vectoriales.

$$y = f(\boldsymbol{w}^T \boldsymbol{x} + b) \tag{3.1}$$

Las redes neuronales son utilizadas para resolver distintos problemas entre los que se encuentra la clasificación. Cuando se utiliza una red neuronal en un problema de clasificación, se pretende que la red tenga la capacidad de identificar la clase a la que pertenece la entrada x de entre un grupo de clases. Sin embargo, cuando se utiliza solo una neurona la entrada x únicamente puede ser clasificar entre dos clases [Hagan et al., 2014]. En la Figura 3.3 se muestran puntos en un plano separados por un limite de decisión, este limite es trazado por una neurona artificial. Los puntos que se encuentren a un lado de la linea pertenecen a una clase, mientras que los que se encuentren al otro lado de la linea pertenecen a la segunda clase.

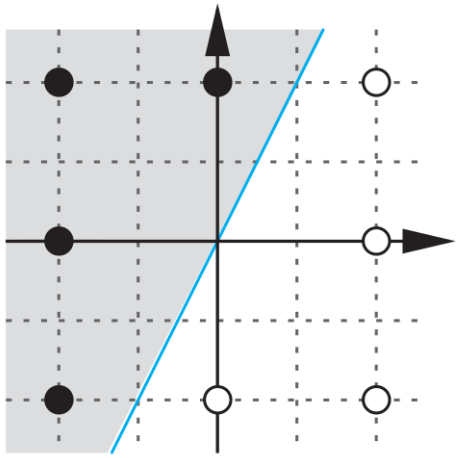


Figura 3.3: El plano es dividido por la linea azul, que es un limite de decisión creado por una neurona artificial. Por tanto, los puntos que se encuentren a la izquierda del limite de decisión se clasifican como negros, mientras que los puntos que se encuentren a la derecha se clasifican como blancos [Hagan et al., 2014]

Las capacidades de una neurona artificial por si sola son insuficientes para resolver problemas complejos, por lo que se suele necesitar de múltiples neuronas trabajando de manera conjunta [Raschka and Mirjalili, 2017]. No obstante, cuando se tienen múltiples neuronas trabajando en resolver un problema la forma en que se organizan es muy importante. En la Figura 3.4 se muestra un esquema que representa un tipo de organización de neuronas conocido como capa. En el caso de una capa las neuronas se organizan en paralelo, esto significa que todas las neuronas reciben las mismas entradas y a la salida de la capa se obtiene un vector $\mathbf{y}$ que tiene la salida de cada una de las neuronas.

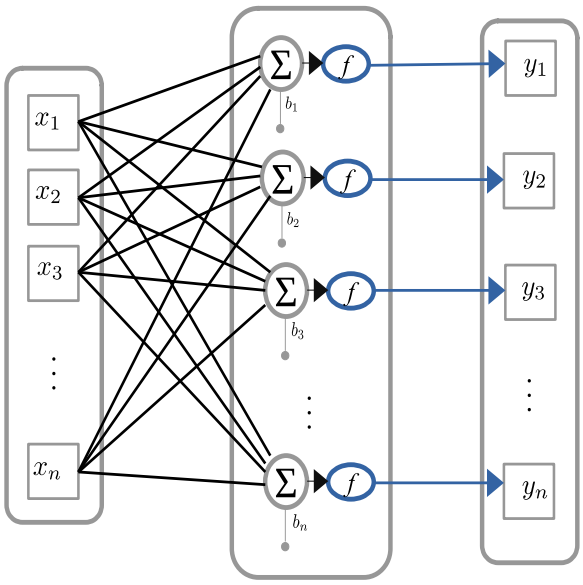


Figura 3.4: En el esquema se puede observar como las neuronas se encuentran organizadas de tal manera que reciben las mismas entradas.

En la Ecuación 3.2 se muestra la representación matemática de una capa de neuronas completamente conectadas [Hagan et al., 2014]. Esta ecuación es similar a la Ecuación 3.1, aunque en realidad son distintas debido a que $\boldsymbol{W}$ ya no es un vector sino una matriz que contiene los pesos de todas las neuronas de la capa. Además, la salida $\boldsymbol{y}$ es un vector que contiene la salida de la función de activación para todas las neuronas.

$$\boldsymbol{y} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{3.2}$$

3.2. Redes Neuronales Multicapa

Las redes neuronales multicapa cuentan con una complejidad superior a las redes de una sola capa. En este tipo de redes las capas se organizan de manera secuencial, en donde la salida de una capa es utilizada como la entrada de la capa siguiente. En la Figura 3.5, se muestra el esquema de una red del tipo alimentada hacia adelante también llamada Perceptrón Multicapa (MLP por sus siglas en inglés). Este tipo de red se caracteriza por ser una red completamente conectada, ya que cada neurona de la capa se encuentra completamente conectada a las entradas y la salida de la capa se encuentra completamente

conectada a la entrada de la capa siguiente [Raschka and Mirjalili, 2017]. En el esquema se tienen tres capas, la capa de entrada, la capa oculta y la capa de salida. La capa de entrada en realidad solo es conformada por las entradas, por lo que no hay neuronas en ésta. La capa oculta se encuentra entre la capa de salida y la capa de entrada, se encuentra oculta ya que no interactúa con el exterior, además las redes neuronales multicapa pueden tener más de una capa oculta. Por otro lado, la capa de salida está formada por las neuronas que entregan los resultados de la red.

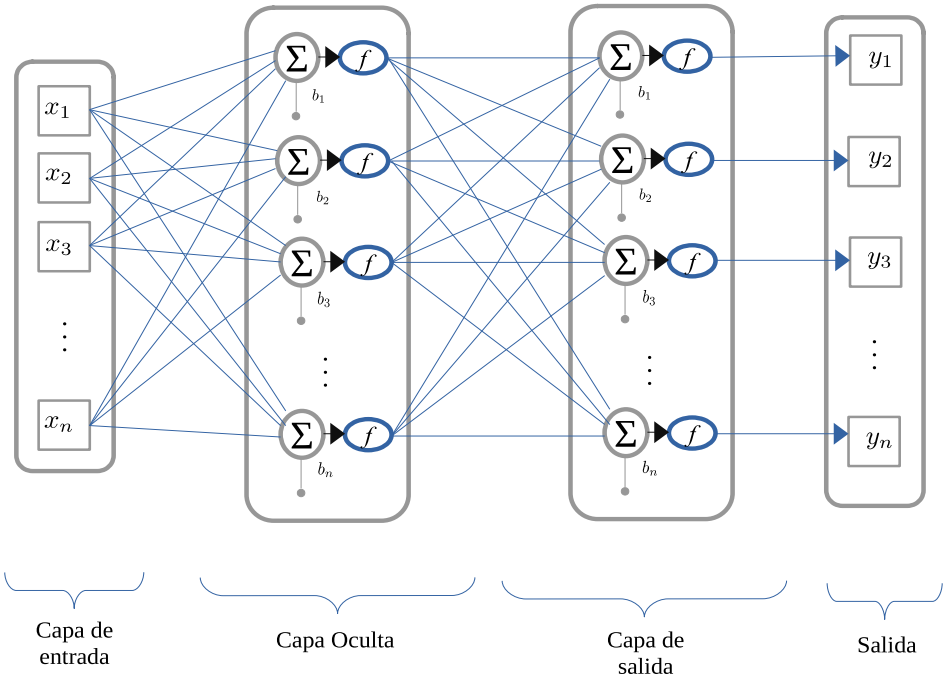


Figura 3.5: En una red neuronal multicapa se utiliza la salida de una capa como entrada de la capa siguiente. Esto permite que a mayor número de capas la red neuronal pueda generar conocimiento más abstracto.

Las redes neuronales con múltiples capas son muy importantes en el campo del aprendizaje profundo, ya que este tipo de redes tienen la capacidad de adquirir conocimiento jerarquizado. Esto significa que en las primeras capas de la red se tienen conocimientos muy simples, pero conforme se avanza dentro de la red, el conocimiento adquirido por las capas se torna más abstracto [Massiris et al., 2018]. Esto es importante, ya que gracias a la capacidad de adquirir conocimiento jerarquizado se pueden abordar problemas increíblemente complejos como el procesamiento de lenguaje natural, el reconocimiento de patrones

en imágenes, la síntesis de voz, el análisis genético y el pronostico de enfermedades.

Hasta el momento los esquemas que se han mostrado sobre capas y redes multicapa representan redes neuronales densas. Sin embargo, existen otros tipos de capas como las recurrentes y las convolucionales. Cada tipo de capa cuenta con sus propias características que les permite resaltar en tareas distintas. En el caso de las capas recurrentes la salida de la capa es utilizada como entrada, lo que permite tener un tipo de memoria. Este tipo de capas son comúnmente utilizados para resolver problemas en los que la temporalidad es importante, como en el campo del procesamiento de lenguaje natural o el reconocimiento y la síntesis de voz [Pérez-Ortiz, 2002]. Por otro lado, las capas convolucionales han tenido un mayor impacto en el campo de la visión por computadora. Este tipo de redes se caracteriza porque sus parámetros en realidad son filtros de convolución [Goodfellow et al., 2016].

3.3. Funciones de activación

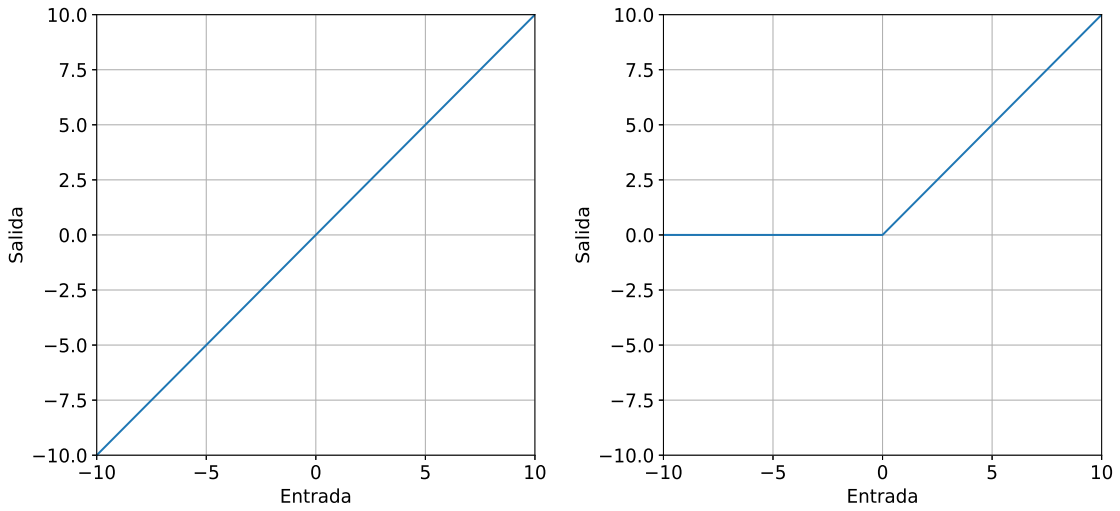
El cuerpo de la neurona biológica se encarga de realizar la suma de las entradas y la aplicación de un umbral. Bueno, en el caso de la neurona artificial la función de activación se encarga de realizar el trabajo de umbral, por lo que dependiendo de la función de activación utilizada la neurona artificial adquiere un comportamiento u otro. Las funciones de activación pueden ser lineales o no lineales, de manera que al utilizar una función de activación no lineal el comportamiento de la neurona es modificado y de igual manera se vuelve no lineal [Raschka and Mirjalili, 2017]. Por lo que dependiendo del problema a tratar será más conveniente utilizar una u otra función de activación. Existen varias funciones de activación, aunque en este trabajo solo explicamos las funciones *Lineal*, *ReLU* y $Softmax$, ya que las últimas dos son utilizadas en la implementación realizada en esta tesis.

La función *Lineal* es la más sencilla de todas ya que la salida de la función de activación es la misma que su entrada $a = n$. En la Figura 3.6 a) se puede observar la gráfica de la función *Lineal*. Como su nombre lo indica es una función lineal, por lo que al aplicarla en una neurona, el comportamiento de la neurona sigue siendo lineal.

La función de activación Unidad Lineal Rectificada (*Rectified Linear Unit*, ReLU, por sus siglas en inglés) es una función de activación que es comúnmente utilizada en

la implementación de redes neuronales profundas. Esto se debe a sus propiedades, ya que no tiene saturación superior, lo cual es útil durante el entrenamiento de redes profundas, además, es una función no lineal por lo que es de ayuda para la red neuronal al aprender funciones complejas [Raschka and Mirjalili, 2017]. La Ecuación 3.3 describe el comportamiento de la función de activación *ReLU*, esta función recibe un valor de entrada que compara con cero y retorna el mayor de los dos. Por tanto, cuando la función recibe como entrada valores negativos retorna cero, pero en el caso de recibir valores mayores a cero retorna el valor de entrada, esto se muestra en la Figura 3.6 b).

$$\phi(z) = \text{máx}(0, z) \tag{3.3}$$



(a) *Función de activación* *Lineal* **(b) *Función de activación* *ReLU***

Figura 3.6: En la figura a) se muestra la función lineal y como se puede notar la salida es igual a la entrada. Mientras que en la función b) la función *ReLU* la salida es cero cuando la entrada es negativa.

La función de activación $Softmax$ es comúnmente utilizada en tareas de clasificación multiclase, donde se tienen C clases y se requiere identificar a cual pertenece la entrada de la red. Por lo que se busca tener a la salida de la red un vector del tipo one-hot, esto significa que el vector indica solo una clase positiva y $C - 1$ clases negativas [Lawrence et al., 1997].

El comportamiento de las neuronas con función de activación $Softmax$ se rige por la Ecuación 3.4, donde s_i es la puntuación que le da la red a la clase i [Lawrence et al., 1997]. Una característica muy importante de esta función de activación es que la salida de una neurona depende de la salida de las demás neuronas, lo que en la ecuación es representado con un sumatorio que es utilizado para normalizar la salida de la neurona. Por tanto, la salida de cada neurona representa la probabilidad de pertenecer a la clase asociada a la neurona y la suma de la salida de todas las neuronas debe ser 1.

$$f(s_i) = \frac{e^{s_i}}{\sum_{j=0}^{C} e^{s_j}} \tag{3.4}$$

3.4. Redes Neuronales Convolucionales

Las redes neuronales convolucionales son una familia de modelos que se desarrollaron inspirándose en la forma que trabaja el cerebro humano al momento de reconocer objetos [Raschka and Mirjalili, 2017]. El nombre de este tipo de redes se debe a que basan su funcionamiento en la aplicación de operaciones de convolución, ya sea en una dimensión o en dos. Este trabajo se centra en el uso de redes convolucionales de dos dimensiones, ya que es necesario extraer características de imágenes. Las imágenes están compuestas por una cuadricula de píxeles que comúnmente se encuentran relacionados de manera local. Las CNN aprovechan estas relaciones locales al aplicar filtros de convolución para extraer características, lo que les ha permitido destacar en el área de visión por computadora.

La convolución es una operación aplicada a dos funciones [Goodfellow et al., 2016], esta operación comúnmente se denota por un asterisco como se muestra en la Ecuación 3.5. En esta ecuación I representa la imagen a la que se le ha de aplicar la operación convolución y K representa el filtro o kernel de convolución. Mientras que las variables j e i representan la coordenada de la imagen en la que se aplica el filtro de convolución. En una capa convolucional el filtro se aplica a toda la imagen, por lo que la operación de convolución se aplica reiteradas veces desplazando el filtro al modificar las coordenadas j e i. El desplazamiento se suele hacer moviendo el filtro por todas las columnas renglón por renglón, aunque esto puede ser modificado dando pasos de tamaño distinto. Por otro lado, el resultado de aplicar

el filtro de convolución a toda la imagen se le llama mapa de características, ya que relaciona las características extraídas con la posición de la imagen en que se encuentran.

$$s(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{3.5}$$

En la Figura 3.7 se muestra una imagen y como a partir de aplicar un filtro de convolución se obtiene un valor en el mapa de características. Además, al mismo tiempo se muestra como cada píxel en el mapa de características está relacionado con varios píxeles de la imagen original.

