

**SOFTWARE TRADUCTOR DE PALABRAS BÁSICAS DE LENGUA DE SEÑAS
COLOMBIANA**

JUAN ANDRÉS ORTIZ TORO

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN**

2024

**SOFTWARE TRADUCTOR DE PALABRAS BÁSICAS DE LENGUA DE SEÑAS
COLOMBIANA**

JUAN ANDRÉS ORTIZ TORO

Trabajo de grado para optar al título de Tecnólogo en Desarrollo de Software

Asesor

Diego Hernando Orozco Gómez

Magister en Ingeniería – Automatización Industrial

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

MEDELLÍN

2024

Agradecimientos

Agradezco a Diego Hernando Orozco por ser mi asesor en esta creación del proyecto y por su gran apoyo y paciencia hacia mi persona, haciendo posible que este proyecto saliera a la luz.

A los intérpretes de lengua de señas colombiano, Carlos Alberto Areiza y Adriana Arboleda Henao, por sus guías y explicaciones a la cultura de sordos, sin ellos no hubiera sabido lo fascinante y difícil que es la cultura de los sordos. Mas que todo a Carlos por su gran apoyo a enseñarme la inclusión hacia personas sordas y su ayuda para conseguir modelos que me permitieron realizar la toma de muestras.

A los sordos Robinson Villarreal, Iván Darío Urda, Juan Esteban y Luis Carlos Mena, por su consentimiento a la hora de tomar fotos para la creación de la base de datos y sus grandes gestos de apoyo y felicidad, por decirme en señas: “lo que estás haciendo es algo increíble, gracias por esto y esperamos que este programa pueda salir a la luz”. Muchas gracias de verdad. Prometo que esta APP va a salir para ayudar a las personas y no va a quedar en el olvido.

A mi familia, más que todo a mi papa por tolerarme y creer en mí, ahora con este trabajo te demuestro que clase de hombre soy, espero que estés orgulloso porque ya voy a seguir mi sueño, y como siempre me dices “Se el mejor en lo que te propongas”. Eso voy a hacer papá.

A Samuel Guerra García amigo y compañero de teatro, por la prestación de su voz para la creación del video del consentimiento para la captura de fotos.

Y por último agradezco a mis amigos Ángel Sierra, Juan Pablo, Samuel y doro por su apoyo emocional a seguir con este proyecto y decirme en palabras expresas “Tú puedes, eres bacano”

Contenido

	Pág.
Introducción	11
1. Planteamiento del problema.....	12
1.1 Descripción.....	12
1.2 Formulación	12
2. Justificación	13
3.1 Objetivo general	14
3.2 Objetivos específicos.....	14
4. Marco teórico.....	15
4.1 Lenguaje de señas.....	15
4.2 Red neuronal	18
4.3 Tipos de redes neuronales	20
4.4 Redes Neuronales Convolucionales	21
4.5 TensorFlow.....	22
4.6 Proyectos similares de la lengua de señas	22
5. Metodología.....	24
5.1 Tipo de proyecto.....	24
5.2 Método	24
5.3 Instrumentos de recolección de información	24
6. Resultados del proyecto.....	25
6.1 Adecuación del banco de imágenes	25
6.2 Entrenamiento de la red neuronal convolucional usando visión artificial	27
6.3 Evaluación del desempeño del modelo entrenado	30
7. Conclusiones.....	32
8. Recomendaciones	33
9. Referencias bibliográficas.....	34
10. Bibliografía	36
11. Anexos	37

Lista de figuras

	Pág.
<i>Figura 1.</i> Alfabeto de lengua de señas colombiano.....	17
<i>Figura 2.</i> Representación de una red neuronal	18
<i>Figura 3.</i> Collage de las señas. (A) adiós, (B) casa, (C) gracias, (D) hola, (E) mamá, (F) papá, (H.1 y H.2) sordo	25
<i>Figura 4.</i> Diagrama de flujo de la captura de imágenes del software	26
<i>Figura 5.</i> Carpetas con las fotos de las señas	26
<i>Figura 6.</i> Diagrama de flujo del preprocesamiento de imágenes de la red neuronal	28
<i>Figura 7.</i> Diagrama de flujo del entrenamiento de la red neuronal	28
<i>Figura 8.</i> Diagrama de flujo del software de reconocimiento	30

Lista de tablas

	Pág.
Tabla 1. <i>Total de fotogramas seleccionadas</i>	27
Tabla 2. <i>Épocas del entrenamiento</i>	28
Tabla 3. <i>Precisión de señas</i>	31

Lista de anexos

	Pág.
Anexo A. Consentimiento para captura de imágenes	37

Resumen

SOFTWARE TRADUCTOR DE PALABRAS BÁSICAS DE LENGUA DE SEÑAS COLOMBIANA

JUAN ANDRÉS ORTIZ TORO

La comunicación entre personas sordas y oyentes presenta barreras debido a la falta de conocimiento generalizado sobre la lengua de señas. Este proyecto aborda esta problemática mediante el desarrollo de un software capaz de reconocer y traducir en tiempo real siete palabras (hola, adiós, gracias, mamá, papá, sordo, casa) de la lengua de señas colombiana (LSC), usando redes neuronales convolucionales. El objetivo es facilitar la interacción entre personas sordas y oyentes, especialmente en situaciones cotidianas, promoviendo así una mayor inclusión social. La intervención realizada incluyó la recopilación de un conjunto de imágenes que capturan gestos específicos en LSC, y el entrenamiento de un modelo de inteligencia artificial en Python mediante las bibliotecas TensorFlow y Keras. A partir de pruebas de desempeño realizadas el modelo provee una precisión del 81% a 95% en el reconocimiento de las siete palabras.

Palabras clave: lengua de señas colombiana, reconocimiento de señas, red neuronal convolucional, señal, sordos

Abstract

SOFTWARE FOR TRANSLATING BASIC WORDS IN COLOMBIAN SIGN LANGUAGE

JUAN ANDRÉS ORTIZ TORO

Communication between deaf and hearing individuals significant barriers due to the limited knowledge of sign language among the general population. This project addresses this issue by developing software capable of recognizing and translating seven key words in Colombian Sign Language (LSC)—*hello, goodbye, thank you, mom, dad, deaf, and home*—in real time using convolutional neural networks. The objective is to facilitate interaction between deaf and hearing people, particularly in everyday situations, thus promoting greater social inclusion. The project involved compiling a set of images capturing specific gestures in LSC and training an artificial intelligence model in Python utilizing TensorFlow and Keras libraries. Performance tests indicate that the model achieves an accuracy rate between 81% and 95% in recognizing the selected words.

