Universidad Autónoma de Querétaro

Facultad de Informática

Maestría en Ciencias de la Computación

Sistema de reconocimiento del Lenguaje de Señas Mexicano basado en una cámara RGB-D y aprendizaje automático

Tesis

Que como parte de los requisitos

para obtener el Grado de

Maestro en Ciencias de la Computación

Presenta

Ing. Kenneth Mejía Pérez

Dirigido por:

Dra. Diana Margarita Córdova Esparza

Querétaro, Qro. a 01 de Agosto de 2022

Universidad Autónoma de Querétaro

Facultad de Informática

Maestría en Ciencias de la Computación

Sistema de reconocimiento del Lenguaje de Señas Mexicano basado en una cámara RGB-D y aprendizaje automático

Tesis

Que como parte de los requisitos

para obtener el Grado de

Maestro en Ciencias de la Computación

Presenta

Ing. Kenneth Mejía Pérez

Dirigido por:

Dra. Diana Margarita Córdova Esparza

Dra. Diana Margarita Córdova Esparza
Presidente

Dra. Ana Marcela Herrera Navarro
Secretario

M. en C. Fidel González Gutiérrez
Vocal

M.I.S.D Erika del Rio Magaña
Suplente

Dr. Alfonso Ramírez Pedraza
Suplente

Centro Universitario, Querétaro, Qro.
Agosto, 2022
México

Dedicatoria

Dedico este proyecto de investigación a mi familia, quienes me han apoyado durante toda mi vida, que han estado presentes en los buenos y malos momentos, para darme ese apoyo necesario para seguir adelante pese a cualquier obstáculo.

Agradezco infinitamente a mi mamá Marlene Alejandra Pérez Trejo y a mi papá Nacianceno Mejía Morales por darme la vida y las oportunidades necesarias para poder vivirla, agradezco sus enseñanzas, perseverancia y dedicación para que yo pudiera crecer como un hombre de bien que se esfuerza por alcanzar sus objetivos, agradezco que, a pesar de no tener mucho, me dieron todo para poder realizarme personal y profesionalmente.

Agradezco a mis hermanos Diego Alejandro y Allan Mejía Pérez por estar presentes como amigos y familia en toda mi vida, por crecer, llorar y reír juntos.

Agradezco a mi novia Nayeli Pamela Perales Soto quien desde hace más de 5 años ha estado a mi lado como compañera de vida y de aventuras, por compartir esta etapa tan importante, estos proyectos tan ambiciosos, además de las experiencias, alegrías y penas que le tocan vivir a un estudiante en su formación personal y profesional.

Agradezco a mi tío Everardo Mejía Morales, por extenderme su apoyo incondicional durante toda mi formación profesional, por creer en mí, por sus enseñanzas, lecciones de vida y por nunca darme la espalda.

Una especial dedicatoria y agradecimiento a mi abuela Margarita Morales Jiménez, quien en vida y después de ella fue y es una segunda madre para mí, por todo su apoyo y afecto que me dio en los momentos más difíciles de mi vida.

Agradecimientos

Primero agradezco a la Universidad Autónoma de Querétaro por brindarme la oportunidad de continuar mi trayecto profesional con la maestría en Ciencias de la Computación.

Agradezco a los docentes de la Facultad de Informática que gracias a su esfuerzo y dedicación pude adquirir el conocimiento necesario para egresar de este programa de estudios.

Agradezco especialmente a la Dra. Diana Margarita Córdova Esparza quien gracias a su dedicación, paciencia y compromiso pude conseguir y superar las expectativas de las metas planteadas al inicio del proyecto, además agradezco al Dr. Juan Ramon Terven Salinas quien, dedico muchas horas de su tiempo libre, para brindarme su conocimiento y experiencia.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado para la realización de mis estudios de posgrado, con número de CVU: 1077850, en el tiempo comprendido entre el 01 de agosto del 2020 al 31 de julio del 2022.

Agradezco a la Universidad Autónoma de Querétaro quien me brindo apoyo para equipo de cómputo e insumos a través del Fondo de Proyectos Especiales de Rectoría (FOPER) mediante el proyecto FOPER-2021-FIF02482.

Finalmente, agradezco la asesoría brindada por la empresa Manos Libres y la comisión de personas sordas del estado de Querétaro quienes a través del diplomado de Lengua de Señas Mexicana nivel 1 me brindaron asesoría particular para abordar la temática de este proyecto.

Índice

Contenido de la tesis
Dedicatoria
Agradecimientos
Índice
Índice de Cuadros 6
Índice de Figuras 7
Abreviaturas y siglas 10
Resumen en español 11
Abstract 12
Introducción 13
Planteamiento del problema 14
Justificación 15
Antecedentes 15
Hipótesis 32
Objetivo 32
Metodología 33
Resultados y discusión 43
Conclusiones 53
Referencias 54

Índice de Cuadros

Tabla 1: Descripción del cuerpo de los datos. 34

Tabla 2: Variaciones en las arquitecturas de cada modelo neuronal utilizado para la clasificación de las señas 37

Tabla 3: Porcentaje de exactitud para los modelos neuronales utilizados. 43

Tabla 4: Precisión de la clasificación realizada para datos con y sin ruido. 45

Tabla 5: Precisión de clasificación de múltiples modelos LSTM entrenados con diferentes niveles de aumento de ruido para las entradas evaluadas en conjuntos de prueba ruidosos. 46

Tabla 6. Variación de los resultados de la arquitectura LSTM. 50

Tabla 7. Porcentaje de precisión del modelo al variar las características de entrada. 51

Índice de Figuras

Figura 1. Modelo de una neurona computacional. 20

Figura 2: Representación gráfica de la activación lineal. 22

Figura 3: Representación gráfica de la activación sigmoidea. 22

Figura 4: Representación gráfica de la función de tangente hiperbólica. 23

Figura 5: Representación gráfica de la función de salida identidad. 24

Figura 6: Representación gráfica de la función binaria. 24

Figura 7: Representación gráfica de la función escalón. 25

Figura 8. Estructura de una red neuronal. 26

Figura 9. Estructura de una red neuronal recurrente. 28

Figura 10. Representación gráfica de una neurona LSTM. 29

Figura 11. Representación gráfica de una neurona GRU. 31

Figura 12. Metodología desarrollada para el sistema de reconocimiento automático. 33

Figura 13: Cámara de profundidad OAK-D. 34

Figura 14: Puntos característicos seleccionados de la cara, el cuerpo y la mano. 35

Figura 15: Puntos clave de la cara, las manos y el cuerpo para la seña “gracias”. 36

Figura 16: Curvas de entrenamiento para modelos con 32 neuronas en la capa de entrada y 16 neuronas en la capa de salida. 39

Figura 17: Curvas de entrenamiento para modelos con 64 neuronas en la capa de entrada y 32 neuronas en la capa de salida. 39

Figura 18: Curvas de entrenamiento para modelos con 128 neuronas en la capa de entrada y 64 neuronas en la capa de salida. 40

Figura 19: Curvas de entrenamiento para modelos con 256 neuronas en la capa de entrada y 128 neuronas en la capa de salida. 40

Figura 20: Curvas de entrenamiento para modelos con 512 neuronas en la capa de entrada y 256 neuronas en la capa de salida. 41

Figura 21: Curvas de entrenamiento para modelos con 1024 neuronas en la capa de entrada y 512 neuronas en la capa de salida. 41

Figura 22. Curvas de *precision–recall* de las mejores arquitecturas obtenidas para cada modelo neuronal. 44

Figura 23: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, sin ruido gaussiano. 47

Figura 24: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 10 cm. 47

Figura 25: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 20 cm. 48

Figura 26: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 30 cm. 48

Figura 27: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 40 cm. 49

Figura 28: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 50 cm. 49

Figura 29. Curvas *precision-recall* obtenidas para la combinación de características utilizadas en cada modelo neuronal LSTM. 52

Abreviaturas y siglas

Ecuación (Ec.)

Lengua de Señas (LS)

Lengua de Signos Española (LSE)

Lengua de Señas Mexicana (LSM)

Lengua de Señas Americana (ASL, por sus siglas en inglés American Sign Language)

Lengua de Señas Británica (BSL, por sus siglas en inglés, British Sign Language)

Lengua de Señas Maya (LSMy)

Redes Neuronales Recurrentes (RNN, por sus siglas en inglés, Recurrent Neural Networks)

RGB, por sus siglas en inglés Red Green Blue

Memoria a Corto y Largo Plazo (LSTM, por sus siglas en inglés, Long short-term memory)

Unidad recurrente cerrada (GRU, por sus siglas en inglés, Gated Recurrent Unit)

Resumen

El reconocimiento automático de la lengua de señas es una tarea compleja en el área de visión por computadora y aprendizaje automático. La mayoría de los trabajos que se encuentran en la literatura se han centrado en reconocer la lengua de señas usando solo gestos con las manos. Sin embargo, el movimiento del cuerpo y los gestos faciales juegan un papel esencial en la interacción de la lengua de señas. Teniendo esto en cuenta, en este proyecto de investigación se desarrolló un sistema de reconocimiento de la lengua de señas basado en la detección de puntos característicos de las manos, el cuerpo y la cara que se emplean al realizar una seña. Para la adquisición de las señas se utilizó una cámara de profundidad con el propósito de obtener las coordenadas 3D que caracterizan cada seña, capturando un total de 3000 secuencias de datos que corresponden a 30 señas estáticas y dinámicas de la Lengua de Señas Mexicana. Para la clasificación automática del conjunto de señas, se evaluaron tres arquitecturas diferentes que permiten el tratamiento de secuencias temporales: la red neuronal recurrente (RNN, por sus siglas en inglés), la memoria a corto y largo plazo (LSTM, por sus siglas en inglés) y la unidad recurrente cerrada (GRU, por sus siglas en inglés). Para evaluar el rendimiento de cada clasificador se calculó la precisión, la recuperación y la exactitud. Al finalizar la etapa de experimentación y validación de resultados, se puede concluir que la memoria a corto plazo (LSTM) funcionó mejor con entradas ruidosas y la unidad recurrente cerrada (GRU) funcionó mejor sin entradas ruidosas y con menos parámetros entrenables.

Palabras clave: LSM, clasificación, redes neuronales, cámara RGB-D.

Abstract

Automatic sign language recognition is a complex task computer vision and machine learning. Most of the work found in the literature has focused on recognizing sign language using only hand gestures. However, body movement and facial gestures play an essential role in sign language interaction. Taking this into account, in this research project, a sign language recognition system was developed based on detecting feature landmarks of the hands, body, and face that are used when making a sign. For the acquisition of the signs, a depth camera was used to obtain the 3D coordinates that characterize each sign, capturing a total of 3000 data sequences corresponding to 30 static and dynamic signs of the Mexican Sign Language. For the automatic classification of the set of signs, three different architectures that allow the treatment of temporal sequences were evaluated: Recurrent Neural Networks (RNN), Long Short-Term Memories (LSTM), and Gated Recurring Units (GRU). To evaluate the performance of each classifier, precision, recall, and accuracy were calculated. At the end of the experimentation and validation of the results stage, it can be concluded that the short-term memory (LSTM) performed better with noisy inputs, and the closed recurrent unit (GRU) performed better without noisy inputs and with fewer trainable parameters.

Keywords: MSL, classification, neural networks, RGB-D camera.

I. Introducción

Las personas pertenecientes a la comunidad sorda presentan una dificultad para la comunicación oral en una o ambas direcciones (como receptores o emisores). Por tal motivo, la comunidad ha desarrollado su propia lengua, es decir la lengua de señas (LS).

Se le conoce como lengua de señas, a la forma estructurada de comunicación viso-gestual, en la que los participantes utilizan principalmente las manos como medio de comunicación para la realización de señas, las cuales poseen una estructura fonológica (querológica), a su vez, como lo menciona el autor Sánchez-Barrera (Sánchez-Barrera , 2018) están apoyadas de expresiones faciales, movimientos corporales y escenarios virtuales, los cuales, en conjunto ofrecen una forma de comunicación completa y efectiva.

