ISSN: 1692-7257 - Volumen 2 – Número 46 - 2025

RCTA Revista Colombiana de Tecnologías de Avanzada UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v2i46.4105

Traducción de lenguaje de signos a texto mediante python con redes neuronales LSTM

Sign language to text translation using Python with LSTM neural networks

Ing. Luis Duván Torrado Mora [1], PhD. César Alberto Collazos [2], PhD. Dewar Rico Bautista [1]

[1] ***Universidad Francisco de Paula Santander,*** *Ocaña, Norte de Santander, Colombia.*
[2] ***Universidad del Cauca,*** *Popayán, Cauca, Colombia.*

Correspondencia: *dwricob@ufpso.edu.co*

Recibido: *24 abril 2025.* ***Revisado:*** *16 junio 2025.* ***Aceptado:*** *03 julio 2025.* ***Publicado:*** *25 julio 2025.*

Cómo citar: *L. D. Torrado Mora, C. Alberto Collazos, y D. Rico Bautista, «Traducción de lenguaje de signos a texto mediante python con redes neuronales LSTM», RCTA, vol. 2, n.º 46, pp. 150–159, jul. 2025.*
Recuperado de https://ojs.unipamplona.edu.co/index.php/rcta/article/view/4105

Resumen: existe la dificultad que enfrentan las personas sordomudas para comunicarse eficazmente con quienes no conocen el lenguaje de señas. A pesar de la existencia de métodos como la escritura y la lectura de labios, estos presentan limitaciones y no siempre son efectivos. La solución propuesta incluye el desarrollo de un sistema de reconocimiento de lenguaje de señas en tiempo real, utilizando las redes neuronales convolucionales y la plataforma MediaPipe. Detecta y clasifica las posiciones de los puntos de las manos para identificar letras. Los gestos hechos frente a la cámara se traducen en letras que se almacenan para formar párrafos en una caja de texto. El tipo de investigación es cuantitativa y experimental. Al final, se destaca la importancia del reconocimiento y la enseñanza del lenguaje de señas, especialmente en países como Colombia, donde ha recibido un reconocimiento significativo.

Palabras clave: LSTM, mediapipe, reconocimiento de signos, reconocimiento de gestos.

Abstract: there is a difficulty that deaf people with speech disabilities face in communicating effectively with those who do not know sign language. Despite the existence of methods such as writing and lip-reading, these have limitations and are not always effective. The proposed solution includes developing a real-time sign language recognition system using convolutional neural networks and the MediaPipe platform. It detects and classifies the positions of hand points to identify letters. Gestures made in front of the camera are translated into letters that are stored to form paragraphs in a text box. The type of research is quantitative and experimental. Ultimately, the importance of sign language recognition and teaching is highlighted, especially in countries such as Colombia, where it has received significant recognition.

Keywords: LSTM, mediapipe, sign Language, gesture sign recognition.

1. INTRODUCCIÓN

Desde la antigüedad se ha intentado comunicar con las personas sordas con discapacidades del habla para integrarlas en la sociedad, ya sea mediante dibujos, palabras, signos o muchos otros métodos. Las personas con la dificultad de no poder hablar ni oír tienen muchos métodos de comunicación que han aprendido a lo largo de su vida, por ejemplo, el mero hecho de señalar algo ya es un método de comunicación no verbal [1], [2]. El modelo de comunicación más formal para estas personas es la lengua de signos, que aprenden a través de sus familiares o, por el contrario, viendo a otras personas hablar en lengua de signos [3]. La comunicación de las personas sordas en una sociedad con discapacidades del habla que no conocen la lengua actual es un poco confusa para ambas partes. Algunas personas sordas con discapacidades del habla pueden utilizar los medios de la escritura, en el caso de saber escribir correctamente, por otro lado, las personas con esta condición a menudo saben leer los labios lo que les permite un poco menos de complejidad a la hora de comunicarse [4].

En muchas ocasiones la persona con la condición de sordomutismo implica la dificultad de no poder escuchar lo que otra persona le está transmitiendo, esto conlleva al final a reducir la probabilidad de poder entender a otra persona, lo que provoca mucha dificultad para poder escribir correctamente. Las personas con mutismo sordo pueden escribir correctamente, pero con un cierto grado de dificultad para transmitir información [5]. Una de las formas de superar este problema es utilizar el lenguaje de signos [6], un sistema de reconocimiento de imágenes en tiempo real capaz de identificar con precisión las letras del alfabeto LIS proporcionadas por un usuario en el marco de la interacción persona-ordenador (HCI) utilizando la librería Open-Source Computer Vision (OpenCV) de Python y dos modelos basados en redes neuronales convolucionales, concretamente CNN y VGG19 [5]. La lengua de signos, al igual que la lengua hablada o escrita, es diferente en todos los países, lo que va de la mano de la lengua común [4]. Permitiendo que en cada país sea posible entrenar el software aquí propuesto para el desarrollo de un problema como la comunicación con personas sordas con discapacidades del habla [7].