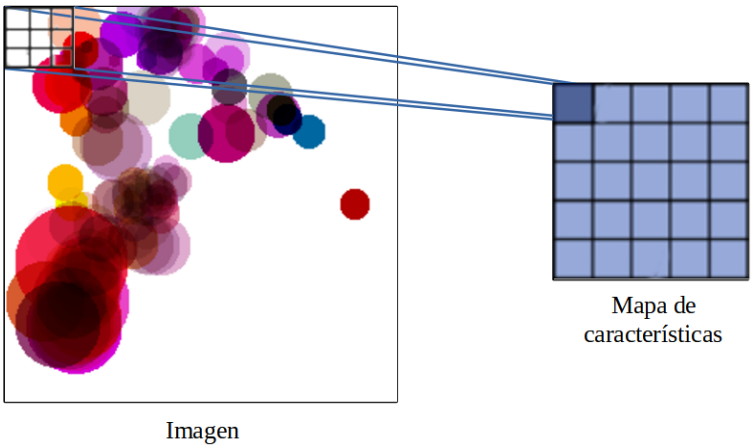


Figura 3.7: Esquema que representa como un píxel en el mapa de características está relacionado con varios píxeles de la imagen.

Los filtros de convolución o kernel para las capas convolucionales en dos dimensiones son en realidad pequeñas matrices. En la Figura 3.8 se muestra un kernel de convolución de tamaño 3×3, el cual comúnmente es utilizado en el área de visión computacional para encontrar bordes.

-1	0	1
-2	0	2
-1	0	1

Figura 3.8: Filtro o kernel de convolución en 2D.

Los filtros de convolución pueden ser aplicados a las imágenes utilizando un relleno de ceros al que se le suele llamar padding. Existen tres tipos de padding, el Full padding, el same padding y el valid padding [Raschka and Mirjalili, 2017]. El full padding es raramente utilizado, ya que el mapa de características resultante es más grande que la imagen original. Por su parte, el same padding nos permite tener mapas de características del mismo tamaño de la imagen original. Mientras que el valid padding se aplica al interior de la imagen y el resultado es una mapa de características de menor tamaño. En la Figura 3.9 se muestran estos tres tipos de padding utilizando cuadros. Los cuadros azules representa a la imagen, los cuadros en blanco al padding, los cuadros sombreados el kernel y los cuadros en verde el mapa de características resultante.

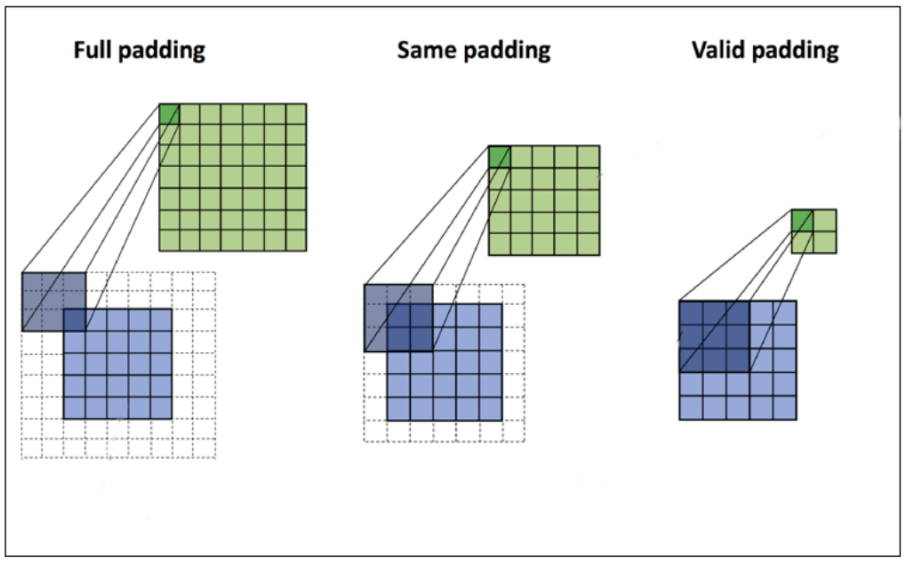


Figura 3.9: Dependiendo del tipo de padding que se utilice puede obtenerse como resultado un mapa de características de mayor o menor tamaño (tomada de [Raschka and Mirjalili, 2017]).

Las CNN destacan por la capacidad de obtener conocimiento jerarquizado y la economía de recursos requeridos [LeCun et al., 1995]. Este tipo de redes son compuestas por una serie de capas convolucionales, las primeras capas se encargan de la extracción de características sencillas como los bordes en las imágenes. Sin embargo, las capas siguientes trabajan sobre los mapas de características, lo que permite detectar rasgos más abstractos y complejos. Por otra parte, cada capa de convolución está compuesta por una serie de filtros,

por lo que cada capa convolucional únicamente contiene los parámetros de los filtros y el bias, lo que se torna en un menor uso de recursos. Por ejemplo, sí a la entrada de una red densa se da una imagen de tamaño 256×256 se necesitan 65536 pesos por cada neurona, un peso por cada píxel, además del bias de cada neurona. Mientras que una capa convolucional que contenga 64 filtros de tamaño 5x5 necesita almacenar y entrenar únicamente 1664 parámetros, de los cuales 1600 son los coeficientes de los filtros y 64 son los bias.

En la Figura 3.10 se muestra una estructura de una red neuronal convolucional. Una red neuronal convolucional está comúnmente compuesta por capas de convolucionales que suelen ser seguidas por capas de pooling también conocidas como capas de reducción. Además, después de las capas convolucionales se realiza un aplanado que consiste en convertir la imagen resultado de las capas de convolución en un vector. Esto es necesario para que las características extraídas por las capas convolucionales sean utilizadas por una serie de capas densas que se encargan de llevar a cabo la tarea de clasificación.

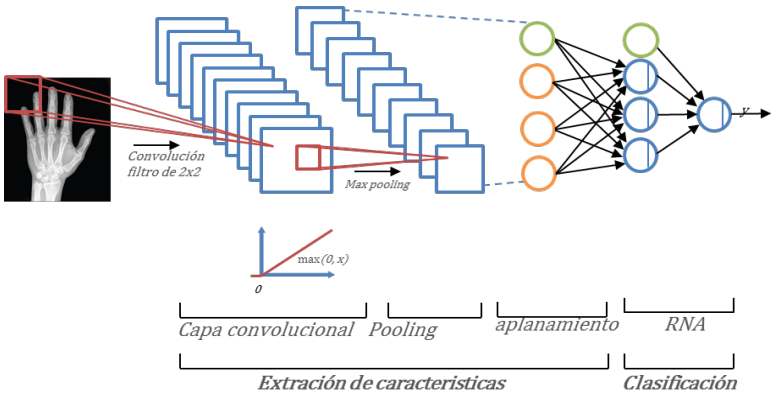


Figura 3.10: La extracción de características es realizada por capas convolucionales en conjunto con capas de pooling. La imagen resultante es aplanada convirtiendose en un vector que es usado por capas densas para la clasificación (tomada de [Arias et al., 2019]).

La operación de pooling en realidad es un submuestreo que ayuda a obtener la información más relevante de los mapas de características, además de aportar robustez ante el ruido [Albawi et al., 2017] [Raschka and Mirjalili, 2017]. En CNN existen dos formas típicas de realizar el submuestreo, el *maxpooling* y el *meanpooling* [Raschka and Mirjalili, 2017], aunque en este trabajo solo se utiliza el *maxpooling*. El *maxpooling* consiste en tomar el

valor máximo en un vecindario de píxeles, al tamaño del vecindario se le suele llamar pooling size. En la Figura 3.11 se muestra como se tiene una pequeña imagen de tamaño 4×4, a la que se le aplica *maxpooling* con un vecindario de tamaño 2×2, al finalizar se obtiene una imagen de tamaño 2×2 con los máximos por vecindario.

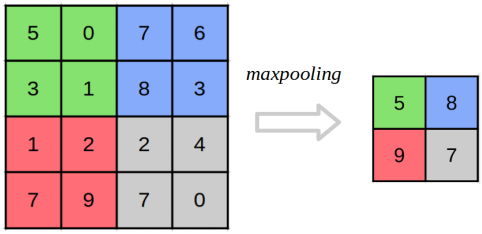


Figura 3.11: Al aplicar un *maxpooling* con vecindario de tamaño 2×2 a la imagen de la izquierda se toman los valores más grandes y como resultado se obtiene una imagen más pequeña.

3.5. Aprendizaje de una red neuronal

La propiedad más importante de las redes neuronales es su capacidad de aprender a través de los datos, para lo cual se puede optar por distintas estrategias, aunque son tres los paradigmas de aprendizaje más comunes, el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo [Goodfellow et al., 2016]. En el aprendizaje supervisado la red neuronal aprende al mostrarle datos que fueron etiquetados con anterioridad, por lo que la red neuronal es guiada durante el proceso de aprendizaje, ya que compara su salida con la etiqueta del dato de entrada. En el aprendizaje no supervisado se tiene un conjunto de datos no etiquetados, por lo que se debe de encontrar patrones en la estructura de los datos. Por otro lado, en el aprendizaje por refuerzo se aprende al interactuar con el ambiente y recibir recompensas o penalizaciones por las elecciones que han sido tomadas.

Cuando una red neuronal se encuentra en el proceso de aprendizaje, los parámetros internos de la red son modificados para cambiar su comportamiento, lo que provoca que la red aprenda. En el caso del aprendizaje supervisado, el proceso para modificar los parámetros de la red consiste en comparar la respuesta de la red a los datos de entrada con sus respectivas etiquetas [Raschka and Mirjalili, 2017]. La comparación permite tener una medida de error que indica cuan alejada se encuentra la salida de la red de la etiqueta.

Para obtener el error entre la salida de la red y la etiqueta se utiliza una función a la que comúnmente se le llama función de pérdida o función de costo. Dependiendo del problema a tratar se ha de utilizar una función de costo u otra, en el caso de este trabajo se utiliza la función de costo *cross_entropy*. Esta función suele ser adecuada en modelos de redes cuya salida representa una probabilidad, como cuando se hace una clasificación categórica con función de activación $Softmax$ [Ignacio G.R. Gavilán, 2021].

El comportamiento de la función de costo *cross_entropy* es representado por la Ecuación 3.6. Como se menciono al utilizar la función $Softmax$ la red regresa un vector de tamaño C con la probabilidad de que la muestra pertenezca a cada clase. En la ecuación, t_i representa el valor objetivo de la clase i, mientras que f_c es la función de activación utilizada, que en este caso es la función $Softmax$. Por otro lado, s_i es el puntaje que le otorga la red a la clase i.

$$cross\_entropy = -\sum_{i=1}^{C} t_i \log(f_c(s_i)) \tag{3.6}$$

Al sustituir f_c por la ecuación de la función $Softmax$ se obtiene la Ecuación 3.7. En la ecuación s_p representa el puntaje de la clase positiva, esto se debe a que de todos los t_i solo el correspondiente a la clase positiva tiene valor de 1 y los demás tienen valor de 0.

$$cross\_entropy = -\log\left(\frac{e^{s_p}}{\sum_{j=0}^{C} e^{s_j}}\right) \tag{3.7}$$

Durante el entrenamiento la función de costo nos dice que tan alejado está el resultado de la red comparado con el resultado objetivo. No obstante, aún es necesario modificar los parámetros de la red, lo que se lleva a cabo con la ayuda de un optimizador y el algoritmo Back Propagation. El optimizador se encarga de minimizar la función de costo para que de esta manera se minimice la diferencia entre los resultados de la red y las etiquetas [Raschka and Mirjalili, 2017]. Existen varios optimizadores, sin embargo, la idea básica detrás de ellos es encontrar un mínimo de la función de costo, ya sea local o global, para lo que se utiliza el gradiente de la función de costo. El gradiente contiene las

derivadas parciales de la función de costo con respecto a los parámetros de la red, esto le permite al optimizador saber como afecta cada parámetro de la red al error de la función de costo. Por lo que el optimizador solo necesita dar pasos en la dirección opuesta al gradiente, ya que el gradiente apunta hacia el máximo de la función. Este proceso es conocido como descenso del gradiente y se puede observar una representación en la Figura 3.12, donde la flecha indica la trayectoria que se sigue para llegar a un mínimo. Al tamaño del paso que da el optimizador se le suele llamar tasa de aprendizaje o leaning rate. En este trabajo se utiliza el optimizador Adam para ajustar los parámetros de la red, este optimizador toma en cuenta los valores pasados del gradiente, lo que genera un suavizado en la trayectoria de descenso. Además, la tasa de aprendizaje se va adaptando al tomar en cuanta el historial de cambios del gradiente durante el proceso de aprendizaje [Kingma and Ba, 2014].

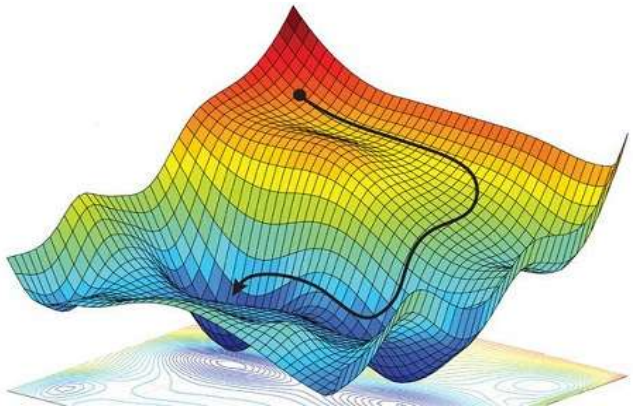


Figura 3.12: La linea negra muestra la trayectoria seguida por el optimizador durante el descenso del gradiente (tomada de [Ignacio G.R. Gavilán, 2021]).

Por otro lado, las redes neuronales pueden ser vistas como composiciones de funciones, por lo que es necesario llevar a cabo un proceso para poder obtener el gradiente que ayude al optimizador a saber la dirección de descenso. Aquí es donde entra el algoritmo back propagation, ya que su trabajo es obtener el gradiente que corresponde a la topología de la red neuronal y por tanto, ayudar a propagar el error a través de las capas de la red neuronal [Goodfellow et al., 2016].

3.6. Sobre-entrenamiento

El problema del sobre-entrenamiento es muy común en el proceso de aprendizaje de las redes neuronales. Este problema se presenta cuando una red neuronal realiza un buen trabajo con los datos utilizados en el entrenamiento. Sin embargo, no lleva a cabo un buen trabajo con los datos desconocidos, como el conjunto de prueba, lo cual implica que la red no generaliza bien [Goodfellow et al., 2016]. Este problema se presenta comúnmente por el hecho de que la red neuronal cuenta con demasiados parámetros, por lo que tiene mucha flexibilidad y aprende el ruido de los datos en lugar de la dinámica del sistema [Raschka and Mirjalili, 2017]. En contraste, cuando un modelo tiene muy pocos parámetros y no puede aprender los patrones de los datos del conjunto de entrenamiento se presenta un problema que es conocido como sub-entrenamiento. En la Figura 3.13 se muestran los tres representaciones de los casos antes descritos.

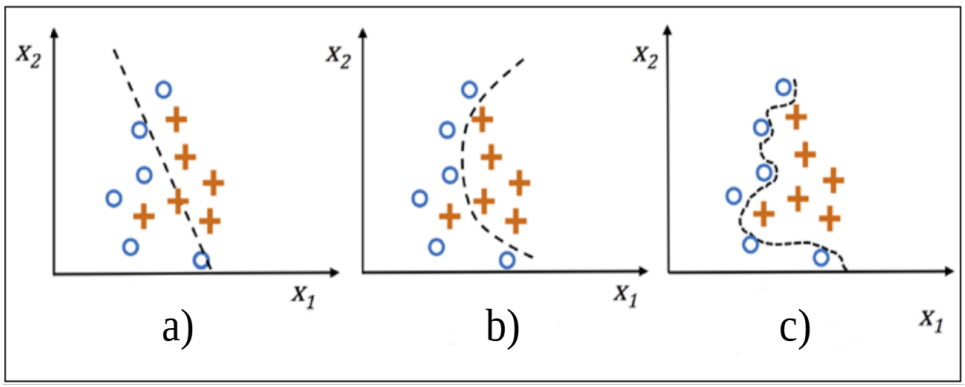


Figura 3.13: En la figura a) se puede notar que la red presenta sub-entrenamiento, ya que no es capas de separar correctamente círculos y cruces. En al figura b) la red generaliza bien. Mientras que en la figura c) la red presenta sobre-entrenamiento, puesto que genera una frontera muy compleja (tomada de [Raschka and Mirjalili, 2017]).