En el mundo existen diversas lenguas de señas, éstas por lo regular se establecen de acuerdo con el país de los señantes y no por su lengua oral nacional, es decir, a pesar de que muchos países en el mundo compartan el mismo idioma, como lo es el español o el inglés, los países adoptan lenguas de señas distintas y éstas se ven influenciadas por variaciones demográficas o incluso por lenguas de señas extranjeras. Por ejemplo, en España se practica la Lengua de Signos Española (LSE), mientras que en México se realiza la Lengua de Señas Mexicana (LSM), analógicamente la Lengua de Señas Americana (ASL, por sus siglas en ingles American Sign Language) es un idioma completamente distinto a la Lengua de Señas Británica (BSL, por sus siglas en inglés, British Sign Language).

La lengua de señas no es una forma de comunicación reciente, si no que existe evidencia documental de su existencia desde hace algunos siglos, tal es el caso del libro “Reducción de las letras y arte para enseñar a hablar a los mudos” de Pablo Bonet, (Pablo Bonet 1620), el cual es conocido como uno de los primeros tratados modernos de la fonética de la lengua de señas en el mundo. Por otra parte, la Lengua de Señas Mexicana (LSM) ha sido un objeto de investigación el cual se ha trabajado desde la década de los ochenta, siendo que esta es reconocida como una lengua oficial mexicana a partir del 10

de junio del 2005 (Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad, 2017). Entre estas investigaciones destacadas se pueden mencionar al estudio de Smith Stark titulado “La lengua manual mexicana” (Smith Stark,1986) en el cual relata el desconocimiento que se tenía sobre la gramática de la LSM, ya que en aquella época se sabía muy poco sobre la variación dialectal existente y las lenguas de señas relacionadas, por lo que, este autor propone un sistema de transcripción de las señas en general y de la LSM en particular. Treinta años después, como se menciona en el trabajo de Aldrete y Serrano (Aldrete M. C. y Serrano J. 2018) se observa un auge entre las investigaciones relacionadas a este tema, en donde se abordan temas como sus variaciones demográficas, procesos de migración, influencia de lenguas de señas extranjeras (en particular ASL) o su coexistencia en México junto a otras lenguas emergentes, rurales o indígenas como podría ser la Lengua de Señas Maya (LSMy).

1.1 Planteamiento del problema

De acuerdo con datos de la OMS (OMS, 2019) 466 millones de personas en el mundo padecen algún problema de audición, sin embargo como se observa en la actualización de datos del 2021 (OMS 2021) esta cifra aumento a 1500 millones de personas, particularmente, en México existen 4,250,910 de personas con limitaciones o discapacidades auditivas, de las cuales 1,350,802 padecen hipoacusia severa o profunda (considerada como discapacidad auditiva o sordera), mientras que 2,900,108 padecen hipoacusia leve o moderada (limitación auditiva) (INEGI, 2020). En el año 2020, se realizó el censo poblacional nacional determinando que México tiene 126,014,024 de habitantes (INEGI, 2021) y alrededor del 3.37 % de la población en México censada presenta algún problema auditivo importante.

La lengua de señas mexicana es una forma de comunicación eficiente entre los usuarios de ésta, sin embargo, aún existe una brecha en la comunicación entre la comunidad sorda y las personas oyentes siendo incluso imposible establecer una comunicación con aquellos que desconocen por completo la lengua de señas, (Serafín, M. y González, R.,

2011), esto genera problemáticas sociales, como: la falta de educación, empleo, desarrollo personal, entre otros.

1.2 Justificación

La LSM consiste en expresar un conjunto de signos gestuales con las manos y además hace uso de expresiones faciales, posturas corporales entre otros signos no gestuales, con el propósito de ser un medio de comunicación para las personas con problemas auditivos y la comunidad sorda en México. Por tal motivo, el estudio de este tema tiene relevancia y se ha trabajado desde hace algunas décadas desde diferentes áreas del conocimiento.

Particularmente, en este trabajo de investigación se implementó un sistema para el reconocimiento de la LSM que ofrece una propuesta de solución para tratar de reducir las barreras de comunicación de la comunidad sorda en México. Mediante el sistema es posible identificar un conjunto de palabras del Lenguaje de Señas Mexicano, haciendo uso de un algoritmo de aprendizaje automático y una cámara de profundidad (RGB-D) como medio de obtención de datos, ya que como se observa en el estado del arte es un dispositivo que permite obtener datos de profundidad que son invariantes a cambios de iluminación en la escena, color de piel y de ropa de la persona señante, etc.

II. Antecedentes

Desde inicios de la década pasada se pueden encontrar trabajos en el estado del arte sobre el desarrollo de sistemas para el reconocimiento automático de la LSM, como lo mencionan los autores Solís et. al (Solís F. et al, 2016), es posible clasificar estos trabajos en dos grupos principales de acuerdo con los métodos que emplean para la adquisición de información: el grupo 1 se refiere a dispositivos electrónicos y sensores que el usuario debe portar con el propósito de capturar información precisa sobre las manos, dedos y brazos. Por otro lado, el grupo 2 hace referencia a dispositivos ópticos para la obtención de datos como lo son: cámaras de color (RGB, por sus siglas en inglés Red Green Blue), o cámaras de profundidad (RGB-D, por sus siglas en inglés Red Green Blue Depth), junto

a estos dispositivos se desarrollan técnicas de visión artificial para poder clasificar las señas.

Ambos grupos presentan ventajas y desventajas uno respecto con del otro, ya que por una parte, los sistemas pertenecientes al grupo 1 presentan una alta precisión en la captura de datos del movimiento de las manos, como se puede observar en el trabajo de Saldaña et al. (Saldaña- González G. et al 2018) en el cual se creó un guante traductor para la LSM, en el cual se utilizan sensores electrónicos como: acelerómetro y giroscopio, para obtener el nivel de inclinación, la posición de la mano, el nivel de flexión de los dedos, entre otras mediciones necesarias para clasificar las señas. Sin embargo, como lo menciona Solís (Solís F. et al, 2016) la desventaja que tienen estos dispositivos es que el usuario siempre debe portarlos al ejecutar la lengua de señas y esto afecta la naturalidad con la que se expresa.

Los sistemas pertenecientes al segundo grupo tienen la ventaja de que el usuario puede realizar las señas con mayor naturalidad al no necesitar llevar algún dispositivo puesto consigo. Sin embargo, como desventaja en muchos casos es que las cámaras de color y profundidad durante la captura de los datos se puede adquirir ruido, por lo tanto, es preciso capturar los datos en ambientes controlados, ubicando un escenario libre de objetos, con fondos de color sólido, y en donde se encuentre únicamente el participante en la escena mirando hacia la cámara. Algunos ejemplos de estas condiciones se describen en los trabajos de Solís (Solís F. et al, 2015; Solís F. et al, 2016), Cervantes (Cervantes J. 2016) y Martínez (Martínez, M. 2016), limitando sus posibles aplicaciones en un ambiente natural.

Por otra parte, es importante mencionar que para poder realizar el reconocimiento automático de señas es necesario disponer de un conjunto de señas que permitan realizar el entrenamiento de dichos sistemas. En la literatura se encuentran diversas bases de datos que constan de imágenes o videos de señas para distintos países del mundo. Sin embargo, en referencia a la lengua de señas mexicana, solo existen dos

bases de datos de libre acceso, la primera de ellas es la base de datos descrita en el trabajo de Rivas-Perea, (Rivas-Perea P. 2019), la cual consiste en imágenes de profundidad de señas estáticas específicamente números del 0 al 9, las cuales fueron adquiridas con un sensor Kinect, para cada una de las clases se incluyen 300 repeticiones, es decir, se obtuvieron un total de 3,000 imágenes. La segunda base de datos pertenece a Galicia et al, (Galicia R. et al, 2015), formada por 21 clases distintas, con 300 repeticiones por clase dando un total de 6,300 imágenes, al igual que la base de datos presentada en Rivas-Perea, (Rivas-Perea P 2019), ésta se enfoca en 21 señas estáticas del alfabeto que hacen uso de una sola mano, cabe mencionar que las imágenes fueron tomadas mediante una cámara de color.

II.1 Marco teórico

II.1.1 Estructura de la LSM

Para interpretar adecuadamente la lengua de señas mexicana se requiere conocer su estructura, debido a que la posición de las manos con respecto al cuerpo puede proporcionar información relevante, no solo es necesario analizar las señas realizadas con las manos sino otras características como lo son el movimiento corporal y las expresiones faciales, debido a que esta clase de señas no gestuales pueden proveer información extra sobre lo que el señante está comunicando.

La lengua de señas como cualquier otro idioma tiene su propia estructura fonológica. Sin embargo, el termino fonología hace referencia a sonidos verbales, como se menciona en el trabajo de Burquest (Burquest, D. A 2009), por consecuencia este concepto no describe adecuadamente el estudio de la lengua de señas. Es por ello por lo que, la fonología de la lengua de señas es conocida como querología y los fonemas como queremas, como se afirma en el trabajo de Torres (Torres S. et al, 2008).

La querología se utiliza para describir los morfemas o unidades mínimas de la lengua de señas como lo mencionan los autores Stark y Aldrete (Stark T. C. S. y Aldrete M. C. 2006),

las cuales en conjunto construyen cada una de las señas del lenguaje; cambiar alguno de estos parámetros en cualquiera de las señas podría cambiar su significado.

Los parámetros fonológicos de la lengua de señas se pueden expresar en seis componentes principales como se lo describen los autores González y Ángeles (González, M. A. R y Ángeles, M. 1992), estos parámetros son los siguientes:

1. Queirema (configuración): Configuración manual de cada seña.
2. Kinema (movimiento): Tipo de movimiento de las manos (circular, zigzag, lineal, etc.)
3. Toponema (ubicación): Ubicación con relación al cuerpo.
4. Kineprosema (dirección): Dirección del movimiento de las manos.
5. Queirotropema (orientación): Orientación de la mano con respecto al cuerpo.
6. Prosoponema (rasgos no manuales): Todos aquellos rasgos que no utilizan las manos, principalmente movimiento corporal y expresiones faciales.

La lengua de señas puede clasificarse por el uso de una o dos manos, esto se conoce como el uso de señas unimanuales o bimanuales respectivamente como se indica en el trabajo de Aldrete (Aldrete M. C. 2009), Cada señante tiene una mano base y una dominante, estas pueden alternarse entre izquierda y derecha y no pierde el sentido de la palabra o el significado, las señas unimanuales se pueden categorizar como estáticas y dinámicas. Las señas estáticas son aquellas que no requieren de movimiento para interpretarse, mientras que las dinámicas sí; las señas bimanuales son dinámicas, es decir, requieren el movimiento de la mano dominante, o incluso de la mano base.

Además, la lengua de señas puede apoyarse de rasgos no manuales, los cuales tratan de expresar sentimientos y sensaciones con el resto del cuerpo. Principalmente con el rostro, para mostrar felicidad, tristeza, repugnancia, intriga, emoción, etc. Estos rasgos también se utilizan para formular interrogantes, ya que como se menciona en el trabajo de Escobedo (Escobedo C. et al, 2017) se requiere fruncir las cejas e inclinar hacia

adelante la cabeza para preguntar algo mientras se hace el señado de la frase, con el objetivo de denotar que la frase es una interrogante.

2.1.2 Redes neuronales

Dentro de los métodos más utilizados para la clasificación automática de señas son aquellos basados en redes neuronales, además de ser muy utilizados, de acuerdo con la revisión del estado arte son los que han aportado una mayor tasa de efectividad en la clasificación. Sin embargo, en su gran mayoría, estos modelos son implementados para señas estáticas, es decir, son señas que no cambian a través del tiempo, y de las cuales cada fotograma representa una única unidad de información, por lo que, para la clasificación de información que cambia en el tiempo se requiere de un modelo distinto de red neuronal.

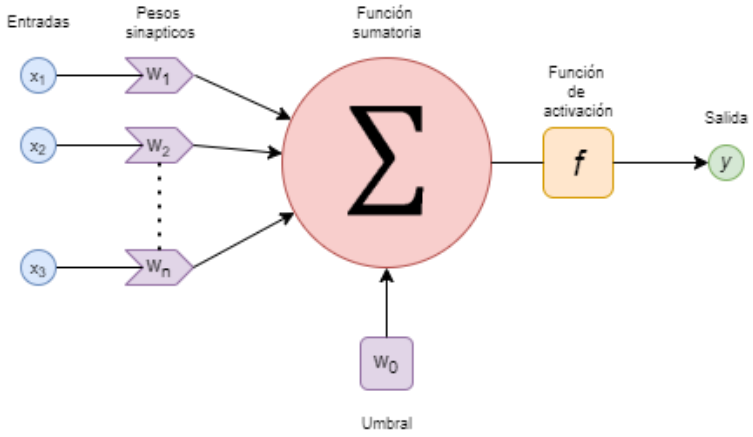
Enseguida se presenta una descripción del modelo de perceptrón simple, lo que es equivalente a una neurona artificial clásica, que es la base de los modelos neuronales existentes.