Se propone un sistema robusto de reconocimiento de gestos de la mano basado en imágenes térmicas de alta resolución que son independientes de la luz. Se construye un conjunto de datos de 14.400 gestos térmicos de la mano separados en dos tonos de color [4]. Se pretende utilizar la mano del usuario como región de interés (ROI) que luego será analizada y procesada para combatir los efectos de la luz y otras anomalías que puedan surgir, para luego utilizar Redes Neuronales Artificiales (CNNs) para el aprendizaje [8].

El estudio de la lengua de señas en Colombia ha tenido una importancia significativa desde 1984 y fue reconocida oficialmente en 1996 durante el gobierno de Ernesto Samper Pizano a través de la ley 324, en cuyo artículo se lee lo siguiente "El Estado colombiano reconoce la lengua de señas como propia de la comunidad sorda del país". El artículo muestra el proceso de desarrollo y prueba de un software de traducción de la lengua de señas colombiana utilizando una serie de herramientas, como redes neuronales artificiales, aprendizaje automático, redes neuronales convolucionales y software para el uso de estas tecnologías [9], [10]. Utiliza una herramienta de IA llamada Mediapipe para el reconocimiento de rostros, objetos, poses, manos, mallas faciales y seguimiento de movimiento de objetos [11].

Otros proyectos se basan en la traducción del signo mediante la visualización de una letra en la pantalla. Una característica novedosa del software mejorado es que no sólo muestra la letra traducida en la pantalla, sino que también la emite por comando de voz [1], [6], [7], [12]. Este artículo es una extensión de la ponencia presentada originalmente en el X Congreso Iberoamericano de Interacción Persona-Ordenador JIHCI2024, 4 al 7 de junio de 2024 Universidad Tecnológica de Pereira, Colombia [13].

El artículo está organizado como sigue. Primero, la introducción donde se presenta la situación del problema. Segundo, es la metodología donde se presenta el diseño del proceso de investigación para obtener y evaluar la solución propuesta, y la tercera parte son los resultados, junto con la discusión respectiva.

2. METODOLOGÍA

El tipo de investigación será cuantitativa ya que se refiere a la aproximación a la realidad entendida como meta y considera que el investigador debe tomar distancia de esa realidad para analizarla [14], [15]. El investigador considera que "los datos deben ser descubiertos y analizados objetivamente [16], [17], [18], [19] de manera que se describan, midan, expliquen y predigan fenómenos en una población". Se ha implementado una metodología experimental

en la que se realizan pruebas con un software de reconocimiento de signos desarrollado y se propone una idea de solución para comprender a las personas sordas con discapacidades del habla. El desarrollo se basa en redes neuronales convolucionales que permiten la clasificación de imágenes por capas y el reconocimiento de las posiciones de los puntos de las manos, el software cuenta con la aplicación de la plataforma de código abierto MediaPipe que tiene detección de objetos 3D en tiempo real, por lo que detecta objetos 2D y estima sus poses a través de un modelo de aprendizaje automático [20]. Mediante el reconocimiento de estos puntos, cada gesto realizado frente a la cámara nos dirá en tiempo real a qué letra se refiere y esta información se guardará para que las secuencias de varios gestos formen un párrafo en la caja de texto que se utilizará para mostrar la traducción, (ver Fig. 1).

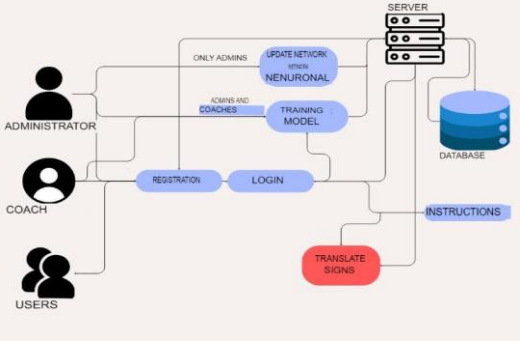

Fig. 1. *Arquitectura del sistema: Cliente - Servidor.*

Se aplican diseños de código y estructurales para el correcto funcionamiento de la red neuronal, donde se piensa en el usuario como parte fundamental del desarrollo de la tecnología existente y futura, el diseño del software se basa en características que definen al usuario como una persona con bajos conocimientos de tecnología, debe estar orientado a todo tipo de personas y además ser intuitivo, estas características facilitan la programación y la comprensión de cómo se comunica una persona sordomuda [21], [22], [23], [24], [25], (ver Fig. 2).

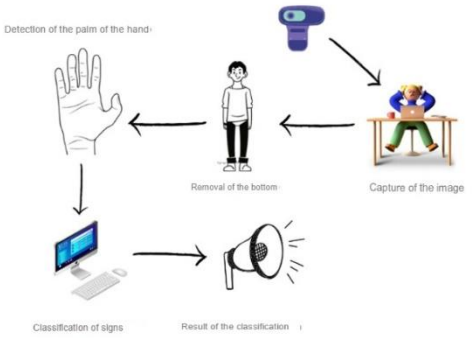

Fig. 2. *Arquitectura de usabilidad: Cliente - Software.*

En primer lugar, se realiza un entrenamiento del software para el reconocimiento de cada letra en el que se toman trescientas imágenes de la posición de cada letra; en segundo lugar, se compila el modelo y el peso de las letras y, en tercer lugar, se realiza la predicción [26]. Otra forma de ver el despliegue del proyecto sería con Inteligencia Artificial, redes neuronales (compuestas por convolucionales), y mediapipe (traducción del lenguaje de signos), [27]. De esta forma, el software se puede implementar con los siguientes pasos:

- Fase de desarrollo: Se centra en el uso de tecnologías como mediapipe para el entrenamiento de redes neuronales.
- Fase de entrenamiento: Donde se toman las trescientas imágenes de la posición de la mano relativas a cada letra.
- Fase de compilación: Se realiza el proceso de compilación de las imágenes tomadas durante el entrenamiento.
- Fase de validación: Se copia el modelo de la fase de entrenamiento en la que se comparan las dos fases para obtener un resultado óptimo.
- Fase de predicción: Al final, se pueden traducir las letras relativas a cada posición de la mano.
- Fase de evaluación: Las personas sordas con discapacidades del habla participan en la prueba del software.

Entendemos que entre dos personas para establecer una comunicación debe haber un emisor y un receptor, el emisor es el encargado de enviar un mensaje y el receptor es el encargado de recibir y entender el mensaje. El software traductor de signos no sólo puede ayudar a las personas sordas y mudas a establecer una comunicación, sino que también puede ayudar a otro tipo de personas, como, por ejemplo. Comunicación para personas sordas y sordomudas un emisor con discapacidad auditiva solo quiere enviar un mensaje a una persona sordomuda, y el receptor recibe el mensaje leyendo los labios de la persona y responde utilizando signos, el software toma esos signos y los traduce en texto y comando de voz, logrando la comunicación entre emisor y receptor, ya que aunque el emisor en este caso es una persona sordomuda y el receptor es una persona con discapacidad auditiva, pueden entender el mensaje leyendo la pantalla del ordenador sin tener que escuchar el comando de voz, (ver Fig. 3).

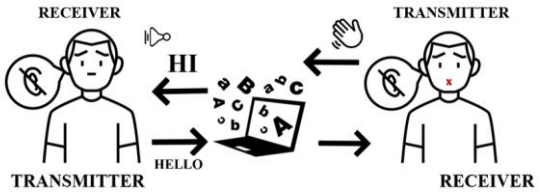

Fig. 3. *Diálogo entre un sordomudo y un ciego.*

3. RESULTADOS

Para entender cómo funciona el software, es necesario comprender las ramas de la inteligencia artificial y dónde se sitúa el artículo, (véase la Fig. 4). El filtrado de las imágenes se realiza mediante redes neuronales artificiales que clasifican el posicionamiento de los dedos dentro del marco de cada foto, realizando así la convolución deseada [11]. Las ramas de la inteligencia artificial se dividen en robótica, planificación, procesamiento del lenguaje natural, visión por ordenador, sistemas expertos, reconocimiento automático del habla, aprendizaje automático y, por último, redes neuronales [28]. Estas últimas se utilizan para entrenar la aplicación mediante convoluciones que son filtros realizados a las imágenes obtenidas en el entrenamiento por un número considerable de muestras. Teniendo como función el seguimiento de las manos a través de la plataforma, se utiliza el código libre llamado Mediapipe para la correcta traducción del lenguaje de signos a texto [29].

El manejo y entrenamiento de este software se realiza en tres fases, que requieren cierto tiempo para su correcta traducción [30]. La primera se basa en el entrenamiento del software mediante la captura de trescientas imágenes de la posición de la mano para cada letra del alfabeto; esta cantidad es para conseguir una traducción exacta de cada letra. La segunda, se realiza una compilación de las mil trescientas imágenes de cada letra del alfabeto [31]. En esta compilación se obtiene una red neuronal que se utilizará para enlazar la cámara con la base de datos de imágenes, y la tercera es la traducción en tiempo real de las imágenes de las manos captadas por la cámara. Se realizó una actualización en el uso de redes neuronales donde se aplica una nueva red llamada LSTM donde se obtienen resultados mejores y más precisos.

3.1. LSTM

Es un tipo de red neuronal más eficiente que otras utilizadas en el desarrollo de este proyecto, con mejor capacidad de aprendizaje y mejor respuesta en menos tiempo. El modelo LSTM está formado por capas que son secuenciales y constan de capas de abandono que se encargan de evitar el sobreajuste, y capas densas que son capas totalmente conectadas y tienen múltiples salidas. Cada célula LSTM consta de tres puertas principales:

- **Puerta de entrada**: Es la que determina cuánta información nueva puede entrar o debe almacenarse.
- **Compuerta de salida:** Controla cuánta información debe salir de la célula o enviarse a la siguiente.
- **Puerta del olvido:** Decide cuánta información que ya ha pasado debe ser olvidada por la célula.
- **Célula de memoria:** almacena toda la información a lo largo del tiempo