El sobre-entrenamiento en dos o tres dimensiones se puede identificar simplemente observando una gráfica. No obstante, para poder identificar el sobre-entrenamiento en redes comunes que trabajan con cientos o miles de dimensiones es necesario optar por otras estrategias. La estrategia más común consiste en tomar el conjunto de datos de entrenamiento y dividirlo en dos, un conjunto de entrenamiento que normalmente se queda con el 80 % de los datos y un conjunto de validación que toma el 20 % restante. Por tanto, durante el

entrenamiento se revisa constantemente cual es el error obtenido con el conjunto de validación. En el momento que el error en el conjunto de validación comienza a incrementar pero el error en el conjunto de entrenamiento sigue disminuyendo, se considera que comienza a presentarse el sobre-entrenamiento.

Para resolver el problema del sobre-entrenamiento existen varias alternativas como los métodos de regularización L1 y L2, o la técnica Dropout [Goodfellow et al., 2016]. Aunque, en este proyecto se utiliza un método más sencillo llamado early-stopping, que consiste en monitorizar el error de validación con la finalidad de identificar el momento en que comienza a incrementar, lo que indica la presencia del sobre-entrenamiento. Cuando se identifica el sobre-entrenamiento el entrenamiento es detenido y se retornan los parámetros de la red con los que se obtuvo los mejores resultados. Sin embargo, durante los entrenamientos es probable que el error de validación incremente por instantes para volver a disminuir, por lo que se utiliza el concepto de paciencia. La paciencia consiste en esperar un plazo determinado a que el error disminuya, si el plazo es superado el entrenamiento es detenido. En la Figura 3.14 se muestra una gráfica con el error de validación y el error de entrenamiento, en ésta se puede observar como fluctúa el error de validación, así como el momento en el que se comienzan a separar el error de validación y el error de entrenamiento.

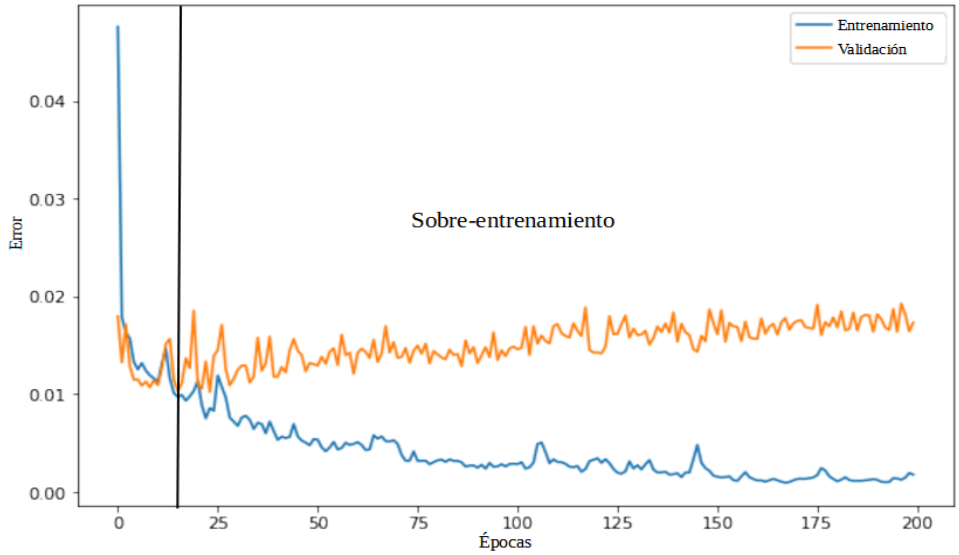


Figura 3.14: En la gráfica se puede notar como después de la linea negra el error de validación comienza a incrementar y el error de entrenamiento sigue disminuyendo.

3.7. Conclusiones del capítulo

Las redes neuronales han tomado una gran relevancia en los últimos años, lo cual se debe a la mejora en los algoritmos, el desarrollo de hardware más avanzado para el entrenamiento de las redes neuronales, así como el incremento en la cantidad y la calidad de información disponible para entrenar a las redes neuronales. Sin embargo, utilizar una red neuronal en la resolución de algún problema involucra un proceso iterativo en múltiples etapas, que es supervisado por el usuario para ajustar el número de capas y la cantidad de parámetros en cada una de ellas, ya que no hay una fórmula para decidir cuantas capas utilizar o cuantos parámetros por capas.

Capítulo 4

Implementación

En esta tesis se propone un método de identificación de parlantes independiente del texto. La propuesta consiste en utilizar los formantes de la voz para crear imágenes que representen a los parlantes, esto permite utilizar una red neuronal convolucional en el proceso de identificación. En este capítulo se describe el proceso que se lleva a cabo para extraer los formantes de los audios, el proceso para crear las imágenes a partir de de los formantes, así como la descripción de la red neuronal empleada. Las imágenes que representan a los parlantes son imágenes de fondo blanco con círculos cuyos parámetros, el radio y la posición del centro, están relacionados con los formantes. Para crear las imágenes se propone el diseño de dos algoritmos que deben trabajar juntos para la creación de las imágenes. El primer algoritmo se encarga de convertir los formantes de la voz en parámetros de un círculo. Mientras que el segundo algoritmo recibe los parámetros y se encarga dibujar los círculos en la imagen de fondo blanco. Además, se propone una modificación al algoritmo encargado de la creación de imágenes para aumentar la cantidad de imágenes clasificadas correctamente.

4.1. Estimación de los formantes

En esta sección se explica el proceso que se lleva a cabo en este trabajo para obtener los formantes de la voz. En la Figura 4.1 se muestra un diagrama de bloques del procedimiento que se sigue. Nuestro sistema comienza por leer el archivo de audio, la señal que se obtiene de este archivo es preprocesada y posteriormente se utiliza para obtener fragmentos de 30ms. Estos fragmentos llamados marcos son analizados para identificar cuales contienen sonido vocalizado, esto se debe a que los marcos que contienen sonido vocalizado se utilizan para estimar los formantes de la voz. Por tanto, una vez que se han identificado los marcos con sonido vocalizado se procede a estimar los primeros cuatro formantes. Al finalizar se obtiene como resultado una lista que contiene los primeros cuatro formantes estimados a partir de cada marco con sonido vocalizado.

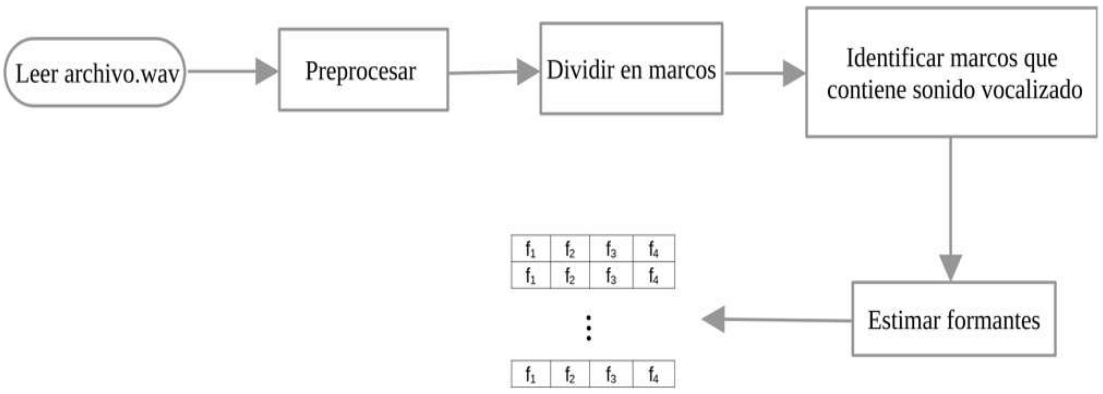


Figura 4.1: Diagrama general de estimación de formantes.

4.1.1. Preprocesamiento

El preprocesamiento es la primera etapa, en ésta se realiza la búsqueda del inicio y el final del audio, así como la aplicación del filtro de pre-énfasis. La Figura 4.2 muestra un diagrama de bloques donde se puede observar la señal de audio antes y después del preprocesamiento. En nuestro sistema se recurre a la Energía de tiempo corto y al régimen de cruces por cero para identificar el inicio y el fin de la señal. Posteriormente se procede a la aplicación del filtrado de pre-énfasis, por lo que se obtiene como resultado una señal que únicamente contiene la elocución, descartando las secciones sin audio.

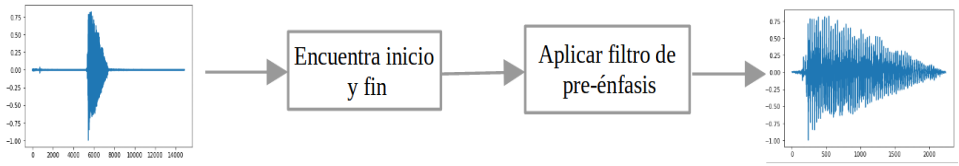


Figura 4.2: Diagrama del preprocesamiento.

4.1.2. División del audio en marcos

La división del audio en marcos consiste en tomar pequeños fragmentos del audio, esto puede ser observado en la Figura 4.3, donde se tiene un fragmento de 60ms de audio y se obtiene a partir de este cuatro marcos. El procedimiento se lleva acabo utilizando una ventana de 30ms que se desplaza 10ms, esto permite que el marco actual comparta información con el marco anterior y con el marco posterior. La ventana es desplazada por todo el audio obteniendo como resultado una lista de marcos. En este trabajo se decidió utilizar la ventana de Hamming en lugar de la ventana rectangular.

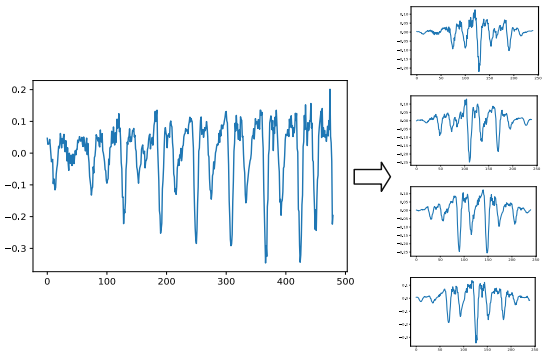


Figura 4.3: La señal de audio consta de 480 muestras lo que equivale a 60ms y se obtienen cuatro marcos de 240 muestras que equivalen a 30ms.

4.1.3. Identificación de marcos con sonido vocalizado

Cada uno de los marcos obtenidos pasa por un proceso para identificar aquellos que contienen sonido vocalizado. Este proceso es representado en la Figura 4.4 por medio de un diagrama de bloques. El proceso comienza con la aplicación de un filtro pasa-bajas

al marco, este filtro tiene una frecuencia de corte igual a 900Hz. Después del filtrado se aplica un aplanador de espectro, esto con la finalidad de identificar más fácil los picos importantes. En el caso de este trabajo se utiliza el Center Clipping. Posteriormente se obtiene la autocorrelación del marco que ha pasado por el aplanador de espectro. En el resultado de la autocorrelación se procede a buscar el pico más grande, ya que su posición indica el tono en caso de contener sonido vocalizado. Sin embargo, es necesario que el pico tenga un tamaño mínimo, por lo que el pico debe pasar el umbral establecido para que el marco sea considerado como contenedor de sonido vocalizado. En el caso contrario el marco es descartado.

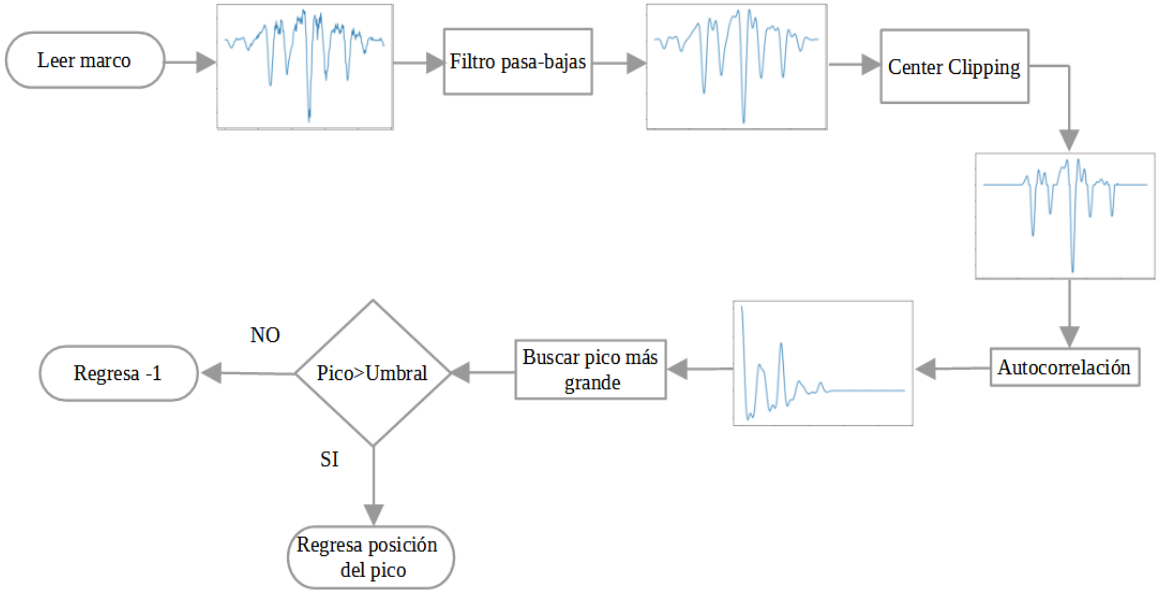


Figura 4.4: Diagrama del proceso para identificar marcos con sonido vocalizado.

4.1.4. Obtener los formantes de la voz

Los marcos que contienen sonido vocalizado son utilizados para estimar los formantes de la voz. El proceso que lleva a cabo nuestro sistema se muestra en el diagrama de bloques en la Figura 4.5. El sistema comienza por obtener los coeficientes LPC del marco, ya que estos coeficientes representan a los parámetros del filtro de predicción lineal. El cálculo de los coeficientes LPC se realiza con la ayuda del algoritmo de Levinson-Durbin. En los

siguientes dos pasos del proceso se crea el polinomio conformado por los coeficientes LPC y se obtienen sus raíces. No todas las raíces del polinomio representan a los formantes, por lo que es necesario identificar las raíces que sí los representan. Este proceso requiere dos pasos en el primero de ellos se calcula el ancho de banda y frecuencia de las raíces. En el segundo paso se calcula la relación entre el ancho de banda y la frecuencia de las raíces, cuando las raíces tienen una razón entre su ancho de banda y frecuencia inferior al umbral entonces son consideradas como formantes.

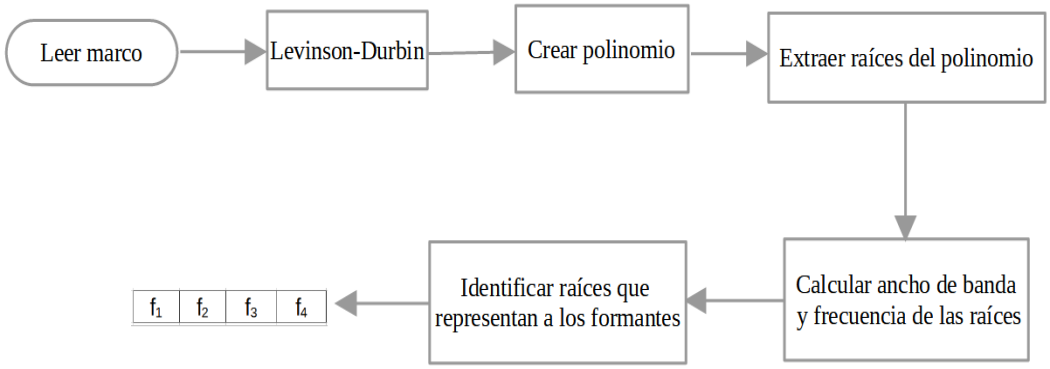


Figura 4.5: Diagrama del proceso de estimación de formantes.

4.2. Base de datos utilizada

En los experimentos se utilizó una base de datos que consta de 2856 archivos de audio, los cuales son archivos del tipo Wave que fueron muestreados a 8000Hz. Los audios pertenecen a 21 personas cuya lengua materna es el español. El grupo de personas está conformado por seis mujeres y quince hombres. Cada persona pronuncia 34 palabras, los números del cero al nueve y las letras del alfabeto griego. Además cada una de las palabras es pronunciada cuatro veces. Esta base de datos fue creada por el M.C. José Francisco Rico Andrade [Andrade and Ibarrola,] y puede ser encontrada en el link: `http://dep.fie.umich.mx/~camarena/dsp/elocuciones21.tar.gz`.