El modelo matemático-computacional de una neurona artificial recibe este nombre debido a que su estructura fue inspirada en los procesos sinápticos de una neurona biológica, Entre 1950 y 1960 el científico Frank Rosenblatt (Rosenblatt, F. 1985), creó el modelo del perceptrón, tomando base el trabajo de Warren McCulloch y Walter Pitts. Este modelo hace la analogía entre los procesos biológicos de una neurona y los traduce a un modelo matemático que puede ser interpretado por una computadora, su estructura y equivalencia computacional se describe de la siguiente manera:

1. Axón de entrada – Entradas de datos
2. Sinapsis – Pesos de los datos
3. Cuerpo – Función de aprendizaje y umbralización de los datos
4. Cuello del axón – Función de activación
5. Axón de salida – Salida de datos

A continuación, se muestra en la Figura 1 el modelo de una neurona computacional.

Figura 1. Modelo de una neurona computacional.



Fuente: Elaboración propia.

En una red neuronal existe una cantidad $\boldsymbol{i}$ de neuronas, denotadas como N_i y cada una de las neuronas puede recibir n cantidad de entradas simples, que se pueden representar por un vector de entrada $(x_{i1}, x_{i2} \dots x_n)$, cada uno de los valores de entrada tiene un peso sináptico w_i, estos pesos determinan el nivel de importancia del dato que se está analizando.

Junto al conjunto de datos se incluye un umbral $\boldsymbol{Bias}$, (θ_i) el cual suele tener un valor de -1 o 0. Sin embargo, puede variar de acuerdo con la función de activación, además al igual que otra entrada, también tiene un peso asociado definido como w_0,

Los datos de entrada son tratados como un único valor llamado: entrada global (*global input*) y se denota por gin_i, para combinar las entradas simples en una entrada global se utiliza una función de entrada, la cual puede describirse como el producto cruz entre x, w y se puede denotar la ec. (1):

$$gin_i = (x_{i1} \cdot w_{i1}); (x_{i2} \cdot w_{i2}); \dots (x_{in} \cdot w_{in}) \quad (1)$$

Dando como resultado el vector de entrada para el procesamiento de la red neuronal,

El siguiente paso es utilizar la función de sumatoria para procesar los datos de entrada, y se representa por la ec. (2).

$$f(gin_i) = \sum_{j=0}^{n}(x_{ij} \cdot w_{ij}), \forall j \in \mathbb{Z}^+ \quad (2)$$

Posteriormente se evalúa la función de activación, analógicamente una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un "estado de activación.

La función activación calcula el estado de actividad de una neurona, transformando la sumatoria de entradas pesadas en un valor (estado) de activación, cuyo rango normalmente va de $(0\ a\ 1)$ o de $(-1\ a\ 1)$. Esto es así, porque una neurona puede estar totalmente inactiva $(0\ o\ -1)$ o activa (1).

Dentro de los modelos de activación más frecuentes se encuentra la activación lineal, sigmoidea y el tangente hiperbólico ($tanh$).

La función de activación lineal se representa por la ec. (3).

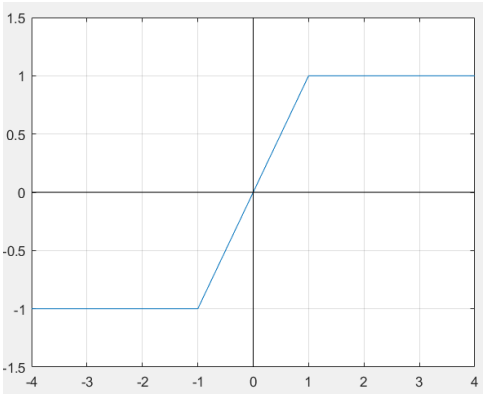
$$f(x) = \begin{cases} -1x \leq -\frac{1}{a} \\ a \cdot x - \frac{1}{a} < x < \frac{1}{a} \\ 1x \geq \frac{1}{a} \end{cases} \quad (3)$$

En donde:

- $x = f(gin_i); a > 0.$
- gin_i es la entrada global
- a es un valor arbitrario para su activación

La representación gráfica de una función de activación lineal se muestra en la Figura 2.

Figura 2: Representación gráfica de la activación lineal.



Fuente: Elaboración propia.

La función de activación sigmoidea es denotada por la expresión dada en la ec. (4):

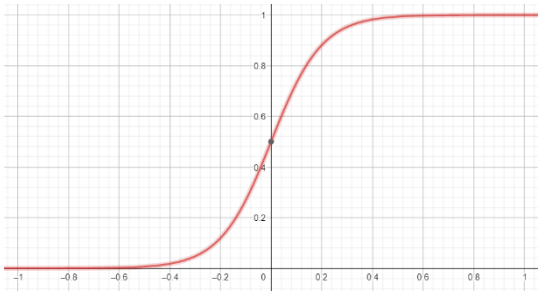
$$\sigma(x) = \frac{1}{1+e^{-gx}} \quad (4)$$

En donde:

- $x = f(gin_i)$
- g es un valor arbitrario que modifica la pendiente de la función de activación

Los valores de salida están comprendidos entre 0 y 1 y están representados gráficamente de la siguiente forma (ver Figura 3):

Figura 3: Representación gráfica de la activación sigmoidea.



Fuente: Elaboración propia.

La función de tangente hiperbólica se denota por la ec. (5):

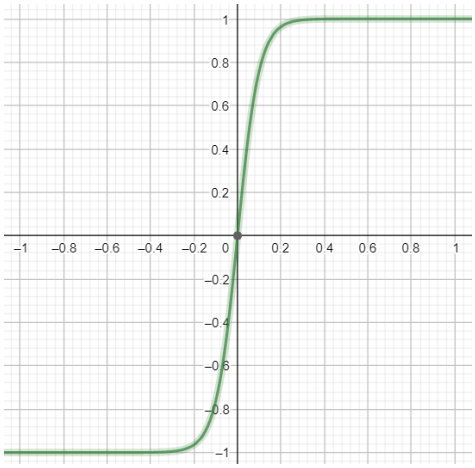
$$tanh(x) = \frac{e^{gx} - e^{-gx}}{e^{gx} + e^{-gx}} \quad (5)$$

Sea

- $x = f(gin_i)$
- g es un valor arbitrario que modifica la pendiente

En este caso los valores de salida están comprendidos entre -1 y 1, y la función de activación se comporta de la siguiente manera, ver Figura 4.

Figura 4: Representación gráfica de la función de tangente hiperbólica.



Fuente: Elaboración propia.

El último componente del modelo neuronal es la función de salida que indica el resultado de cada neurona N_i,esta función establece el valor se transfiere a las neuronas conexas, de tal modo que la salida de una neurona es una de las entradas de otra neurona, si la función de activación está por debajo del umbral determinado θ_i, ninguna salida se transfiere a la neurona subsiguiente.

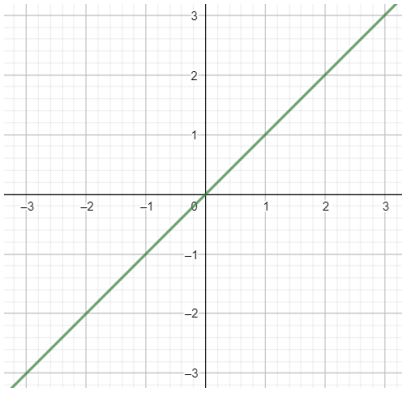
Las funciones de activación más frecuentes son la función identidad, función binaria y la función escalón.

La función identidad es la misma que la función de entrada y se denota por la ec. (6):

$$y = x \tag{6}$$

La representación gráfica de la función identidad se muestra en la Figura 5.

Figura 5: Representación gráfica de la función de salida identidad.



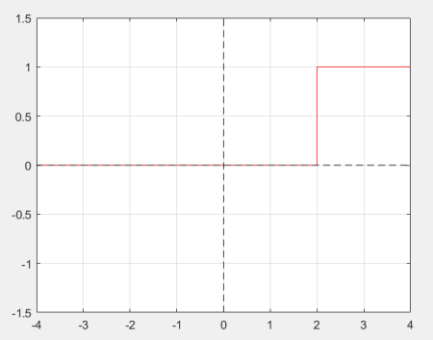
Fuente: Elaboración propia.

La función de salida binaria o umbral solo comprende 2 valores (0 y 1) y este valor está definido por el umbral Θ_i, esta función se expresa por ec. (7):

$$y = \begin{cases} 1, sif(x) \geq \Theta_i \\ 0, sif(x) < \Theta_i \end{cases} \tag{7}$$

La representación gráfica de la función binaria se observa en la Figura 6.

Figura 6: Representación gráfica de la función binaria.



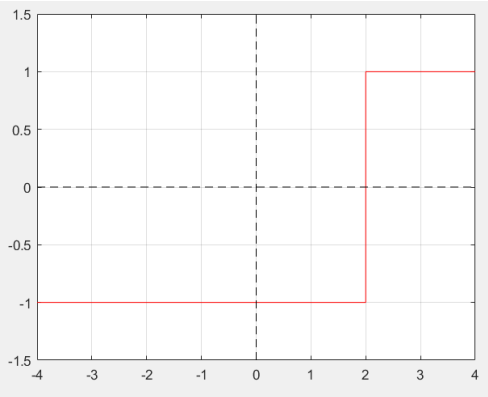
Fuente: Elaboración propia.

La función de activación escalón o signo tiene los valores comprendidos entre -1 y 1 y está representada en la ec. (8):

$$y = \begin{cases} 1, sif(x) \geq \Theta_i \\ -1, sif(x) < \Theta_i \end{cases} \quad (8)$$

La representación gráfica de la función escalón se muestra en la Figura 7.

Figura 7: Representación gráfica de la función escalón.



Fuente: Elaboración propia.

Finalmente, para saber si la neurona aprendió o no, se puede calcular el error restando la salida deseada δ y la salida obtenida y, mediante la ec. (9):

$$error = (\delta - y) \quad (9)$$

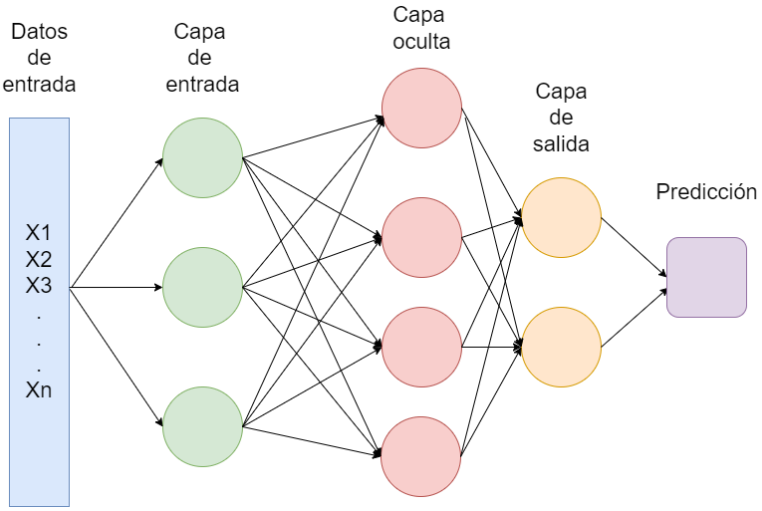
Esto sirve para saber si las neuronas han aprendido correctamente o aún no, de ser necesario se pueden volver a recalcular los pesos w, este proceso de aprendizaje puede utilizar una tasa de aprendizaje α que se encuentra entre 0 y 1 $(0 < \alpha < 1)$. Como se muestra en la ec. (10)

$$w(j)' = \begin{cases} w(j) + \alpha(\delta - y)x(j) \\ w(j) + (\delta - y)x(j) \end{cases} \quad (10)$$

El perceptrón simple está definido por la estructura básica de una red neuronal, en donde se tienen datos de entrada y datos de salida, el perceptrón multicapa o red neuronal está

definido por salidas de perceptrones simples interconectadas a las entradas de otros perceptrones simples como se muestra en la Figura 8.