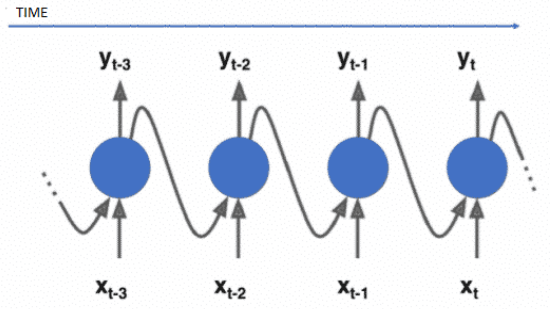

Fig. 4. *Red neuronal LSTM.* ***Fuente:*** *[11].*

3.2. Recogida de datos

Esta fase se centra en la recopilación de todos los datos que se irán incorporando al proyecto donde se irá implementando secuencialmente en función de las necesidades del producto esperado. En primer lugar, partimos de la necesidad de comunicarnos a través de un ordenador para entender a una persona sordomuda, a partir de ahí podemos registrar el comportamiento de estas personas, los medios de comunicación que utilizan, el entorno en el que mejor se defienden, o cuál es el medio de comunicación si una persona no puede entenderles, después de registrar estos datos podemos comenzar con el inicio del proyecto en el cual sabemos que no todos los países utilizan el mismo lenguaje de señas, con esta información podemos enfocar un poco más nuestro proyecto permitiendo el uso de palabras de vocabulario común para agilizar la gestión de otras fases de desarrollo del proyecto donde requerimos que nuestro software no consuma muchos recursos de procesamiento de imágenes. La recogida de datos se realiza a través de un mediapipe que detecta los puntos de las manos y analiza la posición en la que se encuentran para tomar las fotos, (ver Fig. 5).

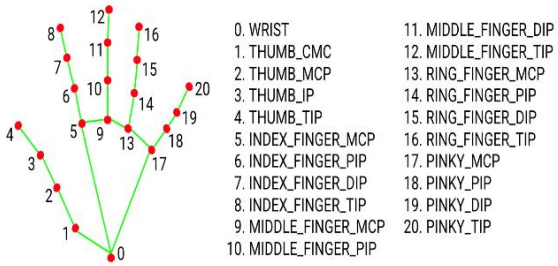

Fig. 5. *Imagen de puntos de referencia de la mano.*

3.3. Fase de desarrollo

En la fase de desarrollo, se proponen una serie de pasos para el estudio de la comunicación de las personas sordas con discapacidades del habla en los que se abordan determinadas cuestiones.

- ¿Cómo se comunican las personas sordas con discapacidades del habla?
- ¿Qué medios de comunicación utilizan las personas sordas con discapacidad del habla?
- ¿La lengua de signos es la misma en todos los países?
- ¿Qué tecnologías existen para facilitar la comunicación entre las personas sordas y con problemas de audición y las personas que hablan normalmente?
- ¿Existe un software que traduzca el signo y lo diga como un comando de voz?

3.4. Fase de formación

Los datos introducidos a partir del entrenamiento de las redes neuronales artificiales ayudan a descifrar cada letra que hay que traducir. Algunas letras resultan un poco confusas para el software, bien porque los dedos se colocan de la misma forma, bien porque algunas letras se realizan con movimiento [32]. La figura 7 muestra la fase 1, que es la fase de entrenamiento del software en la que se introdujeron trescientas imágenes de la colocación de cada letra con la mano. La dactilología debe ejecutarse con la mano más hábil de la persona, es decir, si es diestro, debe utilizarse la mano derecha, si es zurdo, la izquierda; cuando es ambidiestro, sólo debe seleccionarse una de las manos, de lo contrario, confundiría a los receptores [26], (véase la Fig. 6).

El brazo debe colocarse cómodamente, sin ningún esfuerzo añadido. Para el entrenamiento de la red se tomarán trescientas imágenes por cada letra del alfabeto para generar menor porcentaje de error, hay que tener en cuenta que la lengua de signos no es una lengua universal, cada país tiene su propio sistema. Los antecedentes aportan información sobre otros proyectos ya realizados y se concluye que todos ellos se basan en el método de redes neuronales [33]. Propone un modelo sencillo que simula la forma en que el cerebro humano procesa la información, que funciona combinando simultáneamente muchas unidades de procesamiento interconectadas que parecen versiones abstractas de una neurona. Para lograr una comunicación asertiva y una forma de expresión a través de las manos conocida como lenguaje dactilológico se busca plasmar cada letra con el software para traducirla logrando una forma de inclusión [11].

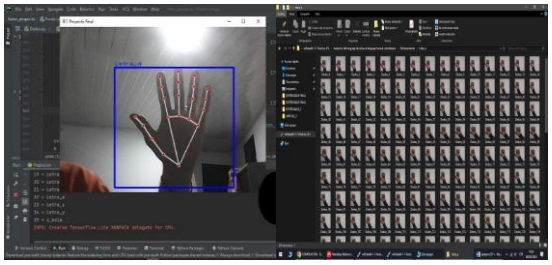

Fig. 6. *Software de entrenamiento a) Captura del movimiento de la mano b) Diferentes posiciones de cada letra con la mano.*

Cabe destacar que el software no solo sirve para traducir una letra del alfabeto, sino que también puede ser entrenado para lograr la captura de ambas manos para traducir una palabra, una recomendación es que el software solo capture imágenes estáticas, también se pretende realizar mejoras donde no solo se capturen las manos, sino también el rostro y torso de las personas para tener mayor conocimiento de la posición, estado de ánimo, y lugar en el que se encuentra para una precisión asertiva. En esta fase no es necesario volver a entrenar el modelo, sino que sólo se actualizan las nuevas palabras y se mejoran los resultados del aprendizaje dentro de la red neuronal. Esto significa que, si tenemos una palabra y al software le cuesta detectarla, se realiza un nuevo entrenamiento sobre la que ya existe y se mejora la calidad de cada palabra.