La base de datos está compuesta por audios cortos que en muchos de los casos contienen señales de audio con poca variedad de sonidos vocalizados. Por tanto, la información que aportan del parlante es muy escasa y queda descartada la opción de crear una

imagen a partir de un solo audio. Debido a esto se opto por utilizar los formantes de marcos que provienen de audios distintos para generar una imagen. La propuesta consiste en tener almacenados los formantes de cada marco obtenido de los audios de la base de datos en una lista. Posteriormente para crear una imagen se toman de forma aleatoria los formantes de un conjunto de N marcos. En la Figura 4.6 se puede ver la estructura de la lista donde se almacenan los formantes de cada uno de los marcos de los audio. Esta lista contiene una lista de formantes por cada audio, estas listas están en orden por parlante, por lo que a cada parlante se le asigna un número, además cada parlante pronuncia 34 palabras distintas y cada palabra tiene cuatro elocuciones, por lo que la lista tiene 2856 listas de marcos.

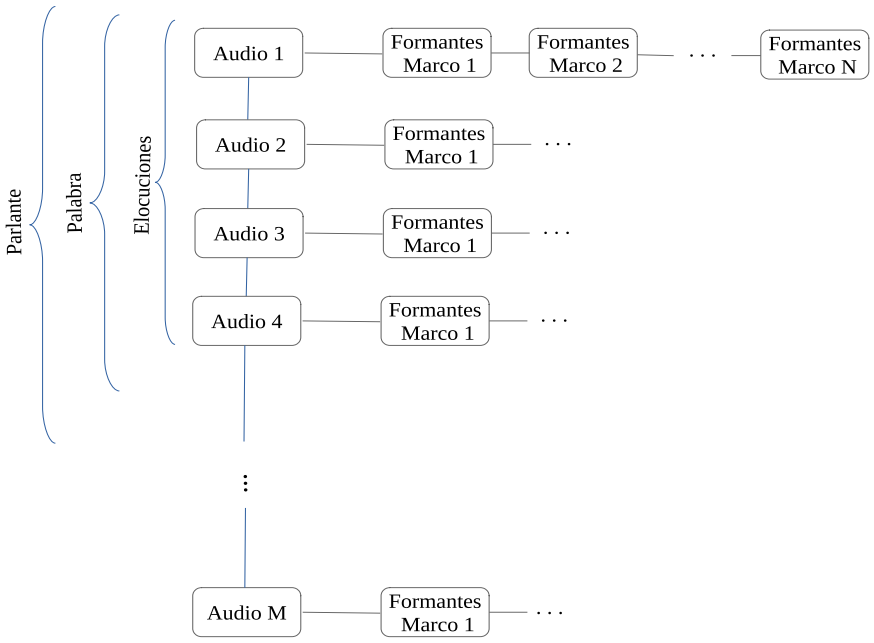


Figura 4.6: La lista que contiene los formantes de los marcos está organizada para poder acceder a los formantes de cada parlante sabiendo el número asignado al parlante, la palabra empleada y la elocución.

Las imágenes se generan tomando los formantes de los marcos de manera aleatoria, por lo que cada que se toma un marco se generan tres números aleatorios. Los números dan la posición de los formantes de un marco en la lista. El primero de los tres números aleatorios es tomado en un rango de 0 a 33, este número indica la palabra de la que ha de ser tomado el marco. El segundo número aleatorio indica cual de las elocuciones es utilizada. Una vez

que se tienen estos dos números ya se tiene el índice del audio a utilizar. El tercer número aleatorio se genera en el rango del número de marcos con sonido vocalizado que fueron obtenidos del audio y su tarea es elegir cual de todos los marcos se ha de utilizar.

4.2.1. Conjunto de entrenamiento y conjunto de prueba

Las imágenes generadas son utilizadas para entrenar una red neuronal, por lo que es necesario generar un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento es generado para que la red lo utilice al aprender los patrones de las imágenes. Mientras que el conjunto de prueba es utilizado para observar el comportamiento de la red neuronal ante imágenes distintas a las utilizadas para entrenar. En este proyecto se ha decidió utilizar la primera y la tercera elocución de cada palabra generar el conjunto de entrenamiento, así como utilizar la segunda y cuarta elocución de cada palabra para generar el conjunto de prueba. De esta manera las imágenes del conjunto de entrenamiento y las imágenes del conjunto de prueba son generadas a partir de archivos de audio distintos.

4.3. Generación de imágenes a partir de los formantes

Las imágenes son archivos digitales que están conformados por matrices cuyo contenido es el valor que toma cada uno de los píxeles en la pantalla. Esto se puede observar en la Figura 4.7 donde se tiene una imagen y su representación en formato RGB. Existen varios tipos de imágenes, así como varios formatos en los que se pueden almacenar. Los tres tipos de imágenes más comunes son las binarias, en escala de grises y las imágenes RGB. En las imágenes binarias los valores que toma cada píxel solo puede ser 0 o 1, por lo que la imagen resultante solo contiene píxeles en blanco o en negro. Por otro lado, en el caso de las imágenes en escala de grises cada píxel puede tomar valores discretos entre 0 y 255. No obstante, en este trabajo se utilizan imágenes del tipo RGB, este tipo de imágenes está constituido por tres matrices. Cada matriz almacena la información de uno de los tres colores, la matriz del color rojo, la matriz del color verde y la matriz del color azul. La combinación de los valores de las tres matrices genera los colores que los píxeles muestran en pantalla.

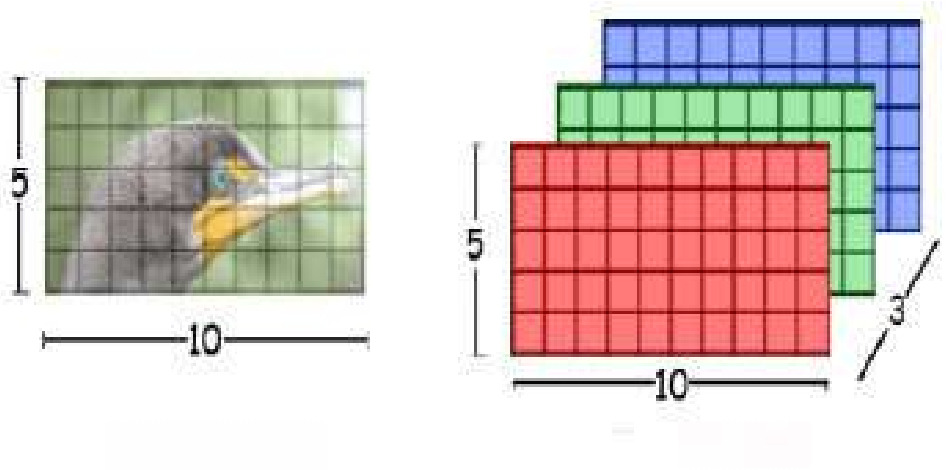


Figura 4.7: Imagen descompuesta en el formato RGB, (tomada de [Al-Azzeh et al., 2020]).

Las imágenes que se generan a partir de los audios tienen como finalidad ser representaciones de los parlantes. Debido a esto se utilizan los formantes, ya que estos aportan información tanto de las cuerdas vocales como de las propiedades acústicas del tracto vocal. La propuesta realizada en este proyecto consiste en tomar una imagen blanca como fondo y sobre ésta dibujar círculos cuyos parámetros están relacionados con los formantes. El proceso que se lleva a cabo para dibujar los círculos en la imagen requiere que primero se realice una conversión, donde a partir de los formantes de un marco se obtienen los parámetros de un círculo. En esta tesis se proponen dos formas de hacerlo, el Modelo I y el Modelo II. Posteriormente se utilizan los parámetros obtenidos de los formantes para dibujar el círculo en la imagen. El proceso se repite para un número determinado de círculos.

4.3.1. Generador de imágenes

Las imágenes son generadas utilizando el Algoritmo 2, este algoritmo se encarga de crear una imagen con la cantidad de círculos requerida. Los parámetros de cada círculo que es dibujado se calculan usando los formantes de uno de los marcos de la lista de listas mencionada en la sección 4.2. El algoritmo recibe un conjunto de parámetros: el número asignado al parlante, el número de círculos requeridos, la lista de listas, así como el ancho y el alto de la imagen. El algoritmo comienza por crear una imagen blanca utilizando los parámetros ancho y alto. Posteriormente en la linea 2 se comienza un ciclo que tiene como finalidad dibujar los círculos en la imagen. Dentro del ciclo se procede en la linea 3 a calcular un índice de la lista. Este índice es calculado de forma aleatoria usando el número del parlante, e indica de cual de los audios que pertenecen al parlante tomar los formantes que se han de utilizar para dibujar el círculo en la imagen. En la linea 4 se utiliza el índice del audio para calcular el número de marcos con sonido vocalizado que se han obtenido de este. El número de marcos se utiliza en la linea 5 para calcular aleatoriamente el índice del marco que contiene los formantes. De esta manera en la linea 6 utilizando los índices se toma un marco que es una estructura que guarda la frecuencia central y el ancho de banda de los formantes. La función $FormantesACirculo$ utiliza la estructura marco para calcular las coordenadas x y y del centro del círculo, así como el radio y el color. Los parámetros del círculo son utilizados en la linea 8 por la función *dibujaCirculo* para dibujar el círculo en la imagen. Este proceso se repite para el número de círculos que compondrán la imagen.

Algoritmo 2 Generador de imágenes

Entrada: *parlante*, *numCírculos*, *lista*, *ancho*, *alto*
Salida: *imagen*

1: $imagen \leftarrow nueva(ancho, alto, 3)$
2: **para** $i \leftarrow 0, numCírculos$ **hacer**
3: $índiceAudio \leftarrow indAleatorio(parlante)$
4: $N \leftarrow longitud(lista[índiceAudio]) - 1$
5: $indiceMarco \leftarrow aleatorio(N)$
6: $marco \leftarrow lista[índice][indiceMarco]$
7: $x, y, r, color \leftarrow FormantesACirculo(marco, ancho, alto)$
8: $dibujaCirculo(imagen, x, y, r, color)$
9: **fin para**
10: **devolver** *imagen*

4.3.2. Modelo I

El Modelo I utiliza los primeros tres formantes de la voz que se obtienen de un marco para dibujar un círculo en la imagen. Cuando se crean imágenes utilizando el Modelo I la función *FormantesACirculo* del Algoritmo 2 lleva a cabo el proceso que se muestra en el Algoritmo 3. Este algoritmo recibe tres parámetros de entrada: la estructura marco que contiene los formantes, el ancho de la imagen y el alto de la imagen. Al terminar el proceso el algoritmo regresa los parámetros de un círculo, sus coordenadas x y y de su centro, el radio y el color del círculo que está compuesto por el valor en formato RGB.

El Algoritmo 3 se caracteriza por realizar interpolaciones lineales, por lo que se necesitan los rangos en los que se encuentran las frecuencias de los formantes y sus anchos de banda, así como los rangos en los que se encuentran los parámetros del círculo. En la linea 2 se realiza la conversión de la frecuencia del primer formante a la coordenada x del centro del círculo. Esta conversión se realiza considerando que la frecuencia del primer formante se encuentra entre 100 Hz y 1000 Hz. Por su parte la coordenada x del centro del círculo se encuentra entre 0 y el ancho de la imagen. En la linea 3 se realiza el mismo proceso para obtener la coordenada en y, aunque en este caso se utiliza la frecuencia del segundo formante cuyo rango es de 800 Hz a 2200 Hz. Mientras que la coordenada y toma valores entre 0 y el alto de la imagen. El radio se calcula usando la frecuencia del tercer formante, esta frecuencia puede tomar valores entre 2000Hz y 3000Hz. En tanto que el radio puede tomar valores entre 0 y un octavo del ancho de la imagen. En el caso del color que toma el círculo se utiliza el ancho de banda de los primeros tres formantes. El rango del ancho de banda de los formantes abarca desde 0 hasta los 400Hz, mientras que el rango de los colores rojo, verde y azul es desde 0 hasta 1. En la linea 8 la función clip se encarga de volver los valores RGB en cero en caso de ser menores de 0 o en 1 en caso de ser mayores de 1. Además, esta función se encarga de empaquetar los valores de los colores en un arreglo que es nombrado como color.

Los valores de los rangos en los que se encuentran las frecuencias de los primeros tres formantes, así como el rango del ancho de banda de los formantes fueron tomados con base en la experimentación al analizar los formantes de un conjunto de muestras de los

audios de la base de datos.

Algoritmo 3 Conversor de formantes a circulo Modelo I

Entrada: $marco$, $ancho$, $alto$
Salida: x, y, r, $color$

1: $maxR \leftarrow ancho/8$
2: $x \leftarrow (marco.F[0] - 100) * ancho)/(1000)$
3: $y \leftarrow (marco.F[1] - 800) * alto)/(2200)$
4: $r \leftarrow (marco.F[2] - 2000) * maxR)/(3000)$
5: $R \leftarrow 1 - marco.BW[0]/400$
6: $G \leftarrow 1 - marco.BW[1]/400$
7: $B \leftarrow 1 - marco.BW[2]/400$
8: $color \leftarrow clip([R, G, B], 0, 1)$
9: **devolver** x, y, r, $color$

En la Figura 4.9 se muestra una imagen creada con el Modelo I, al utilizar los formantes de 200 marcos. En la imagen se puede observar como en las zonas donde se acumulan los círculos el color se comienza a obscurecer hasta que se forman zonas sombreadas en la imagen. Esto se debe a que cuando dos círculos ocupan un mismo lugar se genera un nuevo color que es una mezcla de ambos, pero al acumular muchos círculos en una zona se generan colores cada vez más obscuros.

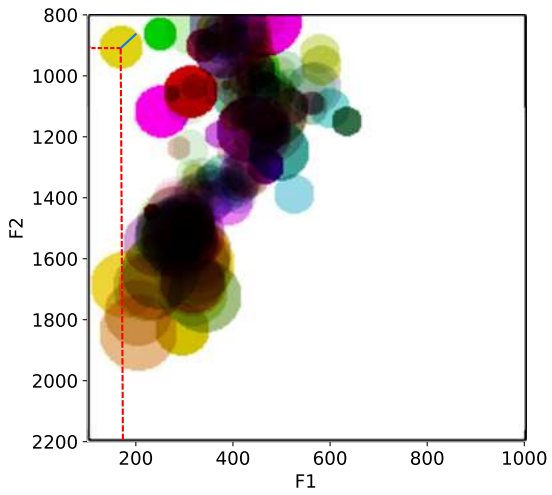


Figura 4.8: En la imagen se puede apreciar como las coordenadas del centro del circulo están relacionado con los primeros dos formantes, también en la imagen se puede notar una linea azul que representa al radio del círculo, el cual se encuentra relacionado con el tercer formante. Además, se puede observar como en las zonas donde se acumulan los círculos se generan sombras.

4.3.3. Modelo II

Cuando se utiliza el Modelo II para generar las imágenes se obtienen los parámetros de los círculos utilizando los primeros tres formantes de la voz. La frecuencia del primer y segundo formante se usan para calcular las coordenadas x y y del centro del círculo como en el Modelo I. Sin embargo, la principal diferencia entre el Modelo I y el Modelo II es que en el caso del Modelo II todos los círculos de la imagen tienen el radio del mismo tamaño. Debido a esto el Algoritmo 3 y el Algoritmo 4 son muy similares, aunque en el Algoritmo 4 el radio no se calcula.

El Modelo II fue creado para observar como son afectados los resultados en las pruebas a la red neuronal cuando no se generan en las imágenes las zonas sombreadas. Estas sombras en las imágenes se generan debido a la acumulación de círculos con radios de distintos tamaños, por lo que se propone utilizar una radio de 3 píxeles en todos los círculos de las imágenes. De esta manera es posible comparar los resultados de las pruebas utilizando el Modelo I con los resultados al utilizar el Modelo II.