Keywords: colombian sign language, convolutional neural network, deaf, sign, sign recognition

Glosario

Época: es el número de veces de ejecución de un algoritmo; en cada ciclo todos los datos del entrenamiento pasan por la red neuronal para realizar el aprendizaje.

Fotogramas: cuadros que se obtienen a partir de videos.

Hiperparámetro: valor definido previo al entrenamiento del modelo que guía el proceso de aprendizaje, como la tasa de aprendizaje, el número de épocas, o el tamaño de lote.

Tamaño de Lote (BZ): cantidad de muestras que se procesan a la vez durante el entrenamiento de un modelo de red neuronal.

Tasa de aprendizaje (LR): hiperparámetro que determina el tamaño de los pasos que da un modelo durante el proceso de optimización.

Introducción

La lengua de señas es el medio que las personas con problemas de comunicación verbal utilizan para interactuar entre ellas haciendo uso de las manos, existiendo una dificultad para comunicarse con el resto de las personas.

El tema responde a la necesidad de herramientas que faciliten la comunicación entre personas sordas y oyentes, particularmente en un contexto donde pocos dominan la lengua de señas. Según el DANE, solo alrededor de 660 intérpretes de LSC están registrados en Colombia, de los cuales el 20% reside en Bogotá (Ministerio de Tecnologías de la Información y la Comunicaciones, 2022). Este déficit en intérpretes plantea un desafío significativo para la inclusión y comunicación efectiva, lo que motiva el desarrollo de soluciones tecnológicas en esta área.

Este proyecto de grado se centra en el desarrollo de un sistema de reconocimiento de siete palabras básicas (hola, adiós, gracias, mamá, papá, sordo, casa) en la lengua de señas colombiana (LSC) mediante el uso de redes neuronales en Python, implementadas con las bibliotecas TensorFlow y Keras.

El proyecto está estructurado en varias fases, comenzando con la recopilación de imágenes y creación de una base de datos, el entrenamiento de un modelo de red neuronal convolucional, y finalmente la evaluación de su desempeño. Para el desarrollo, se utilizaron imágenes preprocesadas que facilitan el reconocimiento de las señas específicas enunciadas anteriormente en LSC.

Los resultados obtenidos indicaron una precisión del 81% a 95% en el reconocimiento de las siete palabras que servirán como base para futuros desarrollos que amplíen el vocabulario de señas reconocibles por el sistema, contribuyendo así a la mejora en la accesibilidad y comunicación inclusiva.

1. Planteamiento del problema

1.1 Descripción

La comunicación es un factor fundamental para los seres humanos en todas las interacciones sociales donde el lenguaje es el ápice más esencial; las personas sin discapacidades auditivas se comunican de forma verbal sin ningún problema y los sordos, que saben lenguaje de señas, se les es fácil comunicarse entre ellos mismos. El problema repercute en que los sordos traten de darse a entender a personas que no padecen de este tipo de discapacidad y no saben interpretar el lenguaje de señas. Según el Departamento Nacional de Estadística DANE aproximadamente 660 personas en todo el territorio nacional son intérpretes de lenguaje de señas, de los cuales, el 20% se encuentra en Bogotá (Ministerio de Tecnologías de la Información y la Comunicaciones, 2022). Es por ello que se debe buscar soluciones con mecanismos tecnológicos que permitan facilitar esa comunicación efectiva entre las dos partes y esta es la razón de realizar este proyecto de grado.

1.2 Formulación

¿Será posible que una red neuronal implementada en Python tenga un desempeño aceptable en el reconocimiento de siete palabras básicas de la lengua de señas colombiana?

2. Justificación

Esta investigación se realiza para obtener el grado de tecnólogo en Desarrollo de Software, ayudando a las personas sordas a comunicarse con personas que no tienen esa discapacidad. Este trabajo estará enfocado, principalmente, en la inclusión de personas sordas y facilitación de su comunicación entre oyentes y no oyentes. Las personas sordas se les presenta obstáculos al comunicarse con otras personas debido a que les dificulta entender el idioma nativo del país, en este caso el español. Debido a esto, las personas sordas de Colombia tienen el español como segundo idioma y el primero es el lenguaje de señas que se inventan con su familia (lenguaje de señas hogareño) hasta que se le educa a entender el lenguaje de señas y el significado de la palabra. Sin embargo, aun así, les es difícil expresarse a otras personas debido a que la mayoría de las personas sin discapacidad no entienden el lenguaje de señas y una parte de ellos no sabe escribir una oración completa. Otro problema es que el lenguaje de señas no es universal, eso quiere decir que en la mayoría de las partes del mundo el significado de las señas varía mucho, incluso en Colombia hay variaciones dialécticas del Lenguaje de Señas Colombiana (LSC) según la región o comunidad específica (Federación Nacional de Sordos de Colombia), dificultando más la comunicación exitosa de un sordo.

La comunidad científica se ha interesado en realizar investigaciones que tienen como propósito el mejoramiento de la calidad de vida de las personas sordas; la revisión de la literatura evidencia que las investigaciones están empleando diferentes dispositivos tecnológicos para resolver este problema. Entre los dispositivos disponibles se encuentran cámaras de vídeo e imágenes, basadas en diversos algoritmos de visión computacional y de extracción de características de las señas filmadas (Galvis Serrano et al., 2019).

Este proyecto busca facilitar la comunicación entre personas sordas y personas sin discapacidad mediante la implementación de un algoritmo que identifique siete palabras básicas de la lengua de señas colombiana, utilizando Inteligencia Artificial y reconocimiento de imágenes aplicados a la visión computacional. Se desarrollará un prototipo que reconozca señas individuales y traduzca en tiempo real.

3. Objetivos

3.1 Objetivo general

Desarrollar un software básico que reconozca siete palabras básicas de la lengua de señas colombiana utilizando una red neuronal codificada en Python.

