COMPUTATION E INFORMATICA

Recibido 07 Jul 2021 *ReCIBE, Año 10 No. 2, Nov 2021*

Aceptado 12 Aug 2021

Reconocimiento de palabras de la Lengua de Señas Mexicana utilizando información RGB-D

Mexican Sign Language word recognition using RGB-D information

Felipe Trujillo-Romero[1]
fdj.trujillo@ugto.mx

Gibran García-Bautista[2]
ybrran@gmail.com

[1]División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato, Guanajuato, México.

[2] Intabi Company, Orizaba, Veracruz, México.

Resumen. La Lengua de Señas es el principal método alternativo de comunicación entre personas con discapacidad en el habla o en la escucha. Sin embargo, la mayoría de la población que no padece esta discapacidad no la comprende. Esto hace que la comunicación de las personas signantes con su entorno social sea casi imposible. En este trabajo se presenta un avance hacia la construcción de un sistema que pueda traducir palabras de la Lengua de Señas Mexicana mediante el reconocimiento de estas a partir de la trayectoria 3D del movimiento de las manos de signantes usando un sensor Kinect. Se construyó un corpus de 53 palabras considerando solo palabras pertenecientes a once campos semánticos. Con el objetivo de eliminar posibles inconsistencias y ruidos en el patrón extraído se usó la adición de puntos intermedios y el algoritmo KNN fue usado para el filtrado. Además, el método descriptor empleado divide el patrón en dos secciones de acuerdo con la cúspide de su trayectoria y mediante la media aritmética se obtienen las posiciones 3D representativas de ambas secciones. Del patrón general, se obtienen también su anchura, altura, profundidad y orientación. Para la clasificación de las palabras del corpus se usa una Red Neuronal Artificial de tipo Perceptrón Multi Capa. Esta red fue entrenada con el algoritmo de *Backpropagation* y para la validación del sistema reconocedor se realizó utilizando el método *K-Fold Cross Validation*. El porcentaje de precisión media alcanzado por esta implementación fue del 93.46%.

Palabras clave: reconocimiento de patrones, corpus de la lengua de señas, redes neuronales, Lengua de Señas Mexicana.

Abstract. Sign Language is the primary alternative method of communication between people with hearing or speech impairment. However, most of the population that does not suffer from this disability cannot understand or interact with them. Consequently, communication of the signatories with their social environment becomes almost impossible. This paper presents progress towards constructing a system to translate words from the Mexican Sign Language into text, by the signatory's hands motion recognition from a 3D trajectory using a Kinect sensor. A corpus of 53 words was built and considered only words belonging to eleven semantic fields. Intermediate points were added, the KNN algorithm was used for filtering to eliminate possible inconsistencies and noise in the extracted pattern. In addition, the descriptor method used divides the pattern into two sections according to the cusp of its trajectory, and the representative 3D positions of both sections are obtained utilizing the arithmetic mean. From the general pattern, its width, height, depth, and orientation are also obtained. For the classification of the words in the corpus, an Artificial Neural Network of the Multi-Layer Perceptron type is used. This network was trained with the Backpropagation algorithm, and for the validation of the recognizing system, it was performed using the K-Fold Cross Validation method. The percentage of mean precision achieved by this implementation was 93.46%.

Keywords: pattern recognition, sign language corpus, neural networks, Mexican Sign Language.

1. Introducción

De acuerdo con el Censo de Población y