Algoritmo 4 Conversor de formantes a circulo Modelo II

Entrada: *marco*, *ancho*, *alto*
Salida: x, y, *color*
1: $x \leftarrow (marco.F[0] - 100) * ancho)/(1000)$
2: $y \leftarrow (marco.F[1] - 800) * alto)/(2200)$
3: $R \leftarrow 1 - marco.BW[0]/400$
4: $G \leftarrow 1 - marco.BW[1]/400$
5: $B \leftarrow 1 - marco.BW[2]/400$
6: $color \leftarrow clip([R, G, B], 0, 1)$
7: **devolver** x, y, *color*

En la Figura 4.9 se muestra una imagen creada con el Modelo II, al utilizar los formantes de 200 marcos. Para crear la imagen se utilizó un radio con tamaño de 3 píxeles, por lo que los círculos en la imagen se encuentran más separados y no generan grandes zonas sombreadas en la imagen. Aunque, debido a la cercanía entre los círculos si se generan combinaciones de colores.

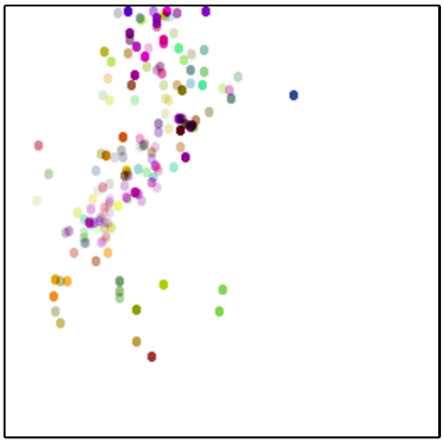


Figura 4.9: En la imagen se puede notar como debido a que el radio de los círculos es pequeño no se generan sombras.

4.3.4. Modificación tipo telaraña

Las zonas sombreadas en las imágenes que se generan con el Modelo I son de ayuda en la identificación de parlantes, por lo que una opción a considerar es la de generar sombras artificiales que ayuden a amplificar este comportamiento. En el caso de este proyecto se propone realizar una modificación al Algoritmo 2, ya que este se encarga de generar las imágenes. A esta modificación se le dio el nombre de telaraña. La telaraña se crea al utilizar los centros de los círculos como aristas que sirven para dibujar líneas. La telaraña está compuesta por un conjunto de líneas, donde cada línea conecta los centros de dos círculos. La creación de la telaraña comienza cuando se dibujan los círculos, ya que en el momento que se dibuja un círculo se almacena la posición de su centro en una lista llamada *centros*. Cuando se terminan de dibujar todos los círculos se procede a dibujar las líneas que conforman a la telaraña. Estas líneas son dibujadas utilizando dos centros como extremos de la línea. La primer línea se dibuja utilizando el primer y el segundo centro de la lista, mientras que la segunda línea se dibuja al usar el segundo y el tercer centro de la lista. El proceso sigue sucesivamente hasta que se dibuja la última línea utilizando los centros del penúltimo círculo y del último círculo. El Algoritmo 5 está basado en el Algoritmo 2, solo se realizaron unas modificaciones para poder dibujar la telaraña. La primera de las modificaciones se presenta en la línea 10, ya que se toma las coordenadas del centro del

círculo y se agregan al final de una lista llamada centros. Posteriormente en la línea 12 se calcula la longitud de la lista y con la ayuda de un ciclo se dibujan las líneas entre los centros consecutivos de la lista.

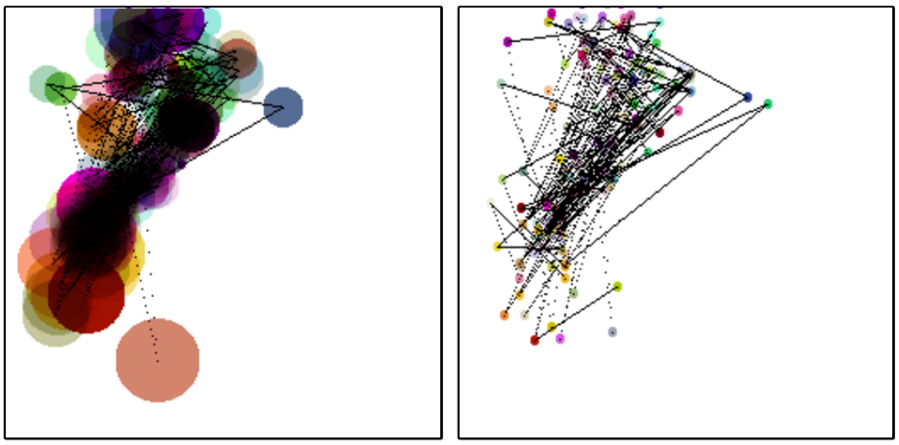
La función *dibujaLinea* toma las coordenadas de los centros de dos círculos para dibujar una linea entre estos. Aunque, en caso de que la pendiente de la linea sea positiva se dibuja un linea completa, pero en caso contrario se dibuja una linea punteada. Esto ayuda a disminuir la cantidad de píxeles a modificar y permite distinguir el sentido de las lineas.

Algoritmo 5 Generador con modificación telaraña

Entrada: *parlante*, *numCírculos*, *bandEnt*, *lista*, *ancho*, *alto*
Salida: *imagen*

1: $imagen \leftarrow nueva(ancho, alto, 3)$
2: $centros \leftarrow []$
3: **para** $i \leftarrow 0, numCírculos$ **hacer**
4: $índiceAudio \leftarrow indAleatorio(parlante)$
5: $N \leftarrow longitud(lista[índiceAudio]) - 1$
6: $indMarco \leftarrow aleatorio(N)$
7: $marco \leftarrow lista[índiceAudio][indMarco]$
8: $x, y, r, color \leftarrow FormantesACirculo(marco, ancho, alto)$
9: $dibujaCirculo(imagen, x, y, r, color)$
10: $agregaAlFinal(centros, [x, y])$
11: **fin para**
12: $L \leftarrow longitud(centros)$
13: $l \leftarrow 0$
14: **mientras** $l + 1 < L$ **hacer**
15: $dibujaLinea(imag, centro[l], centro[l + 1])$
16: $l \leftarrow l + 1$
17: **fin mientras**
18: **devolver** *imagen*

En la Figura 4.10 se muestran dos imágenes, la imagen a) muestra el resultado de utilizar la telaraña con el Modelo I, mientras que en la imagen b) se puede observar el resultado de utilizar la telaraña con el Modelo II. En ambos casos se nota la sombra que es creada por las lineas, en el caso de la imagen creada usando el Modelo I la sombra creada por las lineas se mezcla con la sombra que se crea por la acumulación de círculos. Por otro lado, en el caso de la imagen creada con el Modelo II, al aumentar la cantidad de círculos también incrementa la cantidad de lineas y pueden obstruir por completo el color de los círculos llegando a ser contraproducente.



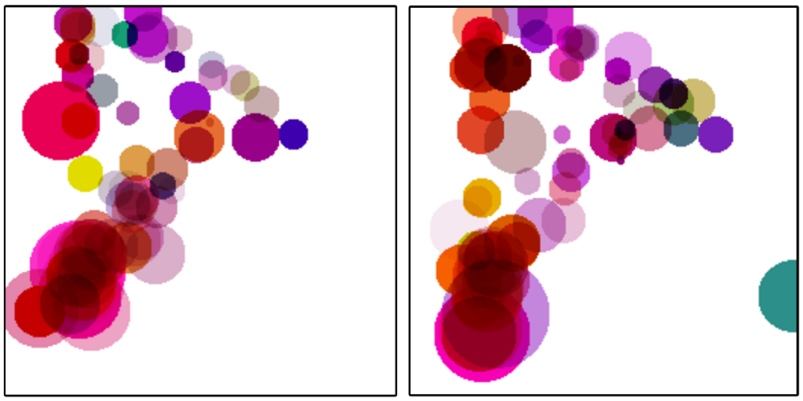
(a) ***Modelo I con telaraña*** **(b)** ***Modelo II con telaraña***

Figura 4.10: En las imágenes se puede observar el resultado de emplear la telaraña. En la imagen a) se puede notar como la telaraña complementa a la sombra generada por la acumulación de círculos. En la imagen b) se puede apreciar como se genera una sombra artificial que imita el resultado de la acumulación de círculos.

4.4. Imágenes obtenidas a partir de los audios

Las imágenes pueden generarse usando el Modelo I o el Modelo II para convertir los formantes de la voz en parámetros de círculos. Además, al generador de imágenes se le puede agregar la telaraña, por lo que en este trabajo se presentan cuatro formas de crear las imágenes. En esta sección se muestran imágenes generadas a partir de los audios de los parlantes Aaron y Coria. Al utilizar las imágenes de dos parlantes distintos se pretende tener un punto de comparación que ayude a identificar similitudes, diferencias entre imágenes de parlantes distintos. Mientras que para poder identificar las similitudes entre imágenes del mismo parlante se utilizan una imagen generada con los audios del conjunto de entrenamiento y una imagen generada con los audios del conjunto de prueba.

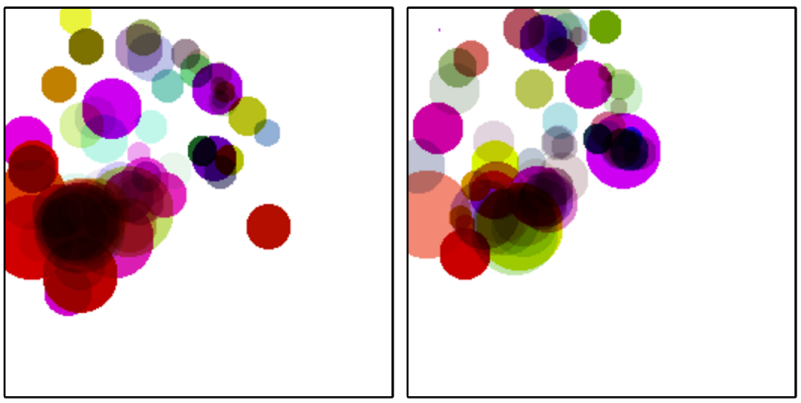
En la Figura 4.11 se muestran dos imágenes que fueron creadas con los audios del parlante Aaron usando el Modelo I. Al observar la imagen a) y la imagen de b) se puede notar que los círculos forman cúmulos que al estar juntos forman una figura similar a una letra 4 invertido. En ambas imágenes los cúmulos ocupan espacios similares y de igual manera las formas que generan los cúmulos de círculos se asemejan.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

Figura 4.11: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Aaron generadas con el Modelo I usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.

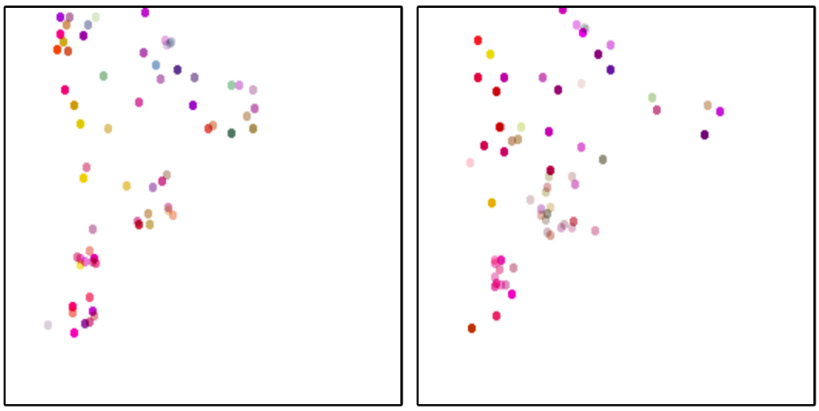
En el caso de la Figura 4.12 se muestran dos imágenes que fueron creadas con los audios del parlante Coria al usar el Modelo I. Al comparar las imágenes a) y b), se nota que los cúmulos de círculos ocupan zonas similares y forman una figura similar a una elipse con unos círculos en el centro. Además, cuando se comparan las imágenes de la Figura 4.11 con las imágenes de la Figura 4.12 se puede notar que las imágenes que pertenecen al mismo parlante tienen una mayor similitud que las imágenes generadas por parlantes distintos.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

Figura 4.12: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Coria generadas con el Modelo I usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.

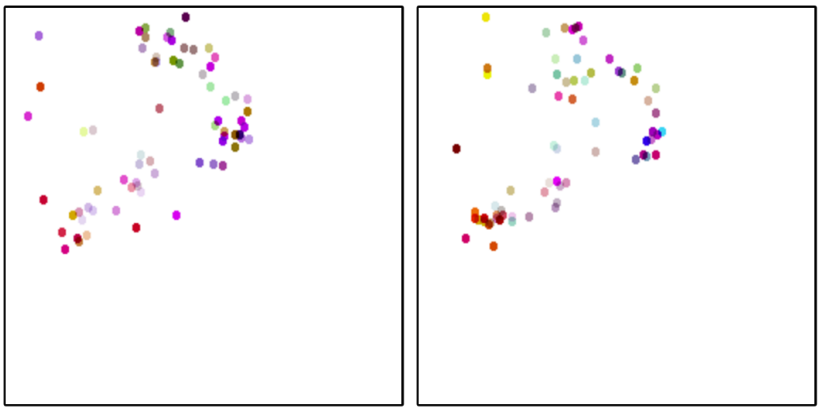
Las imágenes que se generan utilizando el Modelo II necesitan especificar el tamaño que tendrán los radios de los círculos. En este caso se opto por utilizar un radio de 3 píxeles, ya que permite tener círculos pequeños que no generan zonas sombreadas. La Figura 4.13 muestra dos imágenes que fueron generadas empleando los audios del parlante Aaron usando el Modelo II. Cuando se compara la imagen a) con la imagen b) es más difícil encontrar similitudes que al utilizar el Modelo I, sin embargo, aún se puede notar que los cúmulos de círculos ocupan zonas similares.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

Figura 4.13: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Aaron generadas con el Modelo II usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.

Por otro lado, en la Figura 4.14 se muestran dos imágenes que fueron creadas a partir de los audios del parlante Coria al usar el Modelo II. Las imágenes presentan similitudes entre ellas sobre todo si se realiza una comparación con las imágenes de la Figura 4.12 que fueron generadas con los audios del parlante Aaron. En las imágenes del parlante Aaron los cúmulos de círculos llegan casi hasta la parte inferior de la imagen mientras que las imágenes del parlante Coria llegan hasta la mitad de la imagen. Por lo que parece que los cúmulos en la imagen del parlante Coria se quedaran en el primer cuadrante de la imagen.

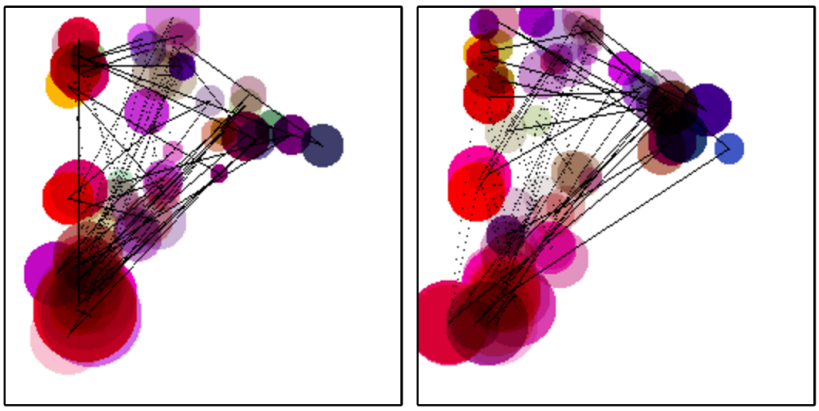


(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

Figura 4.14: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Coria generadas con el Modelo II usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.

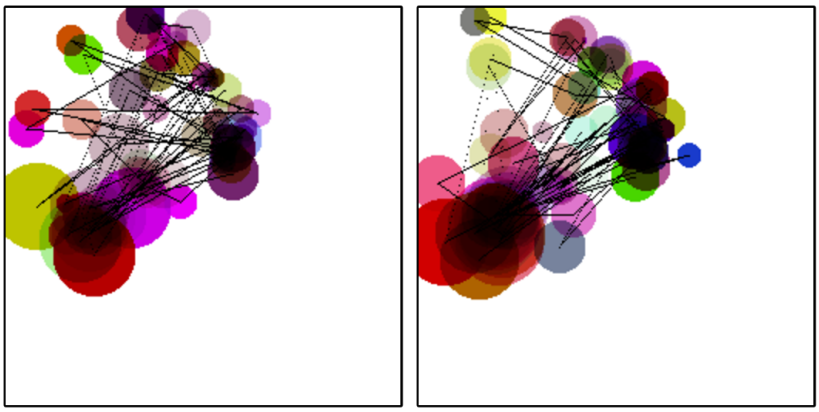
Las imágenes que son generadas utilizando la modificación telaraña en esencia son similares a las presentadas anteriormente, la principal diferencia es que las lineas que son agregadas crean una sombra que cubre parte de las imágenes. Además, al igual que las imágenes mostradas anteriormente estas son generadas utilizando los audios del parlante Aaron y el parlante Coria. Para comenzar la Figura 4.15 muestra dos imágenes que fueron generadas por el Modelo I con la telaraña. La imagen a) parece distinta de la imagen b), sin embargo, ambas imágenes tienen similitudes como las zonas en las que se acumulan los círculos, así como los radios de los círculos en las zonas. Mientras que las lineas de las telarañas son en esencia distintas, pero cumplen con el trabajo de unir los cúmulos de círculos e incrementar la sombra en el interior de la figura que se genera al considerar todos los cúmulos de círculos.