3.2 Objetivos específicos

Adecuar un banco de imágenes de personas sordas realizando un gesto con significado en la lengua de señas colombiano.

Llevar a cabo el entrenamiento de una red neuronal implementada en Python haciendo uso de la base de datos creada.

Evaluar el desempeño del modelo entrenado con el fin de determinar el nivel de precisión en el reconocimiento de las señas empleadas.

4. Marco teórico

4.1 Lenguaje de señas

El lenguaje es el gran instrumento de comunicación que dispone la humanidad, en la antigüedad se creía que el lenguaje estaba limitado a lo sonoro y lo auditivo, por tanto, se pensaba que las personas sordas eran incapaces de aprender y desarrollar una lengua. Sin embargo, con el tiempo se descubrió que las lenguas naturales no están limitadas al plano de lo sonoro, sino que se extienden a la comunicación gesto-espacial. Es importante entender que las lenguas de señas son lenguas naturales, ya que cuentan con una sintaxis propia, tienen variantes dialectales y se desarrollan de forma.

En la mayor parte de las civilizaciones antiguas se veía a las personas sordas —que se creía también eran mudas— como seres enfermos e incapaces de razonar; eran excluidos de la sociedad y en muchas civilizaciones eran arrojados a los ríos o a los precipicios. Incluso el gran filósofo Aristóteles afirmaba que la sordera era sinónimo de falta de inteligencia. Esta idea se mantuvo vigente durante siglos. En la Edad Media la Iglesia ejercía mucho poder sobre la cultura y lo que se consideraba como verdad, difundía la idea de que los sordos estaban siendo castigados por Dios, así que en muchas ocasiones eran enviados a manicomios.

Fue hasta finales del siglo XVI que estas percepciones comenzaron a cambiar. Gerolamo Cardano, un médico italiano, fue una de las primeras personas en decir abiertamente que los sordos tenían la capacidad de comunicarse. Tiempo después, Pedro Ponce de León —monje benedictino— fue el primer educador de las personas sordas a través del alfabeto manual. Un siglo más tarde, Juan Pablo Bonet publicó *Reducción de las letras y arte para enseñar a hablar a los mudos*, en este libro se retoman las ideas de Pedro Ponce de León, además de contener un amplio estudio sobre la gramática y fonética de muchas palabras. En 1771 se abrió la primera escuela exclusiva para estudiantes sordos; ésta se atribuye a Charles-Michel de l'Épé, religioso francés quien después de estar a cargo de dos niñas sordas, decidió fundar una escuela en París donde se enseñaba no sólo a comunicarse, sino a aprender otras materias.

Para el siglo XIX ya era aceptada la lengua de señas en el mundo, pero en realidad todavía quedaba un largo camino por recorrer. En el Congreso de Milán de 1880 —en el que se reunieron diversos profesores de personas sordas— se tomó la decisión de educar solamente a través del habla, lo cual representó un obstáculo para la educación de estas personas; sin embargo, esto no impidió que en las comunidades se comunicaran a través de señas en su vida cotidiana.

Hoy, gracias a Marroquín, se descubre que, como mínimo desde principios del siglo XX hasta aquel año de 1975, el nombre del lenguaje actualmente en discusión se llamaba, no de “señas”, un término puesto en boga en España desde el siglo X, pero ya perdido en los principios del siglo XIX, momento en que los maestros españoles empezaron a denominarlo de “signos”, por influencia lingüística de la escuela francesa abierta por el abate L’Epée a mediados del siglo anterior, sino “mímico” o de “gestos”.

Pero lo más curioso es que Félix-Jesús Pinedo, el sucesor de Marroquín en 1979, diez años más tarde, concretamente en 1989, seguía llamándolo todavía “lenguaje gestual”. Desde el siglo X hasta principios del XIX, el lenguaje de los sordos en España se llamaba de “señas”, sin más. Durante casi todo el XIX, y a causa de influencia de la escuela francesa entre los maestros españoles, pasó llamarse de “signos”, un término hay que decir que muy “afrancesado”, y ya en el siglo XX, desde el principio hasta como mínimo el año 1992, volvió de nuevo a cambiar de nombre al denominarse “lenguaje mímico” o “gestual”, y ahora en la actualidad, en el siglo XXI, ha vuelto otra vez a cambiar a “signos”. Cuestiones semánticas, al fin y al cabo (Gascón Ricao, 2004).

La lengua de señas no es un lenguaje universal. La Federación Mundial de Personas Sordas señala que en el mundo existen más de 300 lenguas de señas, cifra complementaria a las casi siete mil lenguas orales existentes, entre ellas por ejemplo el idioma inglés o español. El deletreo con los dedos es el proceso de deletrear palabras usando formas de manos que corresponden a las letras de la palabra. Un conjunto de estas formas usadas para deletrear palabras se conoce como un alfabeto dactilológico o “alfabeto manual”. Hay muchos alfabetos manuales diferentes en todo el mundo (Rosero Titillo, 2019).

Si se quisiera definirla con mayor exactitud, se podría decir que la lengua de señas es un baile de la comunicación que involucra todo el cuerpo. No se trata sólo de componer símbolos con las manos, sino de vivirlos y llevarlos, estéticamente, al otro. Por supuesto, son las manos las que dictan el diálogo, pero es el movimiento de la cabeza el que pone el énfasis, el ‘tonito’, por decirlo de alguna forma. Y por lo general son las expresiones del rostro las que acentúan las emociones y los sentimientos.

No solo eso. Todo el cuerpo, con su postura en el espacio o con un movimiento hacia adelante o hacia atrás, puede determinar, entre otras, si se está hablando en pasado, presente o futuro. Y a través de toda esa estética, hay una comunicación eficaz e incluyente de la que las personas sordas se sienten orgullosas. Deletrear apropiadamente es una competencia lingüística muy básica y útil al momento de comunicarse en lengua de señas (Quintero Cerón, 2018).