3.5. Fase de compilación

Esta fase no dispone de una interfaz gráfica en la que se pueda ver cómo se compilan las imágenes, sólo se muestra por consola el proceso de generación de la red neuronal y la generación de ficheros para su uso en la fase de predicción [11]. La fase de compilación se basa en tomar las imágenes de entrenamiento y formar una red neuronal en la que se encuentran todas las letras del alfabeto. A partir del entrenamiento realizado en la primera fase, se crean diferentes ficheros que nos ayudarán en la

siguiente fase a poder detectar las muestras introducidas por la cámara en tiempo real y también extraerlas mediante comandos de voz. Estos ficheros son los que contienen las muestras en matrices capaces de traducir una palabra o letra, (ver Fig. 7).

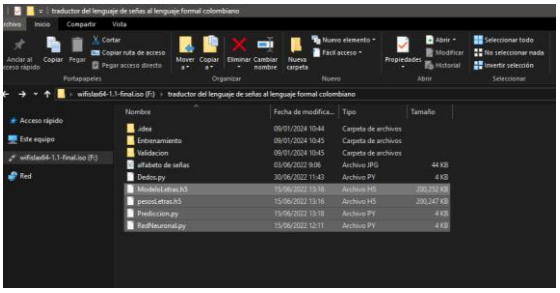

Fig. 7. *Fase de compilación.*

En este caso, por ejemplo, queremos dar reconocimiento al posicionamiento de las manos y traducir lo que ha sido previamente entrenado en el software, el uso del software es requerido para fines educativos, medios de comunicación en lugares privados y comerciales como hospitales, centros de salud, escuelas, universidades, bancos, municipalidades y lugares de atención gubernamental, en fin, un software que pueda comunicarse con personas sordomudas. Como el programa utiliza matrices según el número de palabras o letras que introduzcamos en la fase de entrenamiento, esta matriz funcionará sincronizada con las imágenes captadas por la cámara, consiguiendo mostrar la matriz y el número de columnas que hay en ella.

3.6. Fase de validación

Al igual que en la fase de entrenamiento, se toman imágenes de la cámara para cada posición de cada letra, donde se intenta en la medida de lo posible que sea la misma que en la primera fase para conseguir una mejor precisión de los datos de los puntos de las manos, ya que el software toma las fotos de la fase de entrenamiento y realiza una convolución con las fotos de la fase de validación, logrando así mostrar la letra que se intenta traducir, entender el funcionamiento del software no es complejo, es como tener 2 fotos y encontrar la diferencia en ellas, resaltar el resultado y mostrarlo, la fase de validación tiene las mismas fotos del entrenamiento, la misma cantidad de imágenes para cada letra y la misma cantidad de letras, (ver Fig. 8).

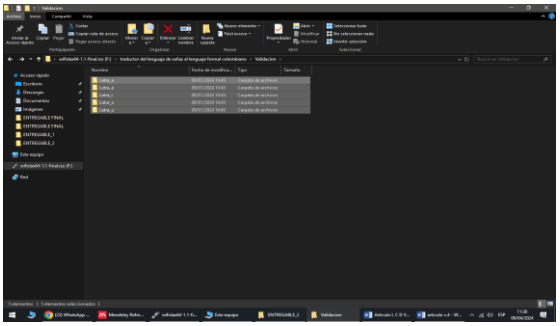

Fig. 8. *Fase de validación.*

3.7. Fase de predicción

El proceso de predicción en el que el software compara las imágenes del entrenamiento con las de la cámara, consiguiendo tomar veintiún puntos de las manos, obteniendo una letra en la caja además de obtener a través del altavoz la letra que se está traduciendo, (ver Fig. 9).

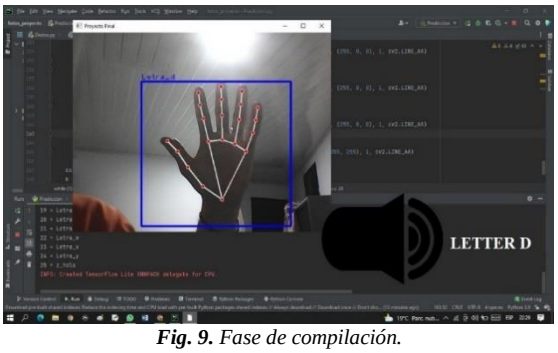

Fig. 9. *Fase de compilación.*

El software traductor de lengua de signos es posible porque no requiere de altos recursos o máquinas caras, hoy en día se puede utilizar un ordenador y una webcam para el uso de la aplicación. El software será capaz de traducir el alfabeto de signos en tiempo real y almacenar las letras traducidas en un cuadro de texto para ensamblar la palabra deseada, y para evitar almacenar letras no deseadas, el almacenamiento de las letras se reflejará a través de una tasa de aprobación del 98%. La aplicación se construirá a partir de mediapipe, una herramienta de código abierto con diferentes funcionalidades, entre las que se encuentra el reconocimiento de los veintiún puntos de las manos. Mediapipe en coordinación con Python utiliza el aprendizaje automático que consiste en varios métodos que están interconectados y trabajan juntos. El primer modelo consiste en detectar la palma de la mano de la imagen que se considera como entrada. Este modelo devolverá un cuadro delimitador orientado a la mano de la imagen de la palma capturada. El cuadro delimitador se utilizará para recortar la imagen de la mano y, a continuación, se introducirá