Figura 8. Estructura de una red neuronal.



Fuente: Elaboración propia.

Como se observa en la Figura 8, en la estructura de una red neuronal multicapa se encuentran cuatro componentes principales: la capa de entrada, la capa oculta, la capa de salida y finalmente la predicción.

Capa de entrada: En esta capa se encuentran los datos que ingresaran a las neuronas para iniciar el proceso de aprendizaje.

Capa oculta: una vez que las primeras neuronas procesan los datos de entrada, sus salidas serán conectadas a las entradas de nuevas neuronas, cabe destacar que pueden tener más de una capa oculta, según el modelo de la red neuronal y su aplicación.

Capa de salida: se encuentran los valores finales de la red neuronal, esta capa puede tener una única salida o varias según el modelo requerido.

Predicción: Es el resultado final de la red neuronal y puede calcularse por diversos métodos, el más común de ellos es simplemente obtener el valor máximo de todas las funciones de salida.

Por otra parte, existen diversos modelos de redes neuronales que permiten realizar la clasificación de datos que varían en el tiempo, el más antiguo de ellos es la Red Neuronal Recurrente (RNN, por sus siglas en inglés). Es un modelo que tiene la misma premisa que una Red Neuronal Artificial, pero añade recurrencia de la capa de salida de la neurona pasada a la capa de entrada de la siguiente neurona, de esta manera la salida de la neurona pasada influye en el compartimiento de la siguiente red.

La función para el cálculo del estado oculto interno se define mediante la ec. (11).

$$h_t = \phi(x_t w_{xh} + h_{t-1} w_{hh} + b_h) \quad (11)$$

En donde:

- h_{t-1} es el estado oculto anterior
- x_t es el dato de entrada
- w_{xh} es un peso asigando para cada valor de x_t
- w_{hh} es un peso asignado para el valor de h_{t-1}
- b_h es el umbral asignado para cada estado oculto

La función de salida para este modelo se expresa mediante la ec. (12).

$$O_t = h_t w_{hq} + b_q \quad (12)$$

Donde:

h_t es el valor oculto actual

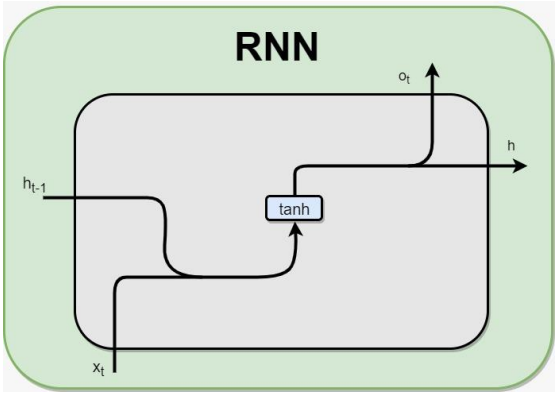
- w_{hq} es un peso asignado para el valor de h_t
- b_q es el umbral asignado para salida

Finalmente, la función de activación utilizada para esta red es una función tangente hiperbólica dada por la ec. (13).

$$y_t = tanh(O_t) \tag{13}$$

La representación de una neurona en una red recurrente se muestra en la Figura 9.

Figura 9. Estructura de una red neuronal recurrente.



Fuente: Elaboración propia.

Como se puede observar en la Figura 9, cada neurona recibe 2 entradas: Estado oculto anterior (h_{t-1}) y entrada de datos (x_t) y entrega 2 salidas: Salida de datos (O_t) y nuevo estado oculto (h).

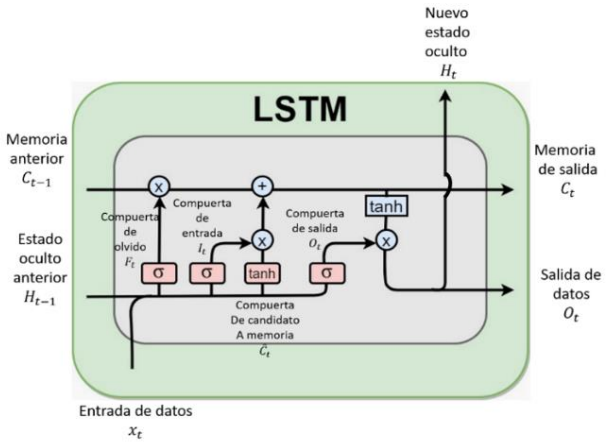
Por otra parte, existen modelos de las redes neuronales recurrentes que tienen modificaciones en su estructura interna, tal es el caso de las Redes de Memoria de Largo-Corto Plazo o LSTM por sus siglas en inglés (Long-Short Term Memory), las cuales conservan la estructura esencial de una Red Neuronal Recurrente, pero cambia el procesamiento interno. Las redes LSTM fueron propuestas por Hochreiter y Schmidhuber en 1997 son capaces de reducir el problema de desvanecimiento de gradiente.

La arquitectura LSTM consta de subredes conectadas recursivamente conocidas como bloques de celdas de memoria. Cada bloque contiene celdas de memoria auto conectadas y unidades multiplicativas que aprenden a abrir y cerrar el acceso al flujo de error constante, lo que permite que las celdas de memoria LSTM almacenen y accedan

a la información durante períodos prolongados. Además, hay puertas de olvido dentro de una red LSTM, que proporcionan instrucciones continuas para operaciones de escritura, lectura y reinicio para las celdas.

La LSTM tiene 3 entradas (ver Figura 10): la entrada de datos (x_t), el acarreo de memoria (C_{t-1}) y la entrada de estado oculto (H_{t-1}) y 3 salidas, salida de datos (O_t), memoria de salida (C_t) y el nuevo estado oculto (H_t).

Figura 10. Representación gráfica de una neurona LSTM.



Fuente: Elaboración propia.

Además, la estructura de una celda LSTM está compuesta por 4 compuertas de activación, cada una de estas compuertas ayuda a tomar una decisión dentro de la red LSTM. El cálculo de estas compuertas internas es muy similar al procesamiento que se tiene en un perceptrón simple, enseguida se describen:

Compuerta de olvido: Ayuda a saber si es necesario seguir recordando la memoria previa o debe desecharse, esta compuerta tiene una activación de tipo sigmoide y se denota por la ec. (14)

$$F_t = \sigma\left(x_t w_{xf} + H_{t-1} w_{hf} + b_f\right) \tag{14}$$

Compuerta de entrada: Ayuda a saber si la información de entrada es relevante para el aprendizaje, esta compuerta tiene una activación de tipo sigmoide y se describe por la ec. (15).

$$I_t = \sigma(x_t w_{xi} + H_{t-1} w_{hi} + b_i) \quad (15)$$

Compuerta de salida: Decide si los datos serán enviados a la siguiente neurona, esta compuerta tiene una activación de tipo sigmoide como se define en la ec. (16).

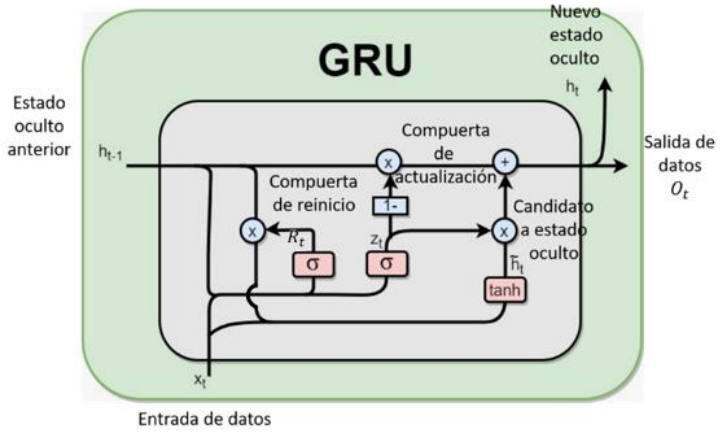
$$O_t = \sigma(x_t w_{xo} + H_{t-1} w_{ho} + b_o) \quad (16)$$

Compuerta candidato a memoria: Decide si los datos se van a almacenar en la nueva memoria, esta compuerta tiene una activación de tipo tangente hiperbólica, como se indica en la ec. (17):

$$\hat{C}_t = tanh(x_t w_{xc} + H_{t-1} w_{hc} + b_c) \quad (17)$$

Por otra parte, existe una variante de la red LSTM, la cual cambia las puertas de activación internas, esta red es llamada Unidad Recurrente Cerrada o GRU por sus siglas en inglés (*gated recurrent unit*) es una red recurrente más reciente creada inicialmente para tareas de traducción automática. Este modelo es similar al LSTM porque puede capturar dependencias a largo plazo. Sin embargo, a diferencia del LSTM, este modelo no requiere celdas de memoria interna, lo que reduce la complejidad. Una unidad GRU combina la puerta de olvido y la puerta de entrada en una sola puerta de actualización; también integra el estado de celda y el estado oculto, resolviendo problemas de estancamiento de mínimos locales y descenso de gradiente.
Una de las principales ventajas del GRU sobre el LSTM es que requiere menos recursos computacionales al tener una estructura menos compleja, la neurona GRU se representa gráficamente como se ilustra en la Figura 11.

Figura 11. Representación gráfica de una neurona LSTM.



Fuente: Elaboración propia.

Una neurona GRU al igual que una RNN tiene 2 entradas y 2 salidas: estado oculto anterior (h_{t-1}) y entrada de datos (x_t) y entrega 2 salidas: salida de datos (O_t) y nuevo estado oculto (h_t); por otra parte, tiene 3 compuertas internas, las cuales se detallan a continuación:

Compuerta de reinicio: Esta compuerta ayuda a olvidar los valores de entrada, en caso de que el entrenamiento así lo requiera y se denota por la ec. (18):

$$R_t = \sigma(x_t w_{xr} + h_{t-1} w_{hr} + b_r) \quad (18)$$

Compuerta de actualización: Esta compuerta decide si se realizará una actualización de los datos, en caso contrario seguirán trabajando con los anteriores y estos pasarán a la salida, descrita por la ec. (19):

$$Z_t = \sigma(x_t w_{xz} + H_{t-1} w_{hz} + b_z) \quad (19)$$

Candidato a estado oculto: Como su nombre lo indica, esta compuerta evalúa si el nuevo estado oculto se entregará a la salida de la neurona, en caso de que no se entregue se seguirá utilizando el estado oculto anterior, se define mediante la ec. (20):

$$\hat{C}_t = \tanh(x_t w_{xh} + (H_{t-1} \cdot R_t) w_{hh} + b_h) \quad (20)$$

III. Hipótesis

La implementación de un sistema basado en una cámara RGB-D y un algoritmo de aprendizaje automático permitirá reconocer un conjunto de señas estáticas y dinámicas de la lengua de señas mexicana.

IV. Objetivos

Objetivo general

- Desarrollar un sistema de reconocimiento de señas mediante una cámara RGB-D y un algoritmo de aprendizaje automático que permita identificar un conjunto de palabras de la lengua de señas mexicana.