Figura 1. Alfabeto de lengua de señas colombiano

Fuente: extraído de <https://www.uniminutoradio.com.co/unificacion-del-lenguaje-de-senas-una-oportunidad-para-comunicarnos-mas/>

A partir de la década de los ochenta, se generaron en Colombia hechos de importancia que condujeron paulatinamente a las actuales transformaciones socioeducativas para las personas no oyentes. Los cambios se dieron específicamente en las políticas oficiales, en las dinámicas de organización de la comunidad sorda y también en las instituciones educativas para sordos. Desde el Ministerio de Educación Nacional se planteó el desarrollo de los programas del currículo oficial, con las adecuaciones necesarias y el desarrollo de habilidades de comunicación, sin especificar la enseñanza particular de una lengua, que posibilitaron la integración social, educativa y laboral de las personas sordas (Patiño Giraldo, 2010).

4.2 Red neuronal

Las redes neuronales son modelos simples del funcionamiento del sistema nervioso. Las unidades básicas son las neuronas, que generalmente se organizan en capas. Las unidades de procesamiento también se organizan en capas. Hay tres partes normalmente en una red neuronal: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida, con una unidad o unidades que representa el campo o los campos de destino. Las unidades se conectan con fuerzas de conexión variables (o ponderaciones). Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente, al final, se envía un resultado desde la capa de salida.

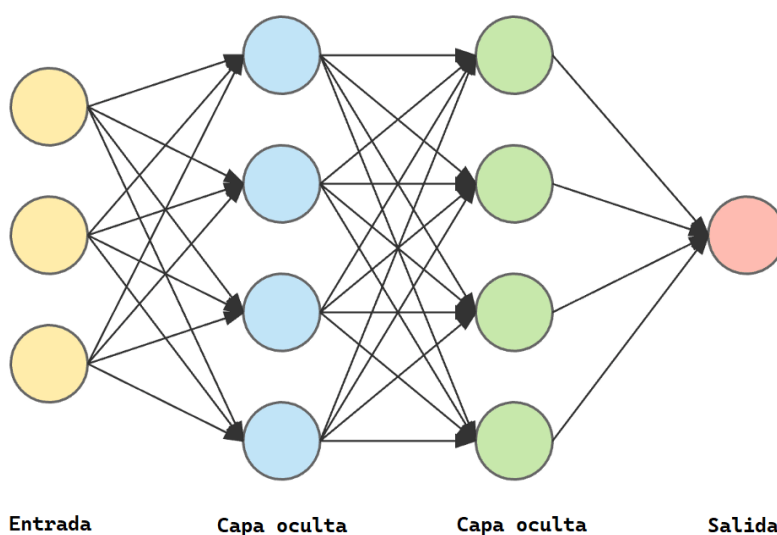


Figura 2. Representación de una red neuronal

Fuente: extraído de <https://openwebinars.net/blog/que-son-las-redes-neuronales-y-sus-aplicaciones/>

La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realiza una predicción incorrecta. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones hasta haber alcanzado uno o varios criterios de parada. Al principio, todas las ponderaciones son aleatorias y las respuestas que resultan de la red son, posiblemente, disparatadas. La red aprende a través del entrenamiento. Continuamente se presentan a la red ejemplos para los que se conoce el resultado, y las respuestas que proporciona se comparan con los resultados conocidos. La información procedente de esta comparación se pasa hacia atrás a través de la red, cambiando las ponderaciones gradualmente. A medida que progresa el entrenamiento, la red se va haciendo cada vez más precisa en la replicación de resultados conocidos. Una vez entrenada, la red se puede aplicar a casos futuros en los que se desconoce el resultado.

Las señales de entrada, la información que recibe la red neuronal artificial, son variables independientes. Los n-valores de entrada son multiplicados por sus respectivos pesos, es decir en la sinopsis el vector entrada es multiplicado por el vector peso, dando como resultado una combinación lineal de las entradas y los pesos, algo que se denomina función de ponderación.

$$x * w^t = (x_1, x_2 \dots x_n) * (w_n) = \sum_{i=1} x_i * w_i$$

Por último, dicho resultado se propaga a la salida. Dicho valor puede ser la nueva entrada de una neurona, formando así las redes neuronales, o bien puede ser el resultado final. Las respuestas obtenidas pueden ser una variable continua como el precio de un objeto, una respuesta binaria (0,1) (Sí, No) si una persona padece algún tipo de enfermedad o no, o puede ser una respuesta categórica que sirva para clasificar, por ejemplo.

El alcance de las funciones de las redes neuronales es muy amplio, debido a su funcionamiento, son capaces de aproximar cualquier función existente con el suficiente entrenamiento. Principalmente las redes neuronales son utilizadas para tareas de predicción y clasificación. Su rango de actuación es amplio y de gran utilidad hoy en día, no solo se utilizan para aplicaciones de Industria 4.0 (reconocimiento de piezas y defectos que no han sido

introducidos previamente, por ejemplo, si no que son utilizados en otras áreas como la economía, en la que pueden ayudar a predecir cuánto van a variar los precios a lo largo de los años, o incluso en medicina donde son de gran ayuda para diagnosticar diversos problemas de salud.

Las redes neuronales se han convertido en una pieza clave para el desarrollo de la Inteligencia Artificial, es uno de los principales campos de investigación y el que más está evolucionando con el tiempo, ofreciendo cada vez soluciones más complejas y eficientes (Qué son las redes neuronales y sus funciones, 2019).

4.3 Tipos de redes neuronales

El perceptrón. El perceptrón es la forma más simple de una red neuronal usada para la clasificación de un tipo especial de patrones, los linealmente separables (es decir, patrones que se encuentran a ambos lados de un hiperplano). Básicamente, consiste en una neurona con pesos sinápticos y umbral ajustables.

El algoritmo usado para ajustar los parámetros libres de esta red neuronal apareció por primera vez en un procedimiento de aprendizaje desarrollado por Rosenblatt (1958) para su modelo de perceptrón del cerebro. En realidad, Rosenblatt demostró que, si los patrones usados para entrenar el perceptrón son sacados de dos clases linealmente separables, entonces el algoritmo del perceptrón converge y toma como superficie de decisión un hiperplano entre estas dos clases. La prueba de convergencia del algoritmo es conocida como el teorema de convergencia del perceptrón.