En la Figura 4.16 se muestran las imágenes creadas con los audios del parlante Coria usando el Modelo I con la telaraña. Al observar las imágenes, tanto del parlante Aaron como del parlante Coria se puede apreciar como las líneas no son iguales incluso entre imágenes del mismo parlante, pero no se requiere que sean iguales ya que su función es la de unir los cúmulos de círculos para generar una figura en la imagen.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

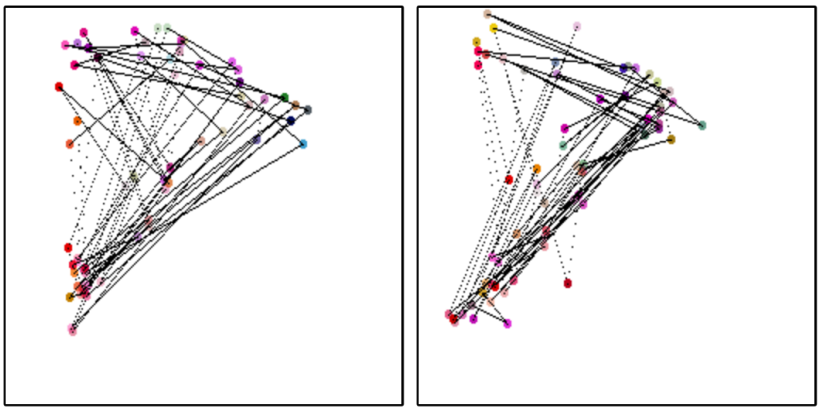
Figura 4.15: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Aaron generadas con el Modelo I con telaraña usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

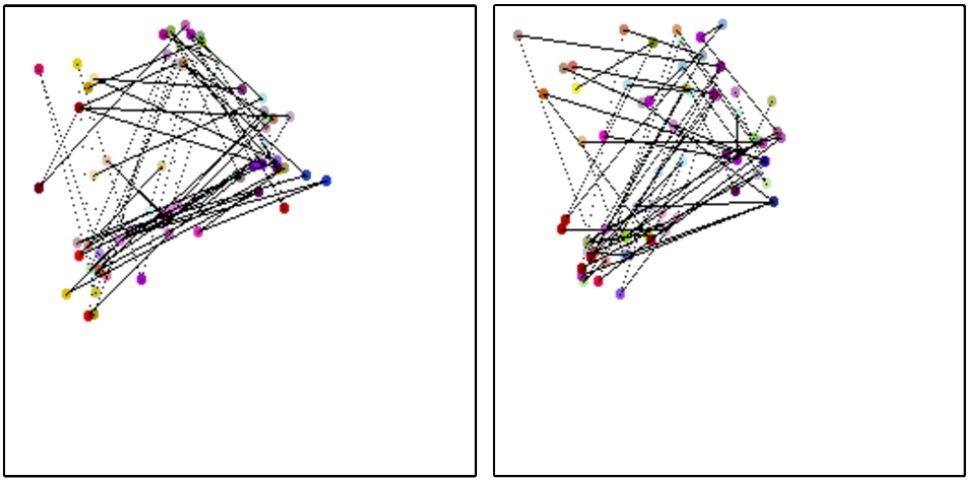
Figura 4.16: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Coria generadas con el Modelo I con telaraña usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.

La comparación también se lleva a cabo utilizando imágenes generadas con el Modelo II empleando la telaraña. Las imágenes obtenidas a partir de los audios del parlante Aaron se muestran en la Figura 4.17. Para generar estas imágenes se utiliza un tamaño de radio para los círculos de 3 píxeles al igual que las imágenes generadas sin la telaraña.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

Figura 4.17: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Aaron generadas por el Modelo II con telaraña usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.



(a) ***Conjunto de entrenamiento*** **(b)** ***Conjunto de prueba***

Figura 4.18: Para comparar y encontrar similitudes se han tomado dos imágenes del parlante Coria generadas por el Modelo II con telaraña usando 70 marcos. La imagen a) pertenece al conjunto de entrenamiento, mientras que la imagen b) pertenece al conjunto de prueba.

Al comparar las imágenes tanto de la Figura 4.17 como de la Figura 4.18 se puede apreciar una mayor relevancia de las telarañas, ya que los círculos tienen un papel menor. A pesar de esto se puede apreciar como las telarañas ocupan zonas similares, esto se debe a

que una zona densa en cuestión de círculos será una zona densa en cuestión de lineas. Esto se puede notar al encontrar similitudes en las imágenes del mismo parlante pero encontrando diferencia en las imágenes de parlantes distintos.

4.5. Arquitectura empleada en la red neuronal

La red neuronal empleada está compuesta por dos etapas. La primer etapa consta de las capas convolucionales, mientras que la segunda etapa consta únicamente de capas densas. En total se emplearon cuatro capas convolucionales que se encargan de extraer las características de la imagen. La capa inicial de la primer etapa cuenta con 16 filtros de tamaño tres por tres. En la segunda capa se utilizan 32 filtros de tamaño tres por tres. Mientras que la tercer capa emplea 32 filtros de tamaño dos por dos. Por último la cuarta capa convolucional cuenta con 64 filtros de tamaño dos por dos. Todas las capas convolucionales de la primer etapa aplican los filtros con $same - padding$ y utilizan la función de activación $ReLU$. Además, después de cada una de las capas convolucionales se aplica un $maxpooling$ de tamaño dos por dos.

La segunda etapa de la red está constituida por tres capas densas. Las primeras dos capas cuentan con 128 neuronas cada una y estas neuronas utilizan la función de activación $ReLU$. Por otro lado, la tercer capa es la capa de salida y está constituida por 21 neuronas con función de activación $softmax$, una neurona por cada parlante en la base de datos. En la Figura 4.19 se muestra el resumen de la arquitectura que proporciona Keras, en esta figura se puede ver la cantidad de parámetros utilizados, así como la entrada y salida en cada capa de la red.

La arquitectura empleada se obtuvo después de realizar una serie de pruebas en las que se modificaban parámetros como el tamaño de los filtros el número de capas de convolución, el tamaño de vecindario en el $maxpooling$ y el número de neuronas en las capas densas. El proceso iterativo consistió en modificar los parámetros, entrenar la red neuronal y observar los resultados que se obtenían para tratar de encontrar una mejor arquitectura.

```
Model: "ForenCNN_Serie1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
conv2d (Conv2D)              (None, 256, 256, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 128, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 128, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 64, 32)        4128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 64)        8256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 16384)             0
_________________________________________________________________
dense (Dense)                (None, 128)               2097280
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_2 (Dense)              (None, 21)                2709
=================================================================
Total params: 2,133,973
Trainable params: 2,133,973
Non-trainable params: 0
```

Figura 4.19: Resumen de la arquitectura proporcionado por Keras.

4.6. Conclusiones del capítulo

Las generación de imágenes que caractericen a los parlantes es un problema que puede ser resulto de múltiples maneras. En el caso de la propuesta realizada en esta tesis se utilizan los formantes de la voz debido a que proporcionan información de las características de los parlante, así como, patrones de comportamiento. Aunque, el proceso está abierto a una gran variedad de posibilidades como utilizar de forma distinta los formantes al momento de convertirlos en parámetros de los círculos, agregar más información a las imágenes o incluso utilizar algún método de extracción de características distinto como los MFCC. [Diagrama del proceso para identificar marcos con sonido vocalizado]Diagrama del proceso para identificar marcos con sonido vocalizado.

Capítulo 5

Resultados

En este capítulo se describen las pruebas que se realizaron a la implementación, el hardware empleado para las pruebas y se exponen los resultados obtenidos en las pruebas. Durante la explicación de las pruebas se abordan aspectos relevantes del entrenamiento como la función de costo, el optimizador y la métrica utilizada. Además de realizar una breve explicación de los conjuntos de imágenes empleados y la forma en la que se dividieron para las pruebas. Por otro lado, los resultados de las pruebas se muestran por medio de tres gráficas. En la primera se muestran los mejores resultados obtenidos por la red neuronal al entrenar con el conjunto de prueba, en la segunda gráfica se muestran los peores resultados y la tercer gráfica muestra los promedios de los resultados. La información reportada en las gráficas es complementada con la ayuda de matrices de confusión que son creadas a partir de los mejores resultados de las pruebas para cada modelo al utilizar imágenes creadas a partir de los formantes de 200 marcos.

5.1. Experimentos

Los experimentos realizados consisten en probar el comportamiento de la red neuronal después de ser entrenada con imágenes generadas por el Modelo I, el Modelo II, así como el Modelo I y el Modelo II con la telaraña. Las imágenes son creadas utilizando los algoritmos mencionados en el capítulo 4, aunque por cada modelo se crean conjuntos de imágenes con distinto número de marcos. Por lo que para cada modelo se tienen conjuntos de imágenes que han sido generadas empleando 14 cantidades distintas de marcos (30, 50, 70, 100, 130, 150, 170, 200, 230, 250, 270, 300, 350 y 400). Por otra lado, los conjuntos de entrenamiento están compuestos por 50 imágenes por cada parlante dando un total de 1050 imágenes, donde las imágenes generadas tienen un tamaño de 256×256.

En las pruebas que se llevaron a cabo la red neuronal es entrenada utilizando el conjunto de entrenamiento, que a su vez es dividido para utilizar el 20 % de las imágenes como el conjunto de validación. En el entrenamiento se utiliza el optimizador Adam en conjunto con la función de costo *categorical_crossentropy*. Al utilizar las librerías TensorFlow y Keras la función *categorical_crossentropy* hace referencia a utilizar la función de costo *crossentropy* junto con la función de activación $Softmax$. Por otra parte, la red neuronal es entrenada durante 20 épocas utilizando un batch_size de 10 y para evitar el sobre entrenamiento se utiliza la función *earlystopping* con una paciencia de seis. Como métrica se utiliza accuracy, que consiste en tomar la cantidad de imágenes que han sido clasificadas de manera correcta y dividir esta cantidad entre el número total de imágenes.

5.2. Resultados

Las pruebas fueron realizadas en una computadora que cuenta con un procesador intel core i5 10400F, 16GB de memoria RAM y una tarjeta gráfica Nvidia RTX 2060. En la implementación se ha empleado Python enconvinación con Keras y TensorFlow en su versión para GPU. Generar una imagen toma entre 16 ms y 145 ms, el tiempo requerido depende del modelo empleado y la cantidad de marcos por imagen. Los tiempos de entrenamiento por cada modelo rondan 40s, puesto que en promedio toma 2s por época. Aunque, en algunas ocasiones $early - stopping$ detiene el entrenamiento antes de llegar a las 20 épocas, por lo

que el tiempo de entrenamiento puede ser menor. Además, a la red neuronal le toma 1s clasificar un conjunto de 1050 imágenes.

Debido a que los parámetros iniciales de la red neuronal son dados de manera aleatoria y durante el entrenamiento se puede caer en un mínimo local que no refleje por completo su funcionamiento, se ha optado por entrenar a la red neuronal 30 veces para cada conjunto de imágenes. Después de cada entrenamiento el comportamiento de la red neuronal es probado utilizando el conjunto de prueba, que está compuesto por 1050 imágenes al igual que el conjunto de entrenamiento. A partir de los resultados obtenidos de la red neuronal se han creado tres gráficas. La primera de ellas contiene los mejores resultados obtenidos, la segunda gráfica contiene los peores resultados y la tercer gráfica muestra el promedio de los resultados. En las etiquetas de las gráficas, los resultado de las pruebas a la red neuronal utilizando imágenes con la modificación telaraña es agregado al nombre del modelo la letra T.

Los mejores resultados obtenido por la red neuronal con el conjunto de entrenamiento se muestran en la gráfica de la Figura 5.1. Esta gráfica nos permite observar el máximo accuracy que obtiene la red neuronal al entrenar con cada tipo de imágenes, ya se toman los mejores resultados obtenidos en las 30 pruebas. Al observar la gráfica se puede notar como al aumentar la cantidad de marcos utilizados para generar las imágenes, se incrementa el porcentaje de imágenes que son correctamente clasificadas. Cada uno de los modelos llega a su máximo nivel de accuracy con una cantidad distinta de marcos por imagen. En el caso del Modelo I son necesarios 200 marcos y se obtiene un 93 %, para el Modelo II son necesarios 250 marcos y consigue un 93 %. Mientras que el Modelo I con la telaraña requiere de 130 marcos para conseguir un 96 %. Por último, en el caso del Modelo II con la telaraña se necesitan 170 marcos y obtiene 96 %. Además, de todos los modelos solo el Modelo I con telaraña mantiene su máximo, los demás comienzan a decrementar al aumentar la cantidad de marcos por imagen.

Los mejores resultados obtenidos alcanzaron el 96 % y fueron obtenidos tanto por el Modelo I como por el modelo II al utilizar la modificación telaraña. Además, de todos los modelos solo el Modelo I con telaraña mantiene su máximo, los demás comienzan a decrementar al aumentar la cantidad de marcos por imagen. Por otro lado, el Modelo II

con la telaraña es el más inestable cambiando de valor constantemente.

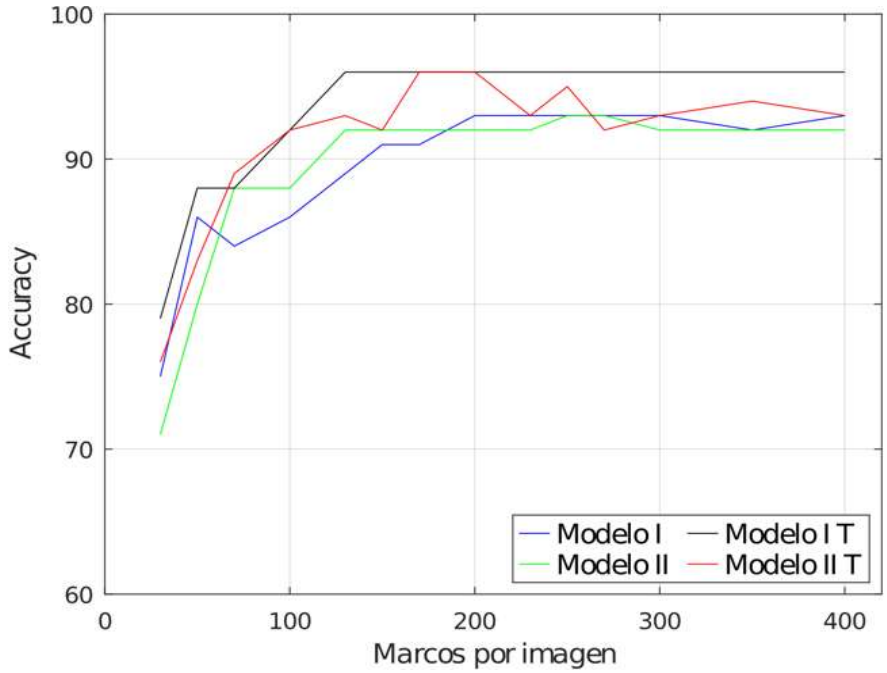


Figura 5.1: En la gráfica se muestran los mejores resultados que se obtuvieron en las pruebas y como se puede observar el Modelo I con telaraña destaca sobre los demás.