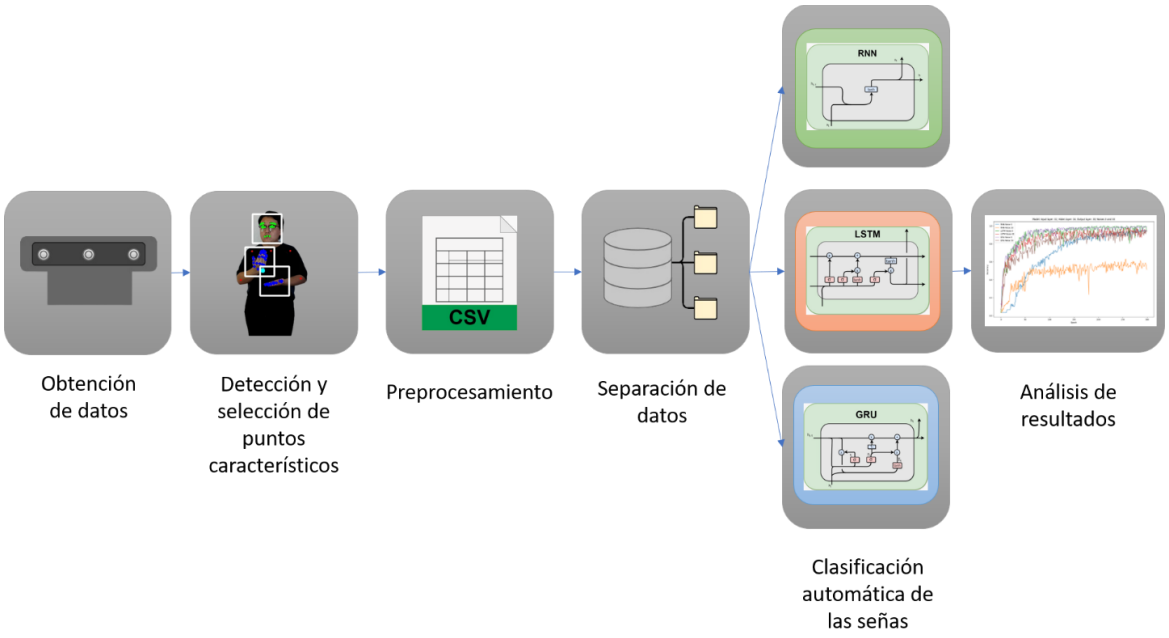
Objetivos específicos

- Recopilar información del estado del arte sobre proyectos similares y sus resultados, además de información sobre algoritmos que se relacionen al propósito del proyecto.
- Realizar un análisis de cámaras RGB-D que puedan entregar información sobre profundidad o mapas de distancias.
- Analizar algoritmos y técnicas para la caracterización y clasificación de la lengua de señas mexicana.
- Implementar un algoritmo para el reconocimiento de la lengua de señas mexicana.

V. Materiales y metodología

En este capítulo se describe de manera detallada cada una de las etapas que llevaron a cabo para la implementación del sistema de reconocimiento automático de la LSM. En la Figura 12 se muestra de manera esquemática la metodología desarrollada.

Figura 12. Metodología desarrollada para el sistema de reconocimiento automático.

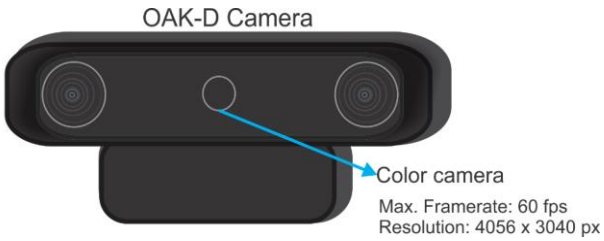


Fuente: Elaboración propia

5.1 Etapa 1. Obtención de los datos

La base de datos utilizada para el desarrollo de este proyecto se obtuvo usando el sensor de profundidad OAK-D (ver Figura 13), este dispositivo consta de 3 cámaras: una cámara central para obtener la información RGB y dos cámaras laterales las cuales permiten medir distancias utilizando la disparidad entre las imágenes. Asimismo, para la captura de los datos e imágenes se utilizó la librería DepthAI (Documentación de DepthAi, 2021).

Figura 13: Cámara de profundidad OAK-D.



Fuente: Mejía-Pérez et al. (2022)

Se adquirieron un total de 30 señas distintas, cuyas características se muestran en la Tabla 1. De estas señas: 4 son estáticas, 26 son dinámicas, 17 son unimanuales y 13 bimanuales, además se pueden clasificar en 4 subgrupos: 8 de ellas son letras del alfabeto dactilológico, 8 son preguntas, 7 son días de la semana y 7 son frases comunes.

Tabla 1: Descripción del cuerpo de los datos.

Tipo de seña	Seña	Estática/ Dinámica	Unimanual/ Bimanual	Simétrico/ Asimétrico	Mano izquierda	Mano derecha
Abecedario dactilológico	A	Estática	Unimanual	Asimétrico	Sin uso	Dominante
	B	Estática	Unimanual	Asimétrico	Sin uso	Dominante
	C	Estática	Unimanual	Asimétrico	Sin uso	Dominante
	D	Estática	Unimanual	Asimétrico	Sin uso	Dominante
	J	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	K	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Q	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	X	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
Preguntas	¿Cómo?	Dinámica	Bimanual	Simétrico	Simultaneo	Simultaneo
	¿Cuándo?	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	¿Cuánto?	Dinámica	Bimanual	Simétrico	Simultaneo	Simultaneo
	¿Dónde?	Dinámica	Bimanual	Asimétrico	Base	Dominante
	¿Para qué?	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	¿Por qué?	Dinámica	Bimanual	Asimétrico	Base	Dominante
	¿Qué es?	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	¿Quién?	Dinámica	Bimanual	Asimétrico	Base	Dominante
Días de la semana	Lunes	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Martes	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Miércoles	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante

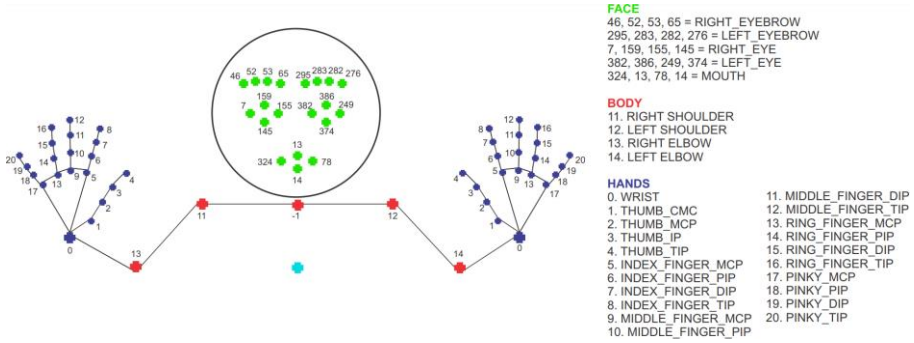
	Jueves	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Viernes	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Sábado	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Domingo	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
Palabras de uso frecuente	Deletrear	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Explicar	Dinámica	Bimanual	Asimétrico	Alternado	Alternado
	Gracias	Dinámica	Bimanual	Asimétrico	Base	Dominante
	Nombre	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	Por favor	Dinámica	Bimanual	Simétrico	Simultaneo	Simultaneo
	Si	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante
	No	Dinámica	Unimanual	Asimétrico	Sin uso	Dominante

Fuente: Mejía-Pérez et al. (2022)

5.2 Etapa 2. Detección y selección de puntos característicos

Para detectar los puntos característicos de la cara, cuerpo y manos empleados al realizar las señas, se hizo uso de la librería MediaPipe (Zhang, F et al, 2020; Singh, A et al, 2021). En este proceso se realizó también la selección de puntos de interés utilizados en cada seña, quedando un total de 67, distribuidos de la siguiente forma: 20 para la cara, 5 para el cuerpo y 20 para cada mano, como se observa en la Figura 14.

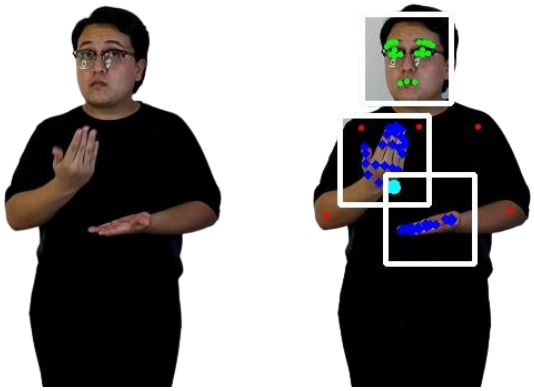
Figura 14: Puntos característicos seleccionados de la cara, el cuerpo y la mano.



Fuente: Mejía-Pérez et al. (2022)

Además, para tener una referencia visual de los puntos de interés capturados, se almacenaron las imágenes de cada una de las secuencias pertenecientes a las señas adquiridas como se muestran en la Figura 15.

Figura 15: Puntos clave de la cara, las manos y el cuerpo para la seña “gracias”.



Fuente: Mejía-Pérez et al. (2022)

Cada punto de interés se representa en la imagen por $P'[X', Y']$, y gracias a las características del sensor es posible obtener su valor de profundidad en Z, de esta forma se para tener su representación en el espacio $P[X, Y, Z]$ se utilizaron las ecuaciones 21 y 22.

$$X = \left(\frac{X'Z}{f}\right) \tag{21}$$

Donde:

X' es la abscisa en el plano imagen, f la distancia focal y Z el valor de profundidad.

$$Y = \left(\frac{Y'Z}{f}\right) \tag{22}$$

Donde:

Y' representa la ordenada en el plano imagen, f la distancia focal y Z el valor de profundidad.

5.3 Etapa 3. Preprocesamiento y almacenamiento de los datos

Los datos capturados se almacenaron en tablas de valores separados por comas (csv), las cuales están estructuradas en 20 filas y 201 columnas, en donde, cada archivo representa una única repetición de una seña individual y cada fila representa la información obtenida en un solo fotograma.

Las filas, por otra parte, están estructuradas con la información obtenida del cuerpo (5 puntos), rostro (20 puntos), mano izquierda, (21 puntos) y la mano derecha (21, puntos), cabe mencionar que cada punto se representa por sus coordenadas (X, Y, Z). Esta información de distancia tiene como unidad de medida el metro, y están medidas respecto al punto central de la toma capturada, además, los datos toman valores negativos en los ejes -X, -Y, -Z. Esto es muy útil cuando se quiere conocer la dirección del movimiento realizado.

5.4 Etapa 4. Clasificación automática de las señas

Después de capturar los datos se dividió el conjunto de datos en tres partes, 70 % para datos de entrenamiento, 15 % para validación y 15 % para pruebas. Se entrenaron los tres modelos neuronales (LSTM, GRU y RNN) descritos teóricamente en la sección 2.1.2.

Como configuración general para los 3 modelos neuronales se utilizó un diseño de 300 épocas y detención temprana, con el fin de evitar un sobreajuste en el entrenamiento de los datos, esto con una paciencia de 100 épocas, se empleó la función de pérdida de entropía cruzada categórica para medir la perdida entre las distribuciones probabilidad y finalmente el optimizador de Adam para reducir el error en la red, mediante las librerías de Keras (Chollet F. et al, 2018) y Tensorflow (Abadi M. et al, 2018).

En la Tabla 2 se muestra las arquitecturas de las redes utilizadas para cada modelo neuronal que se emplearon en este trabajo, distribuidas entre la primera y segunda capa de la red y los parámetros para cada configuración.

Tabla 2: Variaciones en las arquitecturas de cada modelo neuronal utilizado para la clasificación de las señas.

Modelo Neuronal	Capa 1	Capa 2	Parámetros (Miles)
RNN	32	16	8.782
	64	32	21.118
	128	64	56.542

	256	128	170.398
	512	256	570.142
	1024	512	2057.758
LSTM	32	1024	33.60
	64	32	81.502
	128	64	220.318
	256	128	669.982
	512	256	2257.438
	1024	512	8184.862
GRU	32	1024	25.47
	64	32	61..662
	128	64	166.302
	256	128	504.606
	512	256	1697.31
	1024	512	6147.102

Fuente: Mejía-Pérez et al. (2022)

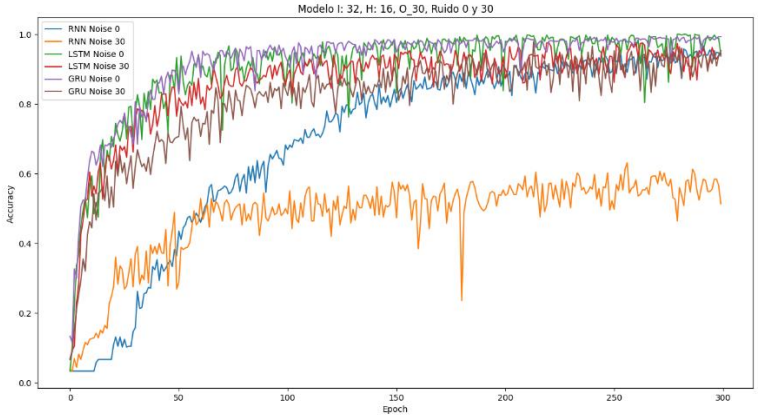
Para el entrenamiento, se agregó la opción de aumentación de datos en línea (*Data augmentation online*) agregando ruido gaussiano a los puntos clave de entrada con una media de cero y una desviación estándar de 30 cm.