Adaline. La diferencia con el Perceptrón es la manera de utilizar la salida en la regla de aprendizaje, el Perceptrón utiliza la salida de la función umbral (binaria) para el aprendizaje, sólo se tiene en cuenta si se ha equivocado o no. En Adaline se utiliza directamente la salida de la red (real) teniendo en cuenta cuánto se ha equivocado. Se utiliza la diferencia entre el valor real esperado y la salida producida de la red.

Para un patrón de entrada x^p , se tendrá en cuenta el error producido $d^p - y^p$, siendo d^p la salida deseada e y^p . El objetivo es obtener una red tal que $y^p \approx d^p$ para todos los patrones p . Será imposible conseguir una salida exacta porque y es una función lineal, pero se minimizará el error cometido para todos los patrones de entrenamiento; hay que elegir una medida de dicho error, por ejemplo, el error cuadrático, la regla de aprendizaje es la regla delta. Se hará mediante un proceso iterativo donde se van presentando los patrones uno a uno y se van modificando los pesos y el umbral de la red mediante la regla del descenso del gradiente.

$$w_j(t + 1) = w_j(t) + \Delta_p w$$

La idea es realizar un cambio en cada peso proporcional a la derivada del error, medida en el patrón actual, respecto del peso:

$$\Delta P w_j = \gamma \frac{\partial E^p}{\partial w_j}$$

Siendo γ la tasa o razón de aprendizaje (Galván & Valls, 2017).

4.4 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son una extensión de las redes neuronales tradicionales, diseñadas para procesar datos con estructura en forma de cuadrícula, como imágenes. Inspiradas en el procesamiento visual del cerebro, las CNN se han convertido en una herramienta esencial en el aprendizaje profundo, especialmente en tareas de visión por computadora.

En una CNN, las neuronas están organizadas en capas especializadas que permiten identificar patrones locales en los datos. Las capas convolucionales aplican filtros a las entradas para crear mapas de características, que detectan elementos como bordes y texturas. Las capas de pooling reducen la dimensionalidad de estos mapas, seleccionando características esenciales y mejorando la eficiencia del modelo. La red aprende ajustando los pesos de los filtros mediante un proceso de retropropagación, donde el error entre las predicciones y los resultados reales se minimiza gradualmente (Deep Learning. MIT Press, 2016).

4.5 TensorFlow

TensorFlow, una biblioteca de código abierto, facilita la implementación de CNN a través de su API Keras, que permite definir y entrenar modelos de manera eficiente. Un ejemplo básico de implementación de una CNN en TensorFlow incluye la definición de capas convolucionales y de pooling, seguidas de capas completamente conectadas que realizan la tarea final de predicción. Este proceso permite a la red aprender de manera iterativa, mejorando su precisión en tareas de clasificación de imágenes, detección de objetos, entre otras aplicaciones clave en la inteligencia artificial.

TensorFlow está basado en un sistema de grafos de flujo de datos. Cada nodo en el grafo representa una operación matemática, y los bordes entre los nodos representan los flujos de datos. Esta estructura permite a TensorFlow ejecutar eficientemente cálculos en una variedad de dispositivos, incluidos CPU, GPU y TPU (TensorFlow (n.d.)).

4.6 Proyectos similares de la lengua de señas

En el artículo Clasificación de Gestos de la Lengua de Señas Colombiana a partir del Análisis de Señales Electromiográficas utilizando Redes Neuronales Artificiales, para clasificarlos se usaron 27 gestos del alfabeto de señas colombiano mediante las señales electromiográficas a partir de un clasificador de redes neuronales, para el estudio obtuvieron los registros de las señales electromiográficas de 13 personas con discapacidad auditiva; el clasificador presentó una de precisión de 88,4% (Galvis Serrano et al., 2019).

Otro ejemplo es el sistema de reconocimiento de lenguaje de señas usando una red neuronal para la implementación de un hardware digital, consiste en reconocer patrones de imágenes que muestra símbolos específicos de este lenguaje; las imágenes se procesan para adaptarlas y mejorar el desempeño, reduciendo y eliminando ruido y detectando el borde; en este caso se evaluaba utilizando los señas sin incluir ningún tipo de movimiento en su representación (Vargas & Barba, 2011).

Además, se realizó una investigación respecto al reconocimiento de lenguaje de señas por medio de una red neuronal, donde se usó una imagen digital que se ingresó a un sistema de entrenamiento y reconocimiento que aprende por medio de una red neuronal multicapa que utiliza un algoritmo de propagación inversa; al final se obtuvo una precisión del 84% como resultado exitoso (Dogic & Karli, 2014).

5. Metodología

5.1 Tipo de proyecto

Este proyecto está catalogado como aplicativo, ya que se pretende utilizar conocimientos teóricos adquiridos en la tecnología para la implementación de un proceso de reconocimiento y entrenamiento de red neuronal a partir de imágenes de señas en el lenguaje de señas colombiano.

5.2 Método

Para lograr este proyecto y su propósito se implementó un método basado en tres fases nombradas: recopilación de imágenes y creación de base de datos, entrenamiento de la red neuronal y prueba con análisis y resultados.

5.3 Instrumentos de recolección de información

5.4.1 Fuentes primarias. Para el desarrollo de este proyecto se consultó con expertos en el tema de lenguaje de señas como Adriana Arboleda Henao quien es intérprete de dicho lenguaje. También se consultó información de varios artículos sobre redes neuronales.

5.4.2 Fuentes secundarias. Se tomó en cuenta foros sobre el lenguaje de señas y redes neuronales.

6. Resultados del proyecto

6.1 Adecuación del banco de imágenes

Con el fin de llevar a cabo el entrenamiento de la red neuronal, fue necesario construir previamente un banco de imágenes que sirviera como base para este proceso. La adecuación de este conjunto de imágenes se llevó a cabo a partir de la toma de fotogramas de siete palabras básicas de la lengua de señas colombiana. Se utilizó la librería MediaPipe desarrollada bajo el lenguaje de programación Python que permitió la toma de frames en la adquisición de imágenes de las manos de las personas al momento de realizar los movimientos correspondientes a cada palabra. En la figura 3 se muestra un collage de las señas capturadas.