En contraste la gráfica de la Figura 5.2 muestra los peores resultados obtenidos al entrenar la red neuronal. La gráfica permite apreciar cuales son los valores mínimos de accuracy que se obtienen al utilizar cada modelo con cierta cantidad de marcos. Esto nos permite apreciar de manera más amplia el comportamiento de la red neuronal, ya que se muestra cual es su peor desempeño al utilizar cada modelo. En la gráfica se puede notar como varían mucho los resultados con cada modelo. No obstante, los mejores resultados en la mayoría de los casos son obtenidos por el Modelo I y el Modelo II con la telaraña, teniendo un máximo de 92 % al utilizar imágenes generadas con 200 Marcos. Por otra parte, los peores resultados mostrados pertenecen a el Modelo I y el Modelo II, aunque todos los Modelos a partir de utilizar imágenes generadas con 100 marcos obtienen un accuracy entre 80 % y 92 %. Además, algo a resaltar es que los modelos dejan de mejorar e incluso comienzan a empeorar sus resultados al utilizar imágenes que fueron generadas con más de 200 marcos.

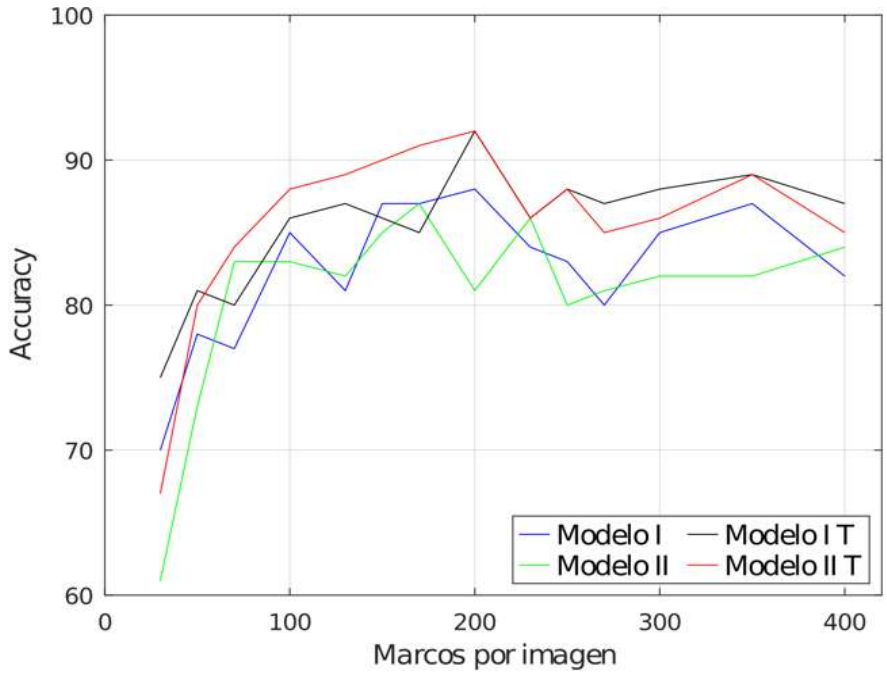


Figura 5.2: En la gráfica se muestran los peores resultados que se obtuvieron en las pruebas.

Con la finalidad de apreciar el comportamiento de la red neuronal desde otro ángulo, la Figura 5.3 muestra la gráfica del promedio de los resultados obtenidos con el conjunto de prueba. Esta gráfica nos permite visualizar en promedio que accuracy se obtiene al entrenar y probar la red neuronal 30 veces, lo que nos da información de que tan cercano se encuentra el valor de accuracy obtenido en las 30 pruebas de los mejores y de los peores resultados. En la gráfica se puede notar que los mejores resultados siguen siendo obtenidos por el modelo I y el Modelo II con la telaraña, teniendo su máximo en 94 % al utilizar imágenes creadas con 200 marcos. Aunque en este caso se puede observar como el Modelo II con la telaraña comienza a disminuir el valor de accuracy para las imágenes creadas con más de 200 marcos, llegando a obtener resultados similares a los del Modelo I y el Modelo II. Los peores resultados son obtenidos por el Modelo I y el Modelo II, pero a pesar de esto con imágenes creadas con más de 130 marcos obtienen valores de accuracy entre 85 % y 90 %. Al comparar las tres gráficas se puede notar que la gráfica mantiene más similitudes con la gráfica de los mejores resultados, por lo que en promedio se tiene un buen comportamiento.

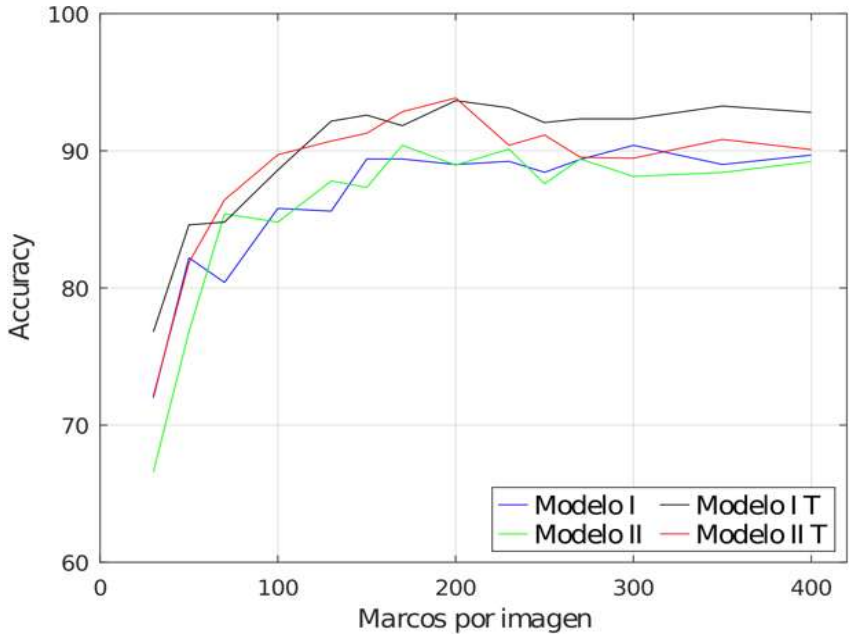


Figura 5.3: En esta gráfica se muestra el promedio de los resultados obtenidos en las pruebas, donde se puede notar como el comportamiento del Modelo I y el Modelo I con telaraña son similares solo se incremento el nivel de accuracy.

Las gráficas proporcionan información sobre el comportamiento de la red neuronal. Sin embargo, la métrica accuracy puede ser engañosa, por lo que para complementar la información se han creado matrices de confusión. Las matrices de confusión se suelen utilizar para observar el desempeño en algoritmos entrenados por medio de aprendizaje supervisado. En las matrices de confusión los renglones representan a las clases reales mientras que las columnas representan a las predicciones. Estas matrices permiten visualizar cuales son las clases que está confundiendo la red neuronal, ya que en la matriz sus celdas almacenan la cantidad de imágenes clasificadas de acuerdo a la predicción y la clase real. Esto permite que al observar los valores en la diagonal de la matriz se pueda observar la cantidad de imágenes clasificadas correctamente por cada clase. Las matrices fueron generadas utilizando los mejores resultados para cada modelo de generación de imágenes al usar 200 marcos, ya que es el punto en el que se obtiene los mejores resultados. En la Figura 5.4 se muestra la matriz de confusión generada utilizando el Modelo I. La matriz de confusión muestra como dos de los parlantes obtuvieron menos de 40 imágenes clasificadas de manera correcta. No

obstante, la mayoría de los parlantes tienen entre 40 y 50 imágenes clasificadas correctamente. En el caso del parlante Ernesto se confunde casi en su mayoría con la parlante Ma. Merari, mientras que en el caso del parlante Rene las confusiones se encuentran dispersas entre 6 parlantes.

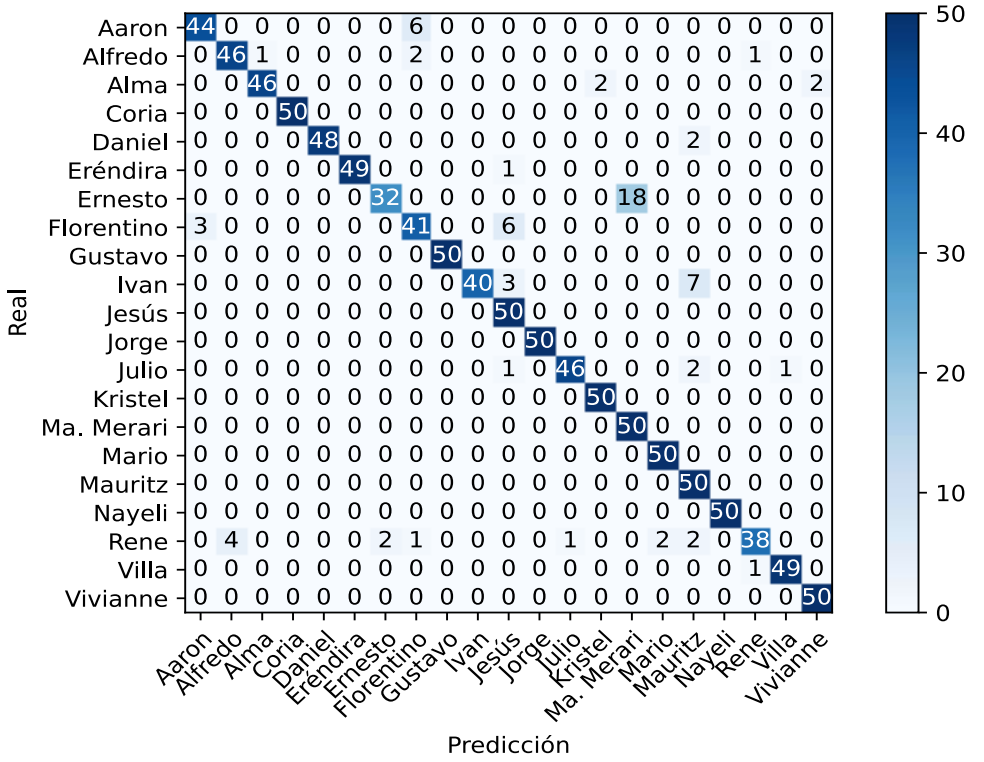


Figura 5.4: En la matriz de confusión del Modelo I utilizando 200 marcos se obtiene un accuracy de 93.23 % y se clasifican correctamente las 50 imágenes de 10 parlantes.

En la Figura 5.5 se muestra la matriz de confusión generada utilizando el Modelo II. En este caso la matriz de confusión muestra como tres de los parlantes obtienen menos de 40 imágenes clasificadas de manera correcta. El parlante Florentino obtiene solo 16 de las imágenes clasificadas correctamente y 21 imágenes son confundidas con el parlante Ivan. En el caso del parlante Ivan se clasifican correctamente 26 imágenes y la mayoría son confundidas con la parlante Eréndira, el parlante Florentino, el parlante Jesús y la parlante Nayeli. Mientras que la parlante Kristel es confundida con la parlante Alma y la parlante Nayeli. Esto implica que el modelo II obtiene un peor desempeño que el presentado en el Modelo I.

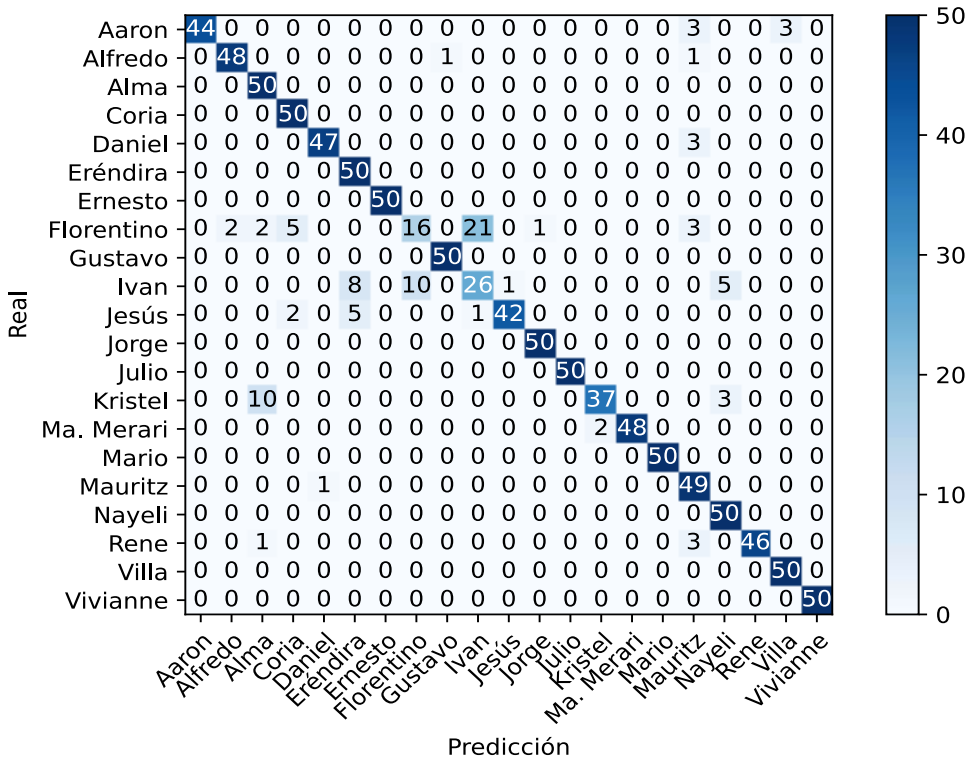


Figura 5.5: En la matriz de confusión del Modelo II utilizando 200 marcos se obtiene un accuracy de 90.76 % y se clasifican correctamente las 50 imágenes de 11 parlantes.

La matriz de confusión del Modelo I utilizando la telaraña con 200 marcos se muestra en la Figura 5.6. En la matriz se muestra como solo en el caso de uno de los parlantes se obtienen menos de 40 imágenes clasificadas de manera correcta, mientras que la mayoría de parlantes obtienen más de 45 imágenes bien clasificadas. En el caso del parlante Rene se confunde con el parlante Alfredo, el parlante Ernesto, el parlante Julio, el parlante Mario y el parlante Mauritz. Al comparar la matriz de confusión con la obtenida usando el Modelo I, se puede observar como se obtiene una mejora en la clasificación debido a que la telaraña acentúa más la sombra generada por los cúmulos de círculos.

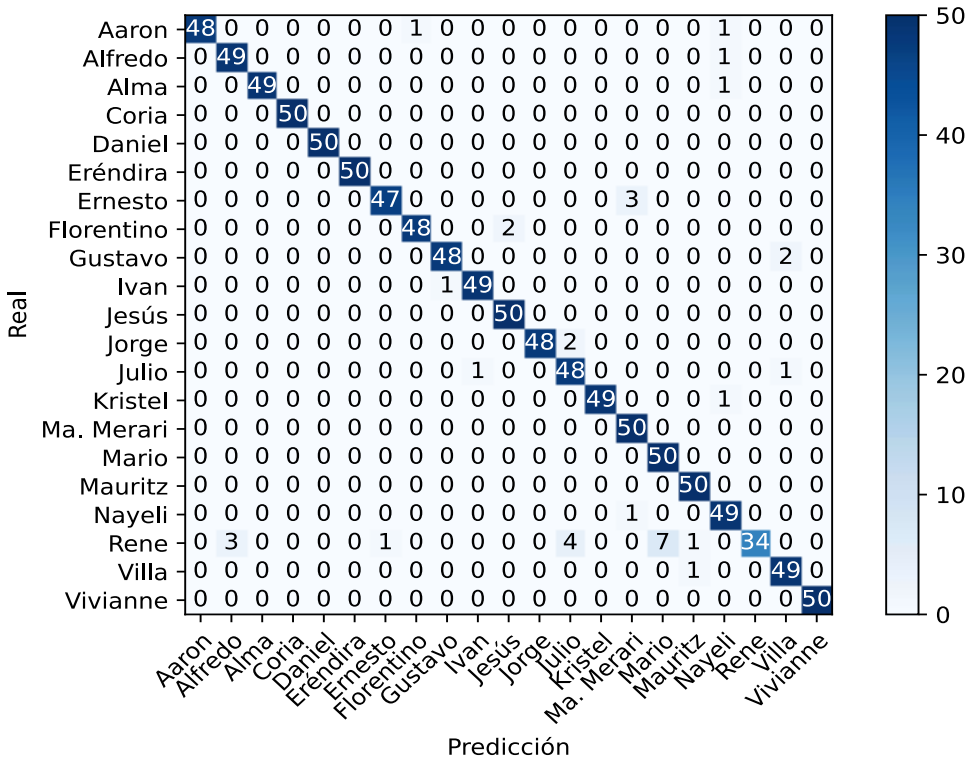


Figura 5.6: Esta matriz de confusión del Modelo I con telaraña utilizando 200 marcos tiene un accuracy de 96.6 % y se clasifican correctamente las 50 imágenes de 8 parlantes.