Esto produce el efecto de variar aleatoriamente los puntos clave desde la posición detectada simulando diferentes formas de realizar una seña.

El ruido se agrega en cada entrada durante el entrenamiento, generando diferentes valores en cada iteración. Este enfoque también ayuda a reducir el sobreajuste y mejorar la generalización.

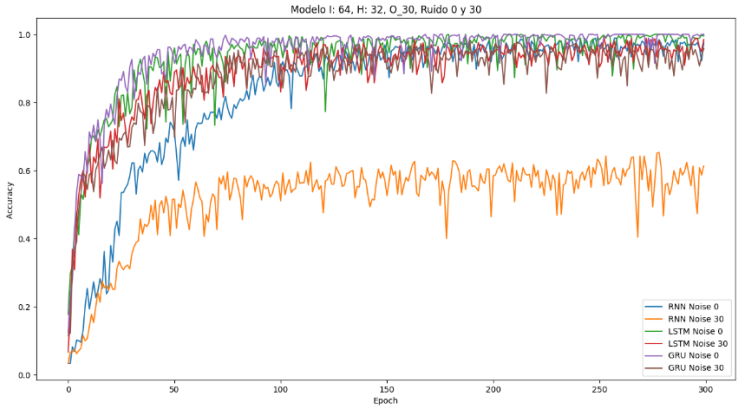
En las Figuras 16 a 21 se muestra la precisión (*Accuracy*) de la validación durante las épocas del entrenamiento para los modelos de redes neuronales con variaciones en la capa de entrada y la capa oculta.

Figura 16: Curvas de entrenamiento para modelos con 32 neuronas en la capa de entrada y 16 neuronas en la capa de salida.



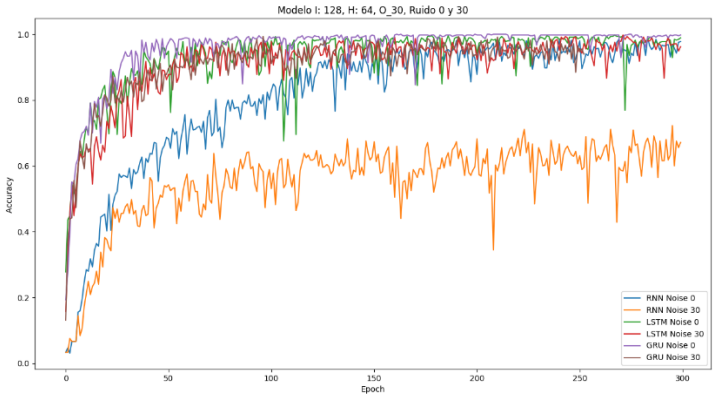
Fuente: Mejía-Pérez et al. (2022)

Figura 17: Curvas de entrenamiento para modelos con 64 neuronas en la capa de entrada y 32 neuronas en la capa de salida.



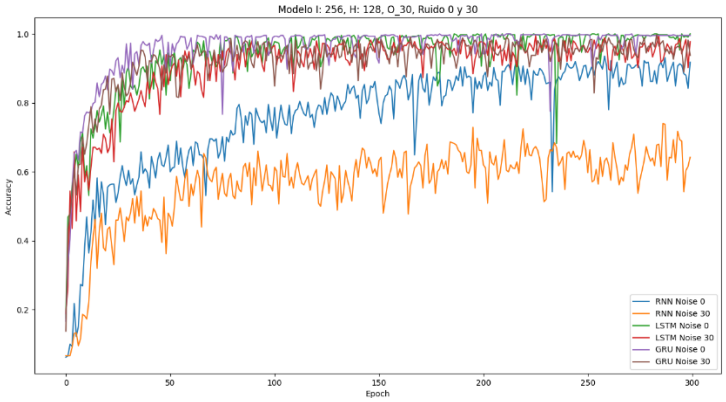
Fuente: Elaboración propia.

Figura 18: Curvas de entrenamiento para modelos con 128 neuronas en la capa de entrada y 64 neuronas en la capa de salida.



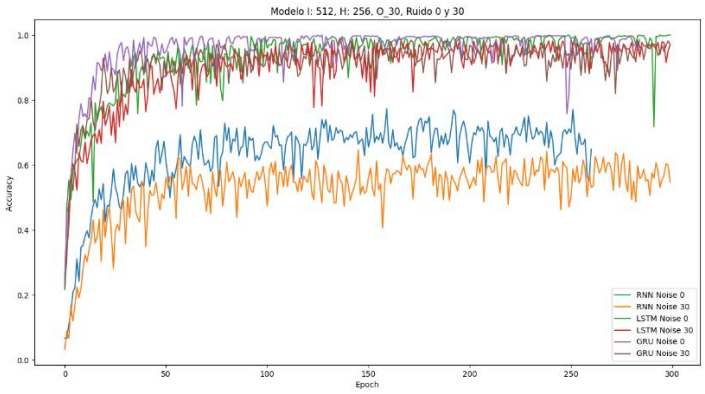
Fuente: Elaboración propia.

Figura 19: Curvas de entrenamiento para modelos con 256 neuronas en la capa de entrada y 128 neuronas en la capa de salida.



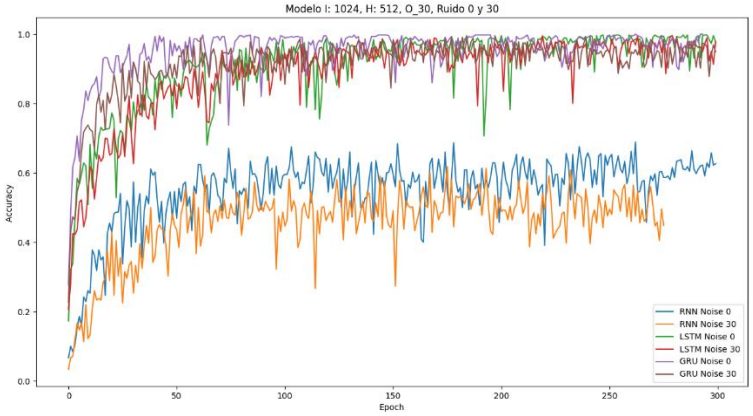
Fuente: Elaboración propia.

Figura 20: Curvas de entrenamiento para modelos con 512 neuronas en la capa de entrada y 256 neuronas en la capa de salida



Fuente: Elaboración propia.

Figura 21: Curvas de entrenamiento para modelos con 1024 neuronas en la capa de entrada y 512 neuronas en la capa de salida.



Fuente: Elaboración propia.

Para evaluar el rendimiento del método de clasificación, se calculó la precisión, la recuperación y la exactitud. Estas métricas se basan en las señas clasificadas correctamente/incorrectamente que se definen con los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN), como se describe en el trabajo de Kuhn (Kuhn 2013):

- Verdaderos positivos (TP) se refiere al número de predicciones en las que el clasificador predice correctamente la clase positiva como positiva.

- Verdaderos negativos (TN) indica el número de predicciones donde el clasificador predice correctamente la clase negativa como negativa.

- Falsos positivos (FP) denota el número de predicciones donde el clasificador predice incorrectamente la clase negativa como positiva.

- Falsos Negativos (FN) expresa el número de predicciones donde el clasificador predice incorrectamente la clase positiva como negativa.

La precisión indica la proporción de identificaciones positivas que fueron realmente correctas. La precisión se calcula con la ec. (23):

$$Precision = \frac{TP}{TP+FP} \tag{23}$$

El recuerdo (recall) representa la proporción de positivos reales identificados correctamente. El recuerdo se calcula con ec. (24).

$$Recall = \frac{TP}{FP+FN} \tag{24}$$

La exactitud (Accuracy) mide la frecuencia con la que las predicciones coinciden con las etiquetas, es decir, el porcentaje de valores predichos que se corresponden con los valores reales. La exactitud se calcula con la ec. (25)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{25}$$

VI. Resultados y discusión

Después de entrenar los modelos enumerados en la Tabla 2, se obtuvieron los resultados que se muestran en la Tabla 3, a partir de estos resultados, se encontró que todas las arquitecturas tienen un sobreajuste cuando se le integran más capas de entrenamiento. Además, el RNN tiende a sobre ajustarse con menos unidades que los modelos LSTM y GRU. GRU entregó la mejor precisión con un 97.11 % para el modelo que comprende 512 unidades en la primera capa y 256 unidades en la segunda capa.

En la Tabla 3 se muestra la exactitud de las pruebas para los tres modelos neuronales utilizados. Los números en negrita representan la mejor precisión de cada arquitectura.

Tabla 3: Porcentaje de exactitud para los modelos neuronales utilizados.

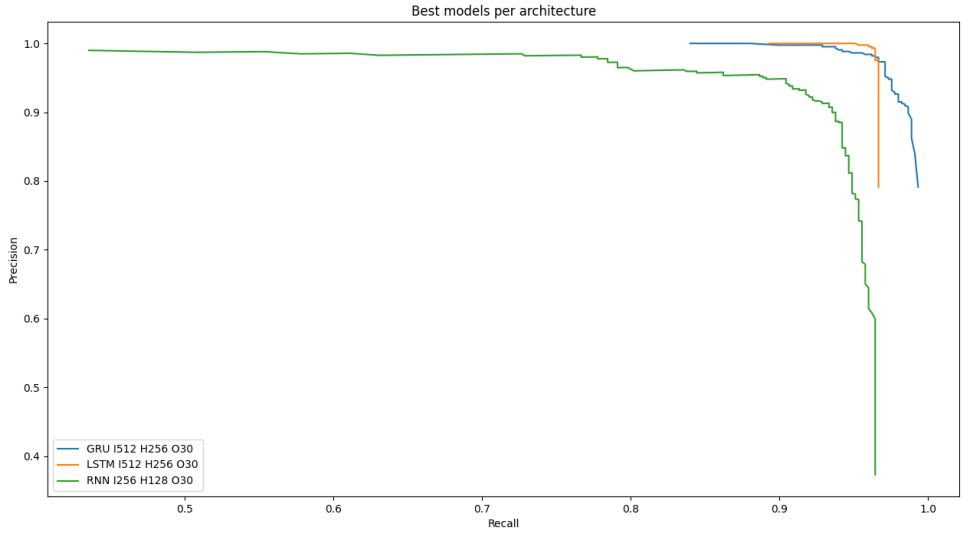
Modelo Neuronal	Capa 1	Capa 2	Porcentaje de exactitud (Accuracy)
RNN	32	16	93.11
	64	32	**94.22**
	128	64	94.0
	256	128	92.44
	512	256	61.55
	1024	512	57.55
LSTM	32	1024	92.44
	64	32	96.44
	128	64	96.22
	256	128	96.44
	512	256	**96.66**
	1024	512	95.77
GRU	32	1024	96.22
	64	32	96.44
	128	64	96.44

	256	128	96.66
	512	256	**97.11**
	1024	512	95.77

Fuente: Mejía-Pérez et al. (2022)

La Figura 22 muestra las curvas de *precision-recall* para las mejores arquitecturas de los modelos neuronales descritos en la Tabla 3. GRU se desempeñó ligeramente mejor que la LSTM con menos complejidad en su arquitectura.

Figura 22. Curvas de *precision – recall* de las mejores arquitecturas obtenidas para cada modelo neuronal.



Fuente: Mejía-Pérez et al. (2022)

6.1 Robustez del sistema al ruido

Para evaluar la robustez del sistema al ruido, se crearon cinco conjuntos de prueba adicionales con diferentes niveles de ruido gaussiano en las coordenadas de los puntos clave. Los niveles de ruido que utilizaron van de cero a 50 cm de desviación estándar en pasos crecientes de 10 cm.

En la Tabla 4 se muestra la precisión de clasificación de la mejor variación del modelo de cada arquitectura evaluada en datos de pruebas con ruido. Los nombres que terminan en *aug* se refieren a un modelo aumentado durante el entrenamiento con ruido gaussiano en las entradas de media cero y desviación estándar de 30 cm como se describe en la Sección 5.4. Cabe mencionar que cada columna representa un conjunto de prueba con un nivel de ruido diferente. El mejor resultado por columna se destaca en negrita.

Tabla 4: Precisión de la clasificación realizada para datos con y sin ruido.

Mejor		Pruebas con ruido				
Modelo	Sin ruido	10 cm	20 cm	30 cm	40 cm	50 cm
RNN	92.44	45.11	45.33	46.44	46.44	46.88
RNN aug	63.55	60.44	58.44	60.0	59.33	59.33
LSTM	96.66	66.22	65.33	63.11	67.77	62.44
LSTM aug	95.55	**89.33**	**90.44**	**89.11**	**90.44**	**88.88**
GRU	**97.11**	48.22	50.66	51.11	46.44	46.66
GRU aug	96.22	69.11	69.33	68.66	68.44	67.33

Fuente: Mejía-Pérez et al. (2022)

De acuerdo con los resultados de la Tabla 4 se puede concluir que el modelo LSTM es más robusto al ruido que los modelos RNN y GRU. Debido a esto, se seleccionó una arquitectura LSTM pequeña con 32 neuronas en la primera capa y 16 neuronas en la segunda capa y se entrenaron múltiples modelos con niveles crecientes de aumento de ruido gaussiano en las entradas que van de 0 a 100 cm de desviación estándar.

En la Tabla 5 se muestra la precisión de estos modelos LSTM evaluados en múltiples niveles de ruido de prueba.

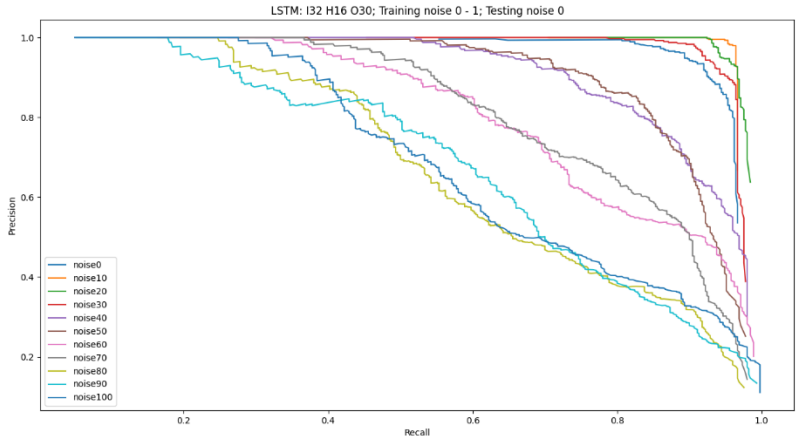
Tabla 5: Precisión de clasificación de múltiples modelos LSTM entrenados con diferentes niveles de aumento de ruido para las entradas evaluadas en conjuntos de prueba ruidosos.

Modelo	Pruebas con ruido					
LSTM	Sin ruido	10 cm	20 cm	30 cm	40 cm	50 cm
0 cm	92.44	62.22	65.33	63.11	67.77	62.66
10 cm	**96.44**	74.22	74.66	74.44	75.33	72.66
20 cm	94.88	84.22	82.44	79.55	83.11	84.22
30 cm	93.11	**86.0**	**87.33**	**85.11**	**87.11**	**87.11**
40 cm	81.55	80.66	81.77	79.55	83.33	82.88
50 cm	82.44	79.77	78.22	79.55	80.66	79.77
60 cm	68.44	68.88	71.77	70.0	71.33	72.44
70 cm	70.66	71.55	71.33	70.0	72.88	69.55
80 cm	59.33	58.22	56.44	57.11	57.55	59.33
90 cm	61.11	57.77	58.22	59.55	58.22	60.22
100 cm	61.33	58.66	60.22	57.55	58.44	60.22

Fuente: Mejía-Pérez et al. (2022)

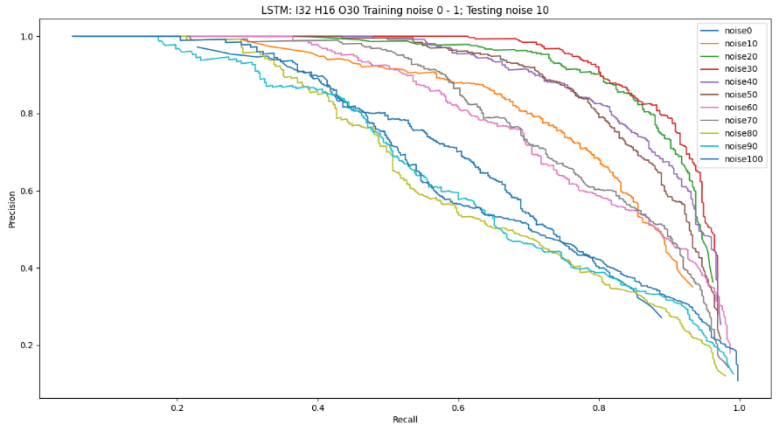
Las Figuras 23 a 28 muestran las curvas de *precision-recall* para los modelos presentados en la Tabla 5 evaluados en un conjunto de prueba con ruido gaussiano de media cero y con variaciones en la desviación estándar.

Figura 23: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, sin ruido gaussiano.



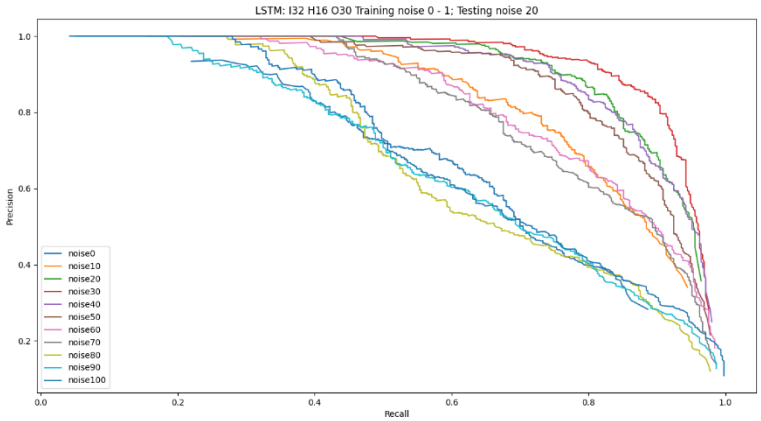
Fuente: Elaboración propia.

Figura 24: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 10 cm.



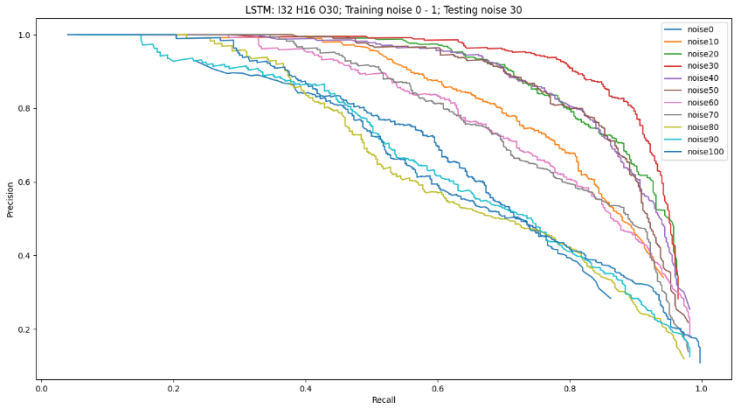
Fuente: Elaboración propia.

Figura 25: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 20 cm.



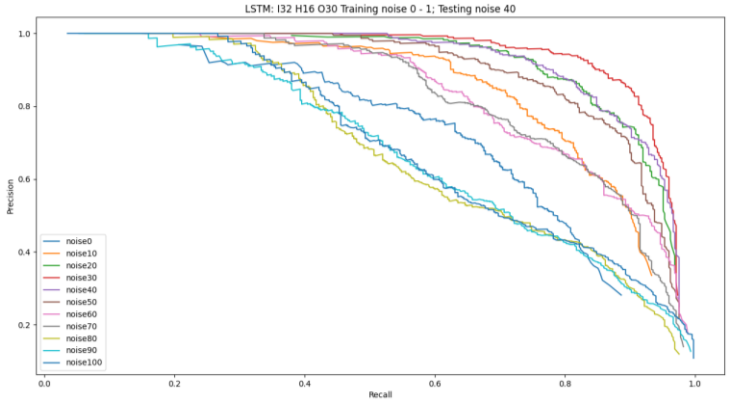
Fuente: Elaboración propia.

Figura 26: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 30 cm.



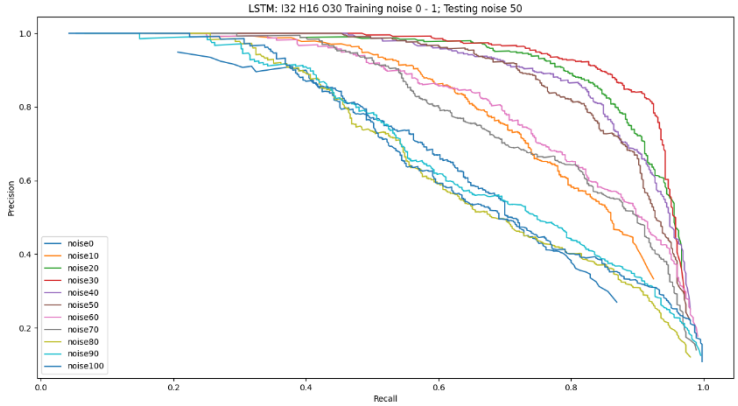
Fuente: Elaboración propia.

Figura 27: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 40 cm.



Fuente: Mejía-Pérez et al. (2022)

Figura 28: Gráfica *precision-recall* para la red neuronal LSTM con 32 neuronas en la capa de entrada, 16 neuronas en la capa oculta, ruido gaussiano con media de 0 y desviación estándar de 50 cm.



Fuente: Elaboración propia.

Como se indica en la Tabla 5, la Figura 27 demuestra que el mejor modelo corresponde al entrenado con ruido gaussiano con una desviación estándar de 30 cm.

6.2 Experimentos de desempeño

Se realizaron además dos experimentos de desempeño: en el primero, se varió la arquitectura del modelo LSTM eliminando y agregando capas y unidades de abandono. En el segundo experimento, se hizo una variación de las entradas al modelo LSTM para determinar cómo contribuye cada conjunto de puntos clave en la clasificación de las señas.

6.2.1 Variando arquitecturas

En este experimento, se evaluó el desempeño de diferentes arquitecturas del modelo LSTM, utilizando una, dos y tres capas y usando aumento y abandono de ruido. En la Tabla 6 se muestran los resultados obtenidos, se puede observar que, en la mayoría de los casos, el modelo de dos capas tiene un mejor desempeño. Los resultados también demuestran que el aumento de ruido más las unidades de abandono recurrentes ayudan en las pruebas con ruido, pero no ayudan en las pruebas sin ruido.

Tabla 6. Variación de los resultados de la arquitectura LSTM.

Numero de capas	Aumento con ruido	Dropout	Pruebas con ruido					
			Sin ruido	10 cm	20 cm	30 cm	40 cm	50 cm
Una capa	No	No	96.44	58.44	60.66	60.0	56.44	56.22
	No	Si	96.44	58.44	57.77	52.22	56.88	56.88
	Si	No	88.22	82.22	81.55	80.88	83.11	82.0
	Si	Si	94.88	88.22	87.77	**89.33**	87.55	88.44
Dos capas	No	No	96.22	34.22	36.22	38.0	35.33	37.11
	No	Si	96.44	39.11	39.33	36.22	37.55	37.11
	Si	No	94.44	87.11	87.77	88.66	88.44	88.66
	Si	Si	95.55	**89.33**	**90.44**	88.44	**89.33**	88.44

Tres capas	No	No	92.0	27.33	30.0	28.88	28.66	24.88
	No	Si	**97.33**	38.0	34.88	34.44	35.77	34.88
	Si	No	94.0	88.66	86.88	84.22	86.22	86.44
	Si	Si	96.66	88.0	88.88	88.88	89.11	**89.11**

Fuente: Mejía-Pérez et al. (2022)

6.2.2 Variando características

En este experimento se probaron diferentes combinaciones de los puntos clave del cuerpo que sirven como entrada al modelo neuronal. Por ejemplo, usar solo las manos, solo el cuerpo, solo la cara y sus combinaciones. La Tabla 7 muestra los resultados obtenidos, en la cual se observa que al utilizar todas las características: manos, cuerpo y rostro, se obtienen mejores resultados cuando los datos de prueba no son ruidosos. Al agregar ruido a las pruebas, el mejor modelo se obtiene al usar solo manos.