Figura 3. Collage de las señas. (A) adiós, (B) casa, (C) gracias, (D) hola, (E) mamá, (F) papá, (H.1 y H.2) sordo

Fuente: diseño propio

Con el objeto de realizar la creación del banco, se contactó a Carlos Alberto Areiza Montoya un experto intérprete en la lengua de señas colombiana y a Iván Darío Urda, Robinson Villareal, Juan Esteban y Luis Carlos Mena, cuatro personas sordas a las cuales se les realizó la toma de frames correspondientes a las imágenes de señas. Todos autorizaron el consentimiento de la toma de imágenes el cual se encuentra en el anexo A. Para la obtención de imágenes de las señas ya antes mencionadas, se usó un salón de la Institución Universitaria Pascual Bravo que brindara condiciones adecuadas en cuanto a iluminación y ubicación de la cámara. Los cuatro participantes fueron informados de la fase de preparación y sus directrices a seguir. A cada persona se le realizó la toma de señas de manera continua, logrando adquirir 300 fotogramas por cada seña.

Una vez tomados los frames, el algoritmo recorta automáticamente cada imagen con el fin que únicamente en la imagen resultante queden las manos de las personas y así evitar un problema por el ruido para el entrenamiento de la red neuronal. En la figura 4 se detalla el diagrama de flujo de la adquisición de las imágenes y en la figura 5 se muestran las carpetas con las fotos de las señas.

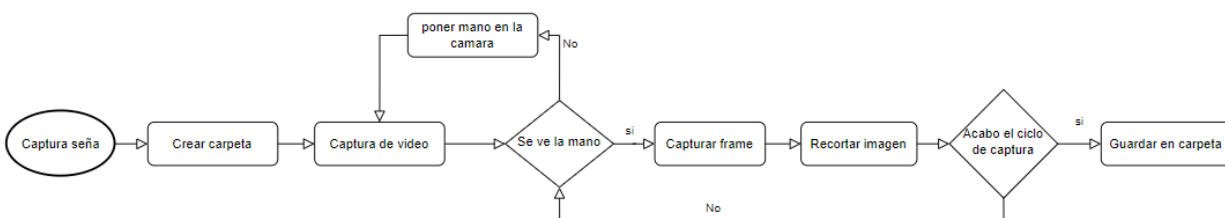


Figura 4. Diagrama de flujo de la captura de imágenes del software

Fuente: diseño propio

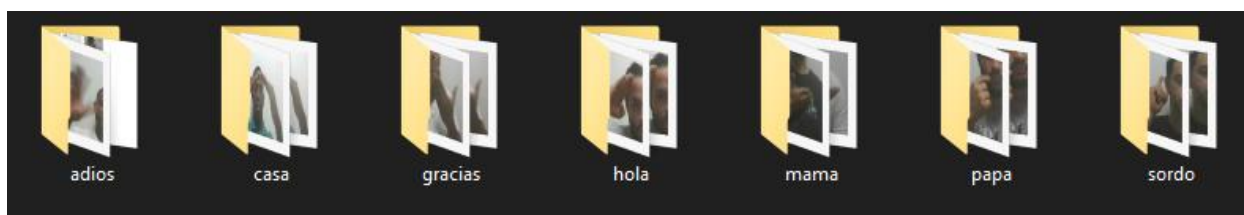


Figura 5. Carpetas con las fotos de las señas

Fuente: diseño propio

Posteriormente a la captura de las señales, se realizó una revisión de las imágenes para identificar cuales no servirán debido al ruido que presentan y por ende ser eliminadas manualmente de las carpetas para no generar ningún inconveniente al entrenar la red. La Tabla 1 detalla el número de imágenes por cada señal.

Tabla 1.

Total de fotogramas seleccionadas

Frase	Cantidad de fotogramas
Adiós	487
Casa	392
Gracias	261
Hola	549
Mamá	506
Papá	498
Sordo	469
Total Fotogramas	3162

Fuente: diseño propio

6.2 Entrenamiento de la red neuronal convolucional usando visión artificial

Se creó un algoritmo de entrenamiento de red neuronal convolucional usando el framework Tensorflow, a diferencia de otros programas de entrenamiento que se necesita tener diferentes clases para entrenar cada señal. Al principio se generaron dos errores que hacía imposible continuar con la investigación y el debido entrenamiento de la red, se estableció que los problemas se presentaban porque la carpeta donde se guardan las librerías de Python tenía un carácter especial que no pertenecían al abecedario inglés, y el otro error era la incompatibilidad de las versiones de las librerías; el programa funciona con las siguientes versiones: Python 3.7.6, Keras:2.3.1, Tensorflow:1.15.5; Mediapipe:0.10.9; Numpy:1.23.0; ejecutando el programa en un entorno virtual.

Este algoritmo preprocesa las imágenes en un mismo código logrando mejorar el entrenamiento. Se entiende por preprocesar como la función de transformar la imagen para presentar adecuadamente las características de esta al entrenador, según los parámetros que se especifiquen. En la figura 6 se muestra el diagrama de flujo del preprocesamiento de las imágenes.

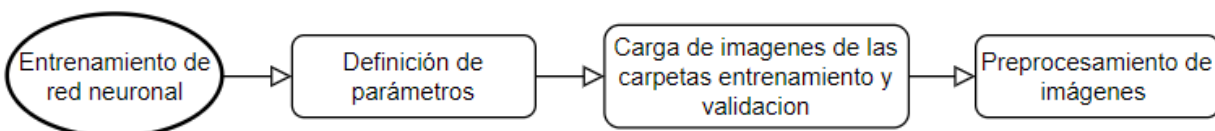


Figura 6. Diagrama de flujo del preprocesamiento de imágenes de la red neuronal
Fuente: diseño propio

Como se mencionó previamente, se adquirieron 300 frames por cada señal para las cuatro personas, esto permitió la obtención de 8400 frames en total. De esta cantidad, el total paso a 3162 frames como se muestra en la tabla 1, las cuales fueron utilizadas en el proceso de entrenamiento de la red y las restantes en la validación, eliminando las del ruido. En la figura 7 se indica el diagrama de flujo del entrenamiento de la red.