En la Figura 5.7 se muestra la matriz de confusión generada utilizando el Modelo II con la telaraña. En la matriz se puede observar como solo uno de los parlantes ha obtenido menos de 40 imágenes clasificadas de manera correcta. El parlante Rene suele confundirse con el parlante Florentino, el parlante Julio, el parlante Mario, el parlante Mauritz y el parlante Villa. No obstante, los resultados obtenidos mejoraron sustancialmente con respecto a los resultados obtenidos al no utilizar la telaraña cuando se generan las imágenes.

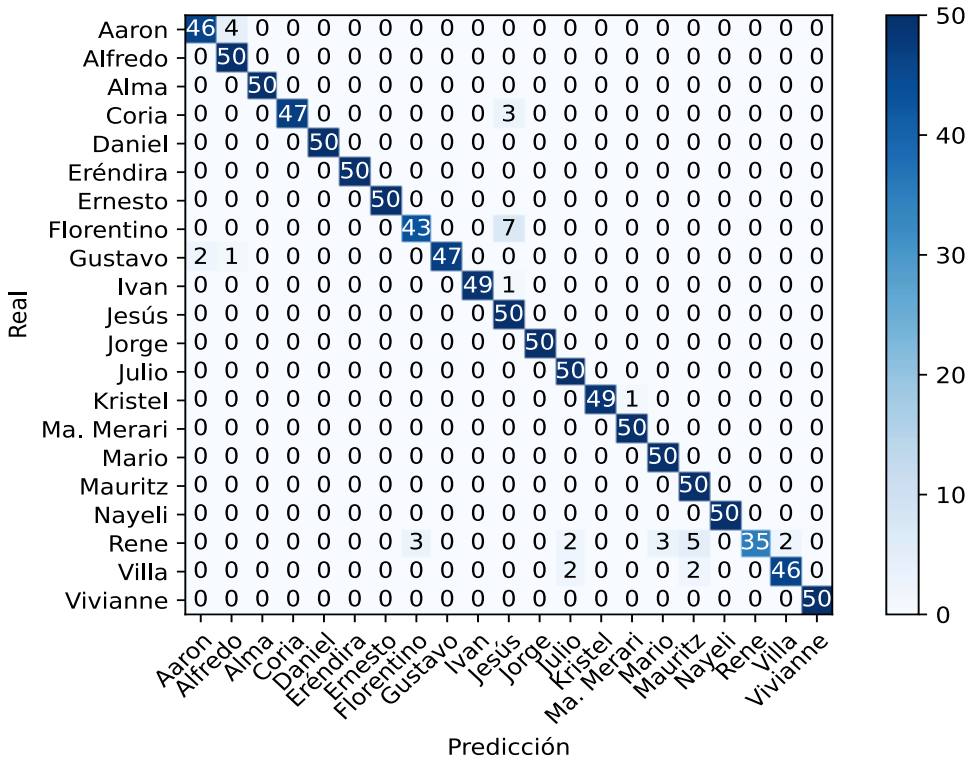


Figura 5.7: La matriz de confusión del Modelo II con telaraña utilizando 200 marcos cuenta con un accuracy de 96.38 % y se clasifican correctamente las 50 imágenes de 13 parlantes.

5.3. Conclusiones del capítulo

Los resultados obtenidos por la red neuronal al ser probadas con las imágenes generadas por los modelos han sido muy buenos. No obstante, se a podido observar como cuando se utilizan imágenes con la telaraña se obtienen los mejores resultados, destacando el Modelo I con la telaraña. En el caso del Modelo II con telaraña sus resultados empeoran al utilizar imágenes creadas con más de 200 marcos, por lo que puede que las lineas de la telaraña estén cubriendo a los pequeños círculos y por tanto eliminando información importante.

Capítulo 6

Conclusiones y Trabajos Futuros

"La ciencia nunca resuelve un problema sin crear otros 10 más."

George Bernard Shaw

6.1. Conclusiones

Los resultados obtenidos a través de las pruebas muestran que es posible realizar la identificación de parlantes texto-independiente utilizando el método propuesto en esta tesis. Nuestro sistema obtiene un 96 % de accuracy, aunque cuenta con un inconveniente. El sistema requiere que los audios utilizados para generar las imágenes contengan distintos sonidos vocalizados. Esto se debe a que para tener una buena representación del tracto vocal es necesario capturar sonidos en los que el tracto vocal adquiere formas distintas.

Antes de los experimentos no se sabia la cantidad de marcos que se necesita utilizar para generar las imágenes. Al observar los resultados obtenidos en las pruebas por los cuatro modelos se puede notar que cuando se utilizan 200 marcos por imagen se obtienen buenos resultados. No obstante, el Modelo I con la telaraña obtiene 96 % de accuracy a partir de los 130 marcos por imagen. Mientras que en el caso del Modelo II con telaraña se necesita 170 marcos por imagen para alcanzar un 96 %.

El método propuesto aún debe de ser explorado, ya que de momento no se han llevado a cabo pruebas importantes como utilizar bases de datos distintas que sean especializadas en el problema de la identificación de parlantes texto-independiente. Al probar el sistema con distintas bases de datos también deben ser comparados los resultados que se obtengan con los resultados de otros proyectos publicados. Además, es necesario realizar pruebas modificando algunos parámetros como el tamaño de las imágenes y los parámetros de la red neuronal, para optimizar el uso de los recursos.

6.2. Trabajos futuros

La propuesta realizada en esta tesis consiste en el desarrollo de un método de identificación de parlantes texto independiente, en el que se crean imágenes a partir de los formantes. Además se utiliza una red neuronal para extraer los patrones de las imágenes e identificar a los parlantes. Sin embargo, en esta propuesta no se han cubierto todas las posibilidades, por lo que la propuesta está abierta a mejoras.

1.- En las pruebas realizadas a la implementación se puede notar que la forma en la que se generan las imágenes a partir de los formantes puede afectar la cantidad de imágenes clasificadas correctamente. Por tanto, una de las modificaciones debe abordar la creación de distintos diseños de generación de imágenes.

2.- La segunda propuesta tiene que ver más con los formantes utilizados. En este trabajo el Modelo I utiliza la frecuencia y el ancho de banda de los primeros tres formantes, mientras que el Modelo II utiliza la frecuencia de los primeros dos formantes y el ancho de banda de los primeros tres formantes. La modificación consiste en utilizar el cuarto y el quinto formante de la voz para realizar las imágenes, esto se debe a que los últimos formantes de la voz están más relacionados con las proporciones del tracto vocal del parlante. No obstante, no se podría utilizar la base de datos empleada en esta tesis, ya que el quinto formante se encuentra por encima de los 4000Hz y lo audios fueron muestreados a 8000Hz.

3.- El método actual solo utiliza los sonidos vocalizados, por lo que una opción a considerar es el desarrollo de un método para complementar las imágenes utilizando al mismo tiempo los sonidos que no son vocalizados.

4.- Probar el comportamiento del sistema al utilizar conjuntos de imágenes de tamaño mayor a 256×256, así como imágenes de tamaño menor a 256×256.

5.- El sistema debe ser probado utilizando distintas bases de datos como la base de datos TIMIT o la base de datos ELSDSR. Los resultados obtenidos pueden ampliar la perspectiva del funcionamiento del método, ya que son bases de datos utilizadas comúnmente en el campo de la identificación de parlantes, lo cual proporciona un punto de comparación. Además, los individuos que participaron en la creación de las bases de datos mencionadas son angloparlantes de distintas regiones.

6.- Diseñar una red neuronal más completa, que no dependa del sistema de generación de imágenes para realizar la identificación de parlantes.

Referencias

[Aalto et al., 2018] Aalto, D., Malinen, J., and Vainio, M. (2018). Formants. In *Oxford Research Encyclopedia of Linguistics*.

[Al-Azzeh et al., 2020] Al-Azzeh, J., Zahran, B., Alqadi, Z., Ayyoub, B., and Mesleh, M. (2020). Creating color image signature based on laplacian equation. *JOIV: International Journal on Informatics Visualization*, 3.

[Albawi et al., 2017] Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee.

[Almaadeed et al., 2016] Almaadeed, N., Aggoun, A., and Amira, A. (2016). Text-independent speaker identification using vowel formants. *Journal of Signal Processing Systems*, 82(3):345–356.

[Andrade and Ibarrola,] Andrade, J. F. R. and Ibarrola, J. A. C. Uso de una discretización de la transformada de fourier para identificación de individuos por voz.

[Arias et al., 2019] Arias, V., Salazar, J., Garicano, C., Contreras, J., Chacón, G., Chacín-González, M., Añez, R., Rojas, J., and Bermúdez-Pirela, V. (2019). Una introducción a las aplicaciones de la inteligencia artificial en medicina: Aspectos históricos. *Revista Latinoamericana de Hipertensión*, 14(5):590–600.

[Ashar et al., 2020] Ashar, A., Bhatti, M. S., and Mushtaq, U. (2020). Speaker identification using a hybrid cnn-mfcc approach. In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pages 1–4. IEEE.

[Camacho, 2015] Camacho, I. E. (2015). Huella genética vs. huella dactilar. *Archivos de Criminología, Seguridad Privada y Criminalística*, (14):7–8.

[Camarena-Ibarrola et al., 2020] Camarena-Ibarrola, A., Figueroa, K., and García, J. (2020). Speaker identification using entropygrams and convolutional neural networks. In *Mexican International Conference on Artificial Intelligence*, pages 23–34. Springer.

[Camarena-Ibarrola et al., 2017] Camarena-Ibarrola, A., Luque, F., and Chavez, E. (2017). Speaker identification through spectral entropy analysis. In *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6. IEEE.

[Castro, 2019] Castro, M. (2019). *Identificación de Parlantes Independiente del Texto Utilizando los Formantes de las Vocales*. PhD thesis, Facultad de Ingeniería Eléctrica, División de Estudios de Posgrado. Universidad Michoacana de San Nicolás de Hidalgo.

[Celdrán et al., 1995] Celdrán, E. M. et al. (1995). En torno a las vocales del español: análisis y reconocimiento. *Estudios de fonética experimental*, pages 195–218.

[Champod and Meuwly, 2000] Champod, C. and Meuwly, D. (2000). The inference of identity in forensic speaker recognition. *Speech communication*, 31(2-3):193–203.

[Chen et al., 1996] Chen, K., Xie, D., and Chi, H. (1996). A modified hme architecture for text-dependent speaker identification. *IEEE Transactions on Neural Networks*, 7(5):1309–1313.

[CONDUSEF, 2021] CONDUSEF (2021). `https://www.condusef.gob.mx/`, 23 de Junio de 2021.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366.

[Gallardo, 2013] Gallardo, B. T. (2013). La voz y nuestro cuerpo: un análisis funcional. *Revista de Investigaciones en Técnica Vocal*, 1:40–58.

[Goodfellow et al., 2016] Goodfellow, I. J., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA, USA. `http://www.deeplearningbook.org`.

[Hagan et al., 2014] Hagan, M., Demuth, H., Beale, M., and De Jesús, O. (2014). *Neural Network Design*. Martin Hagan.

[Hong and Jain, 1998] Hong, L. and Jain, A. (1998). Integrating faces and fingerprints for personal identification. *IEEE transactions on pattern analysis and machine intelligence*, 20(12):1295–1307.

[Hualde et al., 2010] Hualde, J. I., Olarrea, A., Escobar, A. M., Travis, C. E., and Sanz, C. (2010). *Introducción a la lingüística hispánica*. Cambridge University Press.

[Ignacio G.R. Gavilán, 2021] Ignacio G.R. Gavilán (2021). `https://ignaciogavilan.com`, 23 de Junio de 2021.

[Jahangir et al., 2020] Jahangir, R., Teh, Y. W., Memon, N. A., Mujtaba, G., Zareei, M., Ishtiaq, U., Akhtar, M. Z., and Ali, I. (2020). Text-independent speaker identification through feature fusion and deep neural network. *IEEE Access*, 8:32187–32202.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Latorre et al., 2009] Latorre, C. C. et al. (2009). Comportamiento de los formantes vocálicos respecto a la apertura mandibular y el género.

[Lawrence et al., 1997] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.

[LeCun et al., 1995] LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

[Ma et al., 2003] Ma, L., Tan, T., Wang, Y., and Zhang, D. (2003). Personal identification based on iris texture analysis. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1519–1533.

[Makhoul and Wolf, 1972] Makhoul, J. I. and Wolf, J. J. (1972). Linear prediction and the spectral analysis of speech. Technical report, BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA.

[Marín et al., 2009] Marín, M. R., Uribe, J. C. R., and Morales, J. C. O. (2009). Una mirada a la biometría. *Avances en Sistemas e Informática*, 6(2):29–38.

[Markel and Gray, 2013] Markel, J. D. and Gray, A. J. (2013). *Linear prediction of speech*, volume 12. Springer Science & Business Media.

[Martínez Celdrán, 1984] Martínez Celdrán, E. (1984). Fonética, teide.

[Massiris et al., 2018] Massiris, M., Delrieux, C., and Fernández Muñoz, J. Á. (2018). Detección de equipos de protección personal mediante red neuronal convolucional yolo. In *XXXIX Jornadas de Automática*, pages 1022–1029. Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura.

[Matsui and Furui, 1994] Matsui, T. and Furui, S. (1994). Comparison of text-independent speaker recognition methods using vq-distortion and discrete/continuous hmm's. *IEEE Transactions on speech and audio processing*, 2(3):456–459.

[Pérez-Ortiz, 2002] Pérez-Ortiz, J. A. (2002). Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto.

[Plumpe et al., 1999] Plumpe, M. D., Quatieri, T. F., and Reynolds, D. A. (1999). Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Transactions on Speech and Audio Processing*, 7(5):569–586.

[Rabiner and Schafer, 1978] Rabiner, L. and Schafer, R. (1978). *Digital Processing of Speech Signals*. Prentice-Hall, 1 edition.

[Rabiner and Schafer, 2010] Rabiner, L. and Schafer, R. (2010). *Theory and applications of digital speech processing*. Prentice Hall Press.

[Raschka and Mirjalili, 2017] Raschka, S. and Mirjalili, V. (2017). Python machine learning: Machine learning and deep learning with python. *Scikit-Learn, and TensorFlow. Second edition ed*.

[Reynolds, 1995] Reynolds, D. A. (1995). Speaker identification and verification using gaussian mixture speaker models. *Speech communication*, 17(1-2):91–108.

[Rose, 2002] Rose, P. (2002). *Forensic speaker identification*. cRc Press.

[Saks, 1997] Saks, M. J. (1997). Merlin and solomon: Lessons from the law's formative encounters with forensic identification science. *Hastings Lj*, 49:1069.

[Scivetti, 2007] Scivetti, A. R. (2007). La voz en la comunicación. *Revista Electrónica de Psicología Política*, 5(13).

[Serratosa, 2008] Serratosa, F. (2008). La biometría para la identificación de las personas. *Universitat Oberta de Catalunya*, pages 8–20.

[Shen et al., 2007] Shen, L., Bai, L., and Fairhurst, M. (2007). Gabor wavelets and general discriminant analysis for face identification and verification. *Image and Vision Computing*, 25(5):553–563.

[Soong et al., 1987] Soong, F. K., Rosenberg, A. E., Juang, B.-H., and Rabiner, L. R. (1987). Report: A vector quantization approach to speaker recognition. *AT&T technical journal*, 66(2):14–26.

[Téllez Ortiz et al., 2020] Téllez Ortiz, J. A. et al. (2020). Detección de vida en huellas dactilares: una revisión. B.S. thesis, Uniandes.

[Torres, 2007] Torres, B. (2007). Anatomía funcional de la voz. *Capítulo 1 del libro: Medicina del Canto. URL:* `http://www.medicinadelcant.com/cast/llibre.htm#`.

[Vallabha and Tuller, 2002] Vallabha, G. K. and Tuller, B. (2002). Systematic errors in the formant analysis of steady-state vowels. *Speech communication*, 38(1-2):141–160.

[Zhang, 2016] Zhang, Z. (2016). Mechanics of human voice production and control. *The journal of the acoustical society of america*, 140(4):2614–2635.

[Zheng and Yuan, 1988] Zheng, Y.-C. and Yuan, B.-Z. (1988). Text-dependent speaker identification using circular hidden markov models. In *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, pages 580–581. IEEE Computer Society.