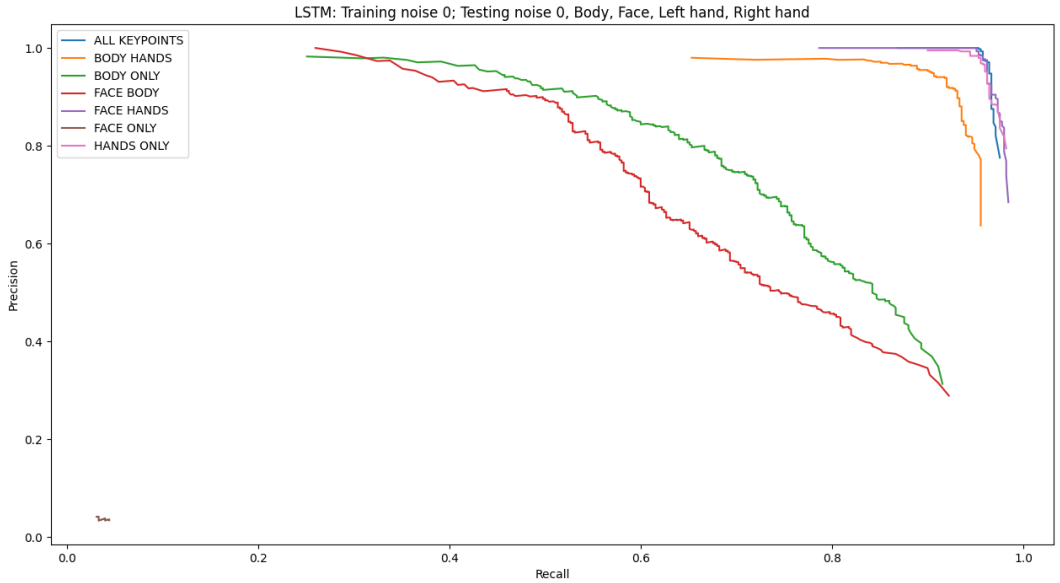
Tabla 7. Porcentaje de precisión del modelo al variar las características de entrada.

Combinación de características	Pruebas con ruido					
	Sin ruido	10 cm	20 cm	30 cm	40 cm	50 cm
Todas las características	**96.44**	64.44	63.55	61.77	63.55	62.44
Solo manos	96.0	**69.11**	**69.33**	**68.66**	**68.44**	66.0
Solo rostro	3.55	5.11	5.55	4.22	5.77	4.0
Solo cuerpo	71.55	8.66	10.44	10.22	10.66	10.0
Cara + cuerpo	63.55	12.88	12.44	12.88	10.88	13.77
Manos + cara	96.22	56.88	58.44	58.0	58.22	58.88
Manos + cuerpo	92.0	65.55	68.22	65.33	66.22	**67.33**

Fuente: Mejía-Pérez et al. (2022)

La Figura 29 muestra las curvas de *precision-recall* para el modelo LSTM al utilizar diferentes combinaciones de características de entrada, evaluadas en los datos de prueba sin ruido. Las curvas muestran que todos los conjuntos que contienen los puntos característicos de las manos obtuvieron resultados similares, como se muestra en la parte superior derecha del gráfico. Por el contrario, cuando el conjunto de características no considera las manos el modelo tiene un rendimiento deficiente. Esto indica que las manos son las características más importantes para el reconocimiento del lenguaje de señas, al utilizar solo los puntos clave de la cara se obtuvieron los peores resultados, como se muestra en la parte inferior izquierda de la gráfica.

Figura 29. Curvas *precision-recall* obtenidas para la combinación de características utilizadas en cada modelo neuronal LSTM.



Fuente: Mejía-Pérez et al. (2022)

Después de comparar las tres arquitecturas seleccionadas para la clasificación de secuencias, se encontró que la arquitectura basada en LSTM funcionó mejor cuando las entradas eran ruidosas. Por otro lado, GRU se desempeñó mejor cuando las entradas no

eran ruidosas. También se demostró mediante el entrenamiento con aumento de ruido gaussiano en las entradas la robustez y generalización del sistema.

Con los experimentos de desempeño se demostró que una red más extensa no era mejor y que el tamaño óptimo consiste en una red de 2 capas con aumento y caída del ruido. El segundo experimento de desempeño mostró que los puntos clave de las manos son las características más importantes para la clasificación de señas con una precisión del 96 %, seguidos por los puntos característicos del cuerpo con una precisión del 72 %.

Los puntos clave faciales por sí solos no son buenas características, se obtuvo una precisión del 4 %. Sin embargo, cuando se combinan con los puntos clave manuales, la precisión aumenta al 96,2 %. El mejor modelo fue el que combinó los tres conjuntos de puntos clave (manos, cuerpo y cara) con una precisión del 96,44 %, lo que demuestra la hipótesis de que las características faciales y corporales juegan un papel importante en el reconocimiento del lenguaje de señas. Este segundo experimento de desempeño también mostró que al agregar ruido gaussiano a los puntos clave faciales afecta la precisión del modelo. El modelo que solo utilizó las manos tuvo una precisión del 69 %, y el modelo que usa las manos en conjunto con la cara tuvo una precisión menor del 57%.

VII. Conclusiones

En este trabajo de investigación se desarrolló un sistema para el reconocimiento del lenguaje de señas utilizando una cámara RGB-D. Se recolectó un conjunto de datos de 30 señas distintas de la Lengua de Señas Mexicana con 100 muestras de cada seña para realizar el entrenamiento, validación y prueba de tres modelos neuronales. Para caracterizar cada seña se extrajeron puntos característicos de las manos, el cuerpo y los rasgos faciales y se realizó la trasformación de estos puntos en coordenadas 3D para entrenar tres clasificadores: redes neuronales recurrentes (RNN), memorias largas a corto plazo (LSTM) y unidades recurrentes cerradas (GRU). Entre los principales

resultados se destacan que la arquitectura LSTM funcionó mejor con entradas ruidosas, mientras que GRU funcionó mejor sin entradas ruidosas y parámetros menos entrenables. Como trabajo futuro se desea ampliar el conjunto de señas e integrar un prototipo que pueda ejecutarse en tiempo real.

VIII. Referencias

Abadi, M., Agarwal, A.,Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., others. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems.

Aldrete, M. C. (2009). Gramática de la lengua de señas mexicana. Estudios de lingüística del español, (28), 1, (2009).

Aldrete, M. C., y Serrano, J. (2018). la comunidad sorda mexicana. vivir entre varias lenguas: LSM, ASL, LSM, español, inglés, maya. Convergencias. Revista de educación, 1(2). http://revistas.uncu.edu.ar/ojs/index.php/convergencias/article/view/1386

Burquest, D. A. (2009). Análisis fonológico. Un enfoque funcional. Dallas: sil International

Cervantes, J., García-Lamont, F., Rodríguez-Mazahua, L., Rendon, A. Y., y Chau, A. L. (2016). Recognition of Mexican sign language from frames in video sequences. In International Conference on Intelligent Computing, pp. 353-362. Springer, Cham.

Chollet, F., others. (2018). Keras: The python deep learning library. Astrophysics source code library, pp. ascl–1806.

Comunicado de prensa núm 24/21 25 de enero de 2021 página 1/3. Org.mx. Recuperado el 11 de febrero de 2022, de https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/EstSociodemo/ResultCenso2020_Nal.pdf

Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad. (2017, 9 junio). Día Nacional de la Lengua de Señas Mexicana

(LSM). Gobierno de México. Recuperado 9 de junio de 2022, de https://www.gob.mx/conadis/articulos/dia-nacional-de-la-lengua-de-senas-mexicana-lsm?idiom=es

DepthAI. DepthAI's Documentation. https://docs.luxonis.com/en/latest/ Online; accessed 31 march 2021

Escobedo C., Mercader C., Pool M., Escobar L., Cruz M., Ramírez M. (2017). Diccionario de lengua de señas mexicana. Ciudad de México: Capital Social por ti https://www.reev.us/pdfs/rivas2019desarrollo.pdf

Galicia, R., Carranza, O., Jiménez, E. D., y Rivera, G. E. (2015). Mexican sign language recognition using movement sensor. IEEE 24th International Symposium on Industrial Electronics (ISIE), pp. 573-578.

González, M. A. R., Ángeles, M. (1992). Lenguaje de signos. Confederación Nacional de Sordos de España.

Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural computation 1997, 9, 1735–1780.

Instituto Nacional de Estadística y Geografía (INEGI), 15 de marzo de 2020. Tabulados Interactivos-Genéricos. Org.mx. Recuperado el 11 de febrero de 2022, de https://www.inegi.org.mx/app/tabulados/interactivos/?pxq=Discapacidad_Discapacidad_02_2c111b6a-6152-40ce-bd39-6fab2c4908e3&idrt=151&opc=t

Kuhn, M.; Johnson, K. (2013). Applied Predictive Modeling; Springer: New York, USA. Volume 26.

Martínez, M., Rojano-Cáceres, J., Bárcenas I.,y Juárez, F. (2016). Identificación de lengua de señas mediante técnicas de procesamiento de imágenes. Res. Comput. Sci., 128, 121-129.

MediaPipe. MediaPipe Holistic. https://google.github.io/mediapipe/solutions/holistic#python-solution352 api. Online; accessed 29 may 2021. 353

Mejía-Pérez, K., Córdova-Esparza, D. M., Terven, J., Herrera-Navarro, A. M., García-Ramírez, T. y Ramírez-Pedraza, A. (2022). Automatic Recognition of Mexican Sign Language Using a Depth Camera and Recurrent Neural Networks. Applied Sciences, 12(11), 5523.

OMS. (2019). Sordera y pérdida de la audición. Recuperado el 25 de septiembre del 2020, de OMS Sitio web: https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss

OMS. (2021). Sordera y pérdida de la audición. Recuperado el 22 de junio del 2022, de OMS Sitio web: https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss

Pablo Bonet, Juan. (1620). Reducción de las letras y arte para enseñar a hablar a los mudos. Madrid, España: Abarca de Angulo, Francisco

Rivas-Perea P. Desarrollo de un intérprete básico del lenguaje de señas para dactilología empleando inteligencia artificial. Tecnológico Nacional De México, Instituto Tecnológico De Nogales, https://www.reev.us/pdfs/rivas2019desarrollo.pdf (2019)

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

Saldaña González, G., Cerezo Sánchez, J., Bustillo Díaz, M. M., y Ata Pérez, A. (2018). Recognition and classification of sign language for spanish. Computación y Sistemas, 22(1), 271-277.

Sánchez Barrera, H. E. (2018). Clasificación del abecedario dactilológico mexicano utilizando minería de datos.

Serafín, M., González, R. (2011). Manos con voz, diccionario de lenguaje de señas mexicana. Consejo Nacional para Prevenir la Discriminación, 15-19.

Singh, A.K.; Kumbhare, V.A.; Arthi, K. (2021). Real-Time Human Pose Detection and Recognition Using MediaPipe. 356 International Conference on Soft Computing and Signal Processing. Springer, pp. 145–154.

Smith-Stark, T. C. (1986). La lengua manual mexicana. Colegio México

Solís, F., Martínez, D., y Espinoza, O. (2016). Automatic mexican sign language recognition using normalized moments and artificial neural networks. Engineering, 8(10), 733-740.

Solís, F., Toxqui, C., y Martínez, D. (2015). Mexican sign language recognition using jacobifourier moments. Engineering, 7(10), 700.

Stark, T. C. S., Aldrete, M. C. (2006). La morfología en la Lengua de Señas mexicana. In Conferencia magistral preparada para el II Congreso Internacional de Logogenia México.

Torres, S., Sánchez, J., Carratalá, P. (2008). Curso de Bimodal. Sistemas Aumentativos de Comunicación, Universidad de Málaga.

Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.L.; Grundmann, M. (2020). Mediapipe 354 hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214 2020. 355