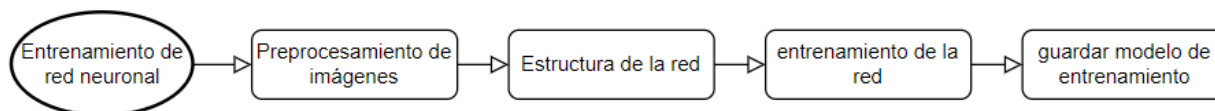


Figura 7. Diagrama de flujo del entrenamiento de la red neuronal
Fuente: diseño propio

El entrenamiento de la red neuronal requiere ajustar tres hiperparámetros correspondientes a Tasa de aprendizaje (LR), Tamaño de Lote (BZ) y Épocas (E). Estos se adecuaron así:

$$LR = 0.0005$$

$$BZ = 93$$

$$E = 50$$

En la Tabla 2 se muestra el resultado de la pérdida, la precisión, la validación de la pérdida y validación de precisión por cada Época en el proceso de entrenamiento de la red neuronal.

Tabla 2.

Épocas del entrenamiento

Epochs	Perdida	Precisión	Validación Perdida	Validación Precisión
1	2.2883	0.8403	29.2745	0.1240
2	0.1558	0.9703	34.5484	0.1240
3	0.1700	0.9738	28.9218	0.2615
4	0.0449	0.9861	28.6636	0.1578
5	0.0430	0.9886	24.9390	0.1575
6	0.0495	0.9892	17.9511	0.1689

Tabla 2.
(Continuación)

Epochs	Perdida	Precisión	Validación Perdida	Validación Precisión
7	0.0489	0.9880	17.1616	0.1600
8	0.0404	0.9896	22.8233	0.1875
9	0.0447	0.9908	29.8782	0.1600
10	0.0190	0.9943	22.1932	0.1600
11	0.0304	0.9934	18.6291	0.2116
12	0.0350	0.9930	19.3518	0.2502
13	0.0305	0.9927	14.3289	0.2951
14	0.0224	0.9956	10.0213	0.3950
15	0.0123	0.9953	6.6273	0.5702
16	0.0268	0.9937	4.0316	0.7242
17	0.0103	0.9968	4.7579	0.6739
18	0.0233	0.9943	5.2220	0.6796
19	0.0427	0.9911	4.6099	0.7211
20	0.0340	0.9915	5.1402	0.6682
21	0.0158	0.9956	4.7884	0.7135
22	0.0198	0.9959	6.3068	0.6385
23	0.0349	0.9949	8.9701	0.5566
24	0.0226	0.9968	4.7578	0.6543
25	0.0274	0.9959	5.7785	0.5933
26	0.0428	0.9943	3.8319	0.6822
27	0.0407	0.9943	5.0443	0.6626
28	0.0135	0.9968	6.5560	0.6031
29	0.0226	0.9962	8.2569	0.5585
30	0.0288	0.9940	11.8916	0.5452
31	0.0160	0.9972	5.8377	0.6240
32	0.0152	0.9978	5.9426	0.6006
33	0.0187	0.9972	10.4442	0.3978
34	0.0378	0.9937	8.0949	0.5278
35	0.0566	0.9956	16.1780	0.3694
36	0.0285	0.9937	5.6164	0.7394
37	0.0361	0.9940	5.7613	0.7650
38	0.0532	0.9934	9.1671	0.6603
39	0.0594	0.9924	22.2842	0.2331
40	0.0275	0.9949	10.2342	0.5117
41	0.0233	0.9972	12.5206	0.6480
42	0.0165	0.9984	9.4975	0.6493
43	0.0050	0.9987	11.1954	0.6518
44	0.0191	0.9975	7.3613	0.6667
45	0.0163	0.9972	7.3275	0.7201
46	0.0185	0.9965	7.5137	0.6256
47	0.0139	0.9978	6.3105	0.6075
48	0.0208	0.9956	8.7591	0.7245
49	0.0085	0.9984	14.7099	0.5914
50	0.0381	0.9962	12.1908	0.6417

Fuente: diseño propio

Se puede observar en la tabla 2 como incrementa la precisión de la red hasta llegar a un punto donde el aumento es significativo y se empieza a estabilizar, y como cada vez la pérdida va disminuyendo hasta que se estabiliza, el hecho de que los valores de la precisión de la red se estabilizaran quiere decir, que el aumento del número de Epoch no aumenta indefinidamente la precisión y llega a cierto punto donde ya se estabiliza los valores.

Al finalizar el entrenamiento el software suministró dos archivos con los modelos entrenados llamados *Modeloabc.h5* y *pesoabc.h5*.

6.3 Evaluación del desempeño del modelo entrenado

Se realizó un algoritmo que importa el modelo de entrenamiento de la red neuronal con el objetivo de hacer las pruebas y verificaciones de las señas, y validar el reconocimiento de éstas. A veces generar error por las dimensiones de las imágenes que se especificó en el entrenamiento. El proceso empleado que se uso fue recrear las señas a tiempo real mostrando que señas es. En la figura 8 se detalla el diagrama de flujo del proceso de reconocimiento.

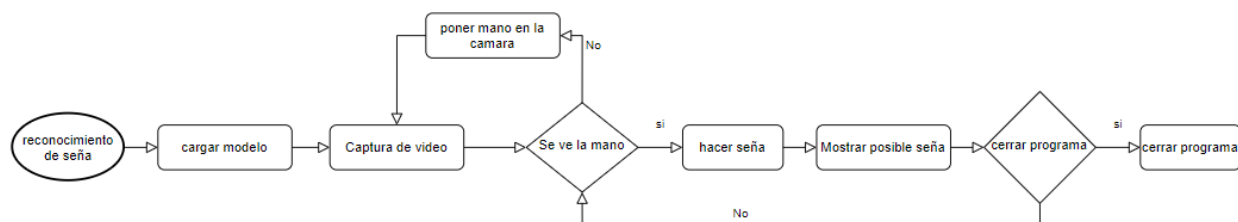


Figura 8. Diagrama de flujo del software de reconocimiento

Fuente: diseño propio

Para realizar la validación se optó por hacer un procedimiento práctico y así obtener la precisión de las señales correspondiente; se determinó por cada frase recrearla 100 veces frente a la cámara para establecer la precisión, dividiendo los aciertos sobre el total. El resultado provee una precisión en el reconocimiento de las señas, tal como se observa en la tabla 3.

Tabla 3.
Precisión de señas

Seña	Precisión
Adiós	0.81
Casa	0.37
Gracias	0.34
Hola	0.70
Mamá	0.95
Papá	0.63
Sordo	0.55

Fuente: diseño propio

Tras la muestra de los resultados de la tabla 3, se da entender que, las señas: mamá, adiós, hola y papá tienen una mayor facilidad de ser reconocidas por el modelo, debido a la sencillez de la seña y su poca movilidad en el entorno de la cámara, ya que no era un movimiento brusco y era una sola mano; en cambio las señas: casa, gracias y sordo, se obtiene una precisión baja debido múltiples factores, en el caso de casa el hecho de usar dos manos se le dificultaba en ciertas ocasiones la detección de las manos y debido a eso el programa detectaba otra seña, igualmente sucedió con la seña de gracias debido al uso de las dos manos, y el movimiento brusco de la seña al modelo se le dificultaba detectarlo; por último la seña de sordos hacia un movimiento muy brusco lo que generaba dificultades en su reconocimiento.

7. Conclusiones

Tras la realización de los algoritmos, el entrenamiento de la red neuronal, la entrega de resultados y con su debido análisis se puede concluir que:

Con el trabajo se demostró que se necesitan versiones específicas del lenguaje de programación Python y de las bibliotecas TensorFlow, Keras, Mediapipe y Numpy para el buen funcionamiento de los algoritmos. Debido a esto se deben actualizar a fin de tener nuevas y mejores funcionalidades en pro de un adecuado desempeño del software.

La creación del banco de imágenes para la red neuronal hizo uso de siete señas, sin embargo, la versatilidad que tiene el algoritmo hace posible que se puedan almacenar N señas, incluso frases u oraciones largas.

El número de Epochs (Épocas) afecta hasta cierto punto la precisión, pero llegará a un límite donde los valores se normalizarán y no aumentarán más la precisión, dando a entender que el número de Epochs tendrá un límite donde ya se normalicen los valores y no aumenten significativamente.

En el proceso de prueba de la red neuronal se obtuvieron diferentes porcentajes de precisión, da a entender que ciertas señas tienen dificultad de ser detectadas, debido a la similitud de otras señas, es decir, la complejidad que tiene el sistema de reconocer dos manos y la complejidad de los movimientos de las señas, haciendo difícil al sistema predecir la seña que se le mostraba.

8. Recomendaciones

Se puede mejorar el entrenamiento de la red haciendo que pueda identificar los puntos de las manos de los fotogramas usando la librería Mediapipe para identificarlos y poder enseñarlos.

Se puede automatizar la selección y eliminación de fotogramas borrosos ya que el proceso se realizó manualmente.

A futuro es posible probar diferentes modelos de redes neuronales y un banco de imágenes más denso.

Así mismo, se pueden usar datos biométricos como la voz para la identificación e interpretación de señas.

9. Referencias bibliográficas

- Atria. (2019). *Qué son las redes neuronales y sus funciones*
<https://atriainnovation.com/blog/que-son-las-redes-neuronales-y-sus-funciones/>
- Dogic, S., & Karli, G. (2014). *Sign Language Recognition using Neural Networks*. *Tem Journal*, 3, 294.
- Federación Nacional de Sordos de Colombia. (2023). “Colombia tiene porcentaje bajo en la implementación de los derechos de las personas sordas”, *InSOR*.
<https://www.infobae.com/colombia/2023/09/24/colombia-tiene-porcentaje-bajo-en-la-implementacion-de-los-derechos-de-las-personas-sordas-insor/>
- Galván, I., & Valls, J. (2017). *Primeros modelos computacionales*.
- Galvis Serrano, E., Sanchez Galvis, I., Florez, N., & Zabala Vargar, S. (2019). Clasificación de Gestos de la Lengua de Señas Colombiana a partir del Análisis de Señales Electromiográficas utilizando Redes Neuronales Artificiales. *La Serena Mar*, 30(0), 10.
- Gascón Ricao, A. (2004). *Historia de la lengua de señas*. Cultura Sorda. <https://cultura-sorda.org/historia-de-la-lengua-de-signos/>
- Ian, G, Yoshua, B & Aaron, C (2016) *Deep learning MIT*
- Ministerio de Tecnologías de la Información y la Comunicaciones. (2022). *Intérpretes de lengua de señas, conectando mundos*. <https://centroderelevo.gov.co/632/w3-article-239345.html>
- Patiño Giraldo, E. (2010). La lengua de señas colombiana como mediadora en el proceso de conceptualización de nociones relacionadas con las ciencias sociales en niños y niñas no oyentes (p. 39). <https://biblioteca.clacso.edu.ar/Colombia/alianza-cinde-umz/20130225050457/tpatino.pdf>

Quintero Cerón, R. (2018). *Estuvo prohibida y cambia por país: curiosidades de la lengua de señas*. <https://www.eltiempo.com/vida/educacion/curiosidades-de-la-lengua-de-senas-185256>

Rosero Titillo, E. (2019). *¿Son todas las lenguas de señas iguales? ¿Por Qué No Existe Una Lengua de Señas Universal?* <https://www.turismoaccesibleec.org/2019/10/16/comparativa-iberoamericana-de-las-lenguas-de-senas-signos/>

Vargas, L., & Barba, L. (2011). *Sign Language Recognition System using Neural Network for Digital Hardware Implementation. 0, 1.*

10. Bibliografía

MediaPipe (2024) MediaPipe. Recuperado el 28 de octubre de 2024, de <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419>

OpenCV. (2024). OpenCV. Recuperado el 28 de octubre de 2024, de <https://opencv.org/>

TensorFlow. (s.f.). TensorFlow. Recuperado el 28 de octubre de 2024, de <https://www.tensorflow.org/?hl=es-419>

11. Anexos

Anexo A. Consentimiento para captura de imágenes

Video creado: <https://youtu.be/ELqbWFXanOc>

Imágenes de muestra:



Personas involucradas:

Ivan Darío Urda Acevedo

Robison Villareal

Juan Esteban

Luis Carlos Mena Jave

Autorización

SI

NO

Autorización

4 respuestas