en la base de datos [11]. Por ejemplo, para el entrenamiento de vocales, el software creará una matriz (A) de cinco columnas donde cada columna representa cada vocal, esta vocal representa la vocal que se está visualizando en la cámara que representa la letra U.

$$[0,0,0,0,1] = A \quad (1)$$

3.8. Fase de evaluación

Tenemos previsto incorporar pruebas de usabilidad en las que se realicen las pruebas pertinentes, (véase la Fig. 10). En la fase de evaluación del modelo previamente entrenado, se comparan las muestras que se están tomando en tiempo real con las muestras de entrenamiento. Por eso es importante realizar el entrenamiento con datos precisos y una buena calidad de imagen para obtener resultados óptimos. Esta fase es la última, la que permite detectar fallos en el modelo y, en base a ello, realizar la optimización de los resultados. Esto, con el fin de generar mejoras al software como el reconocimiento de manos en ambientes con luz variable, diferentes colores de piel, diferentes tamaños de manos, profundidad a la que se encuentran las manos, y otros factores que podrían tener una actualización significativa de los datos de este [21], [22], [23], [24], [25], [34], [35].

Figura 10. *Feria de proyectos Universidad Francisco de Paula Santander Ocaña.*

3.9. Limitaciones técnicas

En el desarrollo del proyecto se han utilizado diferentes componentes para cumplir objetivos específicos. Varios de estos componentes presentan actualizaciones que son de gran ayuda para futuros avances en los que nos apoyaremos en el futuro para implementar nuevas tecnologías y tener un mayor aprovechamiento del software. El uso de estas tecnologías ayuda a obtener el resultado deseado, combinándolas con la traducción de signos, como la implementación de mediapipe, que nos permite detectar las manos en determinados fotogramas de una muestra de la que se pueden extraer los puntos de los dedos. Y las coordenadas donde están situados. Al inicio del proyecto, sólo era posible traducir letras del alfabeto, teniendo dificultades con el software ya que algunas letras eran similares en cuanto al posicionamiento de las manos y en aquel momento se utilizaban tecnologías algo lentas para el tratamiento de imágenes, como el uso de redes neuronales CNN junto con un ordenador con poca capacidad para la recogida y tratamiento de muestras. Las dificultades actuales del proyecto se basan en las versiones de las librerías donde solo se utilizan versiones específicas para su correcto funcionamiento ya que si se utilizan actualizaciones nuevas o antiguas se obtienen resultados erróneos o no se procesan las imágenes registradas o por capturar, por esta razón se utilizan librerías con versiones únicas.

Tensorflow, Keras, y mediapipe entre otras son las aplicaciones o herramientas utilizadas, y que se muestran a continuación con sus respectivas versiones correctas para trabajar con el software.
Tensorflow==2.10.1, mediapipe==0.10.11, numpy==1.26.4, tables==3.9.2, OpenCV-python==4.9.0.80, pandas==2.2.1, protobuf==3.20.3, keras==2.10.0, h5py==3.10.0, flatbuffers==24.3.7, gTTS==2.5.1, pygame==2.5.2

4. CONCLUSIONES

Para lograr una detección eficaz y evitar la confusión de letras, los gestos del lenguaje de las huellas dactilares deben estar perfectamente definidos. Mientras que los enfoques actuales del estado de la técnica se basan principalmente en potentes entornos de escritorio para la inferencia, nuestro método logra un rendimiento en tiempo real en un teléfono móvil e incluso se adapta a múltiples manos. Se desarrolló un software de aplicación que hace hincapié en el aprendizaje del lenguaje de signos, con técnicas de procesamiento digital de imágenes en las que el usuario debe ser capaz de coordinar la inclinación, la rotación y el movimiento para aprender los símbolos más fácilmente.

Hay que considerar que las letras ñ y z no son imágenes estáticas para el programa, deben ser imágenes con movimiento. Hay que considerar que los gestos del lenguaje de signos deben estar perfectamente definidos ya como reconocimiento de cámara, para evitar confusiones entre letras. Los

signos más utilizados en el lenguaje de signos son de la A a la Z, considerando que letras como la m, n y r, tienen cierto grado de confusión en el programa.

El desarrollo del proyecto se vislumbra en el futuro como una plataforma de ayuda para la correcta comunicación con personas sordas con discapacidades del habla implantada en diversos lugares como centros comerciales, colegios y bancos. Será capaz de traducir la lengua de signos en ambos sentidos, de signo a texto y de texto a signo, sin olvidar que también se podrá realizar la implementación de ayudas de voz para personas ciegas. El software podría mejorarse implementando el entrenamiento automático y la corrección de errores con IA para conseguir un software autónomo en la implementación de nuevas palabras o abreviaturas de estas como signos.

Diferentes opiniones sobre el desarrollo del software han ayudado a otros desarrollos del proyecto, como, por ejemplo, las personas que se quedan sin manos y tienen la condición de no poder hablar ni oír, se realizará la implementación del seguimiento ocular y un teclado en pantalla. El uso futuro de tecnologías innovadoras en el software ayudará a realizar pruebas en diferentes lugares para cubrir gran parte de la población de Colombia donde se solucionaría a gran escala la comunicación con personas sordas con discapacidad del habla.

LSTM es mejor en tiempo de respuesta en comparación con las redes neuronales CNN, esto es gracias a su capacidad para aprender y responder en un corto periodo de tiempo. A pesar de los avances en la precisión del reconocimiento de signos, el software se enfrenta a dificultades para diferenciar ciertas letras que tienen posiciones similares de los dedos o implican movimientos. Por ello, se prevén mejoras que incluyan la detección de expresiones faciales y posiciones del torso para ofrecer una traducción más precisa y completa del lenguaje de signos.

REFERENCIAS

[1] B. Villa, V. Valencia, and J. Berrio, “Digital image processing applied on static sign language recognition system/Diseño de un Sistema de Reconocimiento de Gestos No Móviles mediante el Procesamiento Digital de Imágenes,” Prospectiva, vol. 16, no. 2, pp. 41–48, 2018, doi: 10.15665/rp.v16i2.1488.

[2] I. De Souza, P. Mattos, C. Pina, and D. Fortes, “ADHD: The impact when not diagnosed,” J Bras Psiquiatr, vol. 57, no. 2, pp. 139–141, 2008, doi: 10.1590/s0047-20852008000200010.

[3] V. J. Schmalz, “Real-time Italian Sign Language Recognition with Deep Learning,” in CEUR Workshop Proceedings, 2022, pp. 45–57.

[4] D. S. Breland, A. Dayal, A. Jha, P. K. Yalavarthy, O. J. Pandey, and L. R. Cenkeramaddi, “Robust Hand Gestures Recognition Using a Deep CNN and Thermal Images,” IEEE Sens J, vol. 21, no. 23, pp. 26602–26614, Dec. 2021, doi: 10.1109/JSEN.2021.3119977.

[5] D. S. Breland, S. B. Skriubakken, A. Dayal, A. Jha, P. K. Yalavarthy, and L. R. Cenkeramaddi, “Deep Learning-Based Sign Language Digits Recognition from Thermal Images with Edge Computing System,” IEEE Sens J, vol. 21, no. 9, pp. 10445–10453, May 2021, doi: 10.1109/JSEN.2021.3061608.

[6] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, “Machine learning methods for sign language recognition: A critical review and analysis,” Nov. 01, 2021, Elsevier. doi: 10.1016/j.iswa.2021.200056.

[7] R. Rastgoo, K. Kiani, and S. Escalera, “Hand sign language recognition using multi-view hand skeleton,” Expert Syst Appl, vol. 150, p. 113336, Jul. 2020, doi: 10.1016/j.eswa.2020.113336.

[8] H. Liu, Z. Zhang, J. Qian, W. Wang, and K. Kang, “Hand gesture recognition based on deep neural network and sEMG signal,” in IEEE International Conference on Robotics and Biomimetics, ROBIO 2019, Institute of Electrical and Electronics Engineers Inc., Dec. 2019, pp. 3001–3006. doi: 10.1109/ROBIO49542.2019.8961445.

[9] J. A. Zea Gutiérrez, M. J. Suárez Barón, and J. S. González Sanabria, “Aprendizaje por refuerzo como soporte a la predicción de la precipitación mensual. Caso de estudio: Departamento de Boyacá - Colombia,” TecnoLógicas, vol. 27, no. 60, p. e3017, Jun. 2024, doi: 10.22430/22565337.3017.

[10] L. Antonelli, M. Lezoche, and J. Delle Ville, “Knowledge Extraction from the Language Extended Lexicon Glossary Using Natural Language Processing,” TecnoLógicas, vol. 27, no. 59, p. e2917, Apr. 2024, doi: 10.22430/22565337.2917.

[11] L. D. Torrado-Mora and D. Rico-Bautista, “Systematic Mapping: Translator Language from Sign Language to Colombian Formal

Language," in Proceedings - 3rd International Conference on Information Systems and Software Technologies, ICI2ST 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 44–48. doi: 10.1109/ICI2ST57350.2022.00014.

[12] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," Expert Syst Appl, vol. 182, p. 115657, Nov. 2021, doi: 10.1016/j.eswa.2021.115657.

[13] L. D. Torrado-Mora, C. A. Collazos, and D. Rico-Bautista, "Sign Language to Colombian Formal Language Trans-lation Software," in Human-Computer Interaction. HCI-COLLAB 2024. Communications in Computer and Information Science, V. Agredo-Delgado, P. H. Ruiz, and C. A. Meneses, Eds., SPRINGER, 2024, pp. 1–14.

[14] R. Hernández Sampieri, C. Feránadez Collado, and M. D. P. Baptista Lucio, Metodología de la investigación. McGraw Hill España, 2014.

[15] L. A. Tovar, "Elaboración de tesis. Estructura y metodología," Trillas, p. 384, 2017.

[16] M. Gómez, Introducción a la metodología de la investigación científica, 1st ed. Argentina, 2015.

[17] G. Briones, Metodología de la investigación cuantitativa en las ciencias sociales, vol. 1, no. 958-9329-09–8. 2002. doi: 10.1038/2191218a0.

[18] E. Del-Canto and A. Silva, "Metodología Cuantitativa: Abordaje desde la complementariedad en ciencias sociales," Rev Cienc Soc, vol. 0, no. 141, p. 45, 2013, doi: 10.15517/rcs.v0i141.12479.

[19] A. Radmehr, M. Asgari, and M. T. Masouleh, "Experimental Study on the Imitation of the Human Head-and-Eye Pose Using the 3-DOF Agile Eye Parallel Robot with ROS and Mediapipe Framework," in 2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM), IEEE, Nov. 2021, pp. 472–478. doi: 10.1109/ICRoM54204.2021.9663445.

[20] S. Adhikary, A. K. Talukdar, and K. Kumar Sarma, "A Vision-based System for Recognition of Words used in Indian Sign Language Using MediaPipe," in Proceedings of the IEEE International Conference Image Information Processing, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 390–394. doi: 10.1109/ICIIP53038.2021.9702551.

[21] T. Granollers i Saltiveri, "MPIu+a. Una metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares," 2004.

[22] J. Lorés and T. Granollers, "Ingeniería de la Usabilidad y de la Accesibilidad aplicada al diseño y desarrollo de sitios web," no. May, pp. 3–7, 2004.

[23] T. Granollers, "Usability Evaluation with Heuristics . New Proposal from Integrating Two Trusted Sources 2 Combining Common Heuristic Sets," pp. 1–16, 2018.

[24] T. Granollers, "Diseño Centrado en el Usuario (DCU). El modelo MPlu+a," p. 71, 2013.

[25] S. Inform and L. Lleida, "MPIu + a . Una metodología que integra la ingeniería del software, la interacción persona-ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares," 2004.

[26] S. Ikram and N. Dhanda, "American Sign Language Recognition using Convolutional Neural Network," in 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies, GUCON 2021, Institute of Electrical and Electronics Engineers Inc., Sep. 2021. doi: 10.1109/GUCON50781.2021.9573782.

[27] A. Radmehr, M. Asgari, and M. T. Masouleh, "Experimental Study on the Imitation of the Human Head-And-Eye Pose Using the 3-DOF Agile Eye Parallel Robot with ROS and Mediapipe Framework," in 9th RSI International Conference on Robotics and Mechatronics, ICRoM 2021, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 472–478. doi: 10.1109/ICRoM54204.2021.9663445.

[28] V. Chunduru, M. Roy, N. S. Dasari Romit, and R. G. Chittawadigi, "Hand Tracking in 3D Space using MediaPipe and PnP Method for Intuitive Control of Virtual Globe," in IEEE Region 10 Humanitarian Technology Conference, R10-HTC, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/R10-HTC53172.2021.9641587.

[29] N. F. Thejowahyono, M. V. Setiawan, S. B. Handoyo, and A. H. Rangkuti, "Hand Gesture Recognition as Signal for Help using Deep Neural Network," International Journal of Emerging Technology and

Advanced Engineering, vol. 12, no. 2, pp. 37–47, Feb. 2022, doi: 10.46338/ijetae0222_05.

[30] S. Njazi and S. Ng, “Veritas: A Sign Language-To-Text Translator Using Machine Learning and Computer Vision,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Nov. 2021, pp. 55–60. doi: 10.1145/3507623.3507633.

[31] M. L. Wehmeyer et al., “The intellectual disability construct and its relation to human functioning,” Intellect Dev Disabil, vol. 46, no. 4, pp. 311–318, Jan. 2008, doi: 10.1352/1934-9556(2008)46[311:TIDCAI]2.0.CO;2.

[32] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh, “Sign Language Recognition System Using TensorFlow Object Detection API,” in Communications in Computer and Information Science, Springer Science and Business Media Deutschland GmbH, Jan. 2022, pp. 634–646. doi: 10.1007/978-3-030-96040-7_48.

[33] S. Gulati, A. K. Rastogi, M. Virmani, R. Jana, R. Pradhan, and C. Gupta, “Paint/Writing Application through WebCam using MediaPipe and OpenCV,” in Proceedings of 2nd International Conference on Innovative Practices in Technology and Management, ICIPTM 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 287–291. doi: 10.1109/ICIPTM54933.2022.9753939.

[34] T. Granollers, “Usability evaluation with heuristics. New proposal from integrating two trusted sources,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2018, pp. 396–405. doi: 10.1007/978-3-319-91797-9_28.

[35] J. Lorés Vidal and T. Granollers, “La Ingeniería de la usabilidad y de la accesibilidad aplicada al diseño y desarrollo de sitios web,” 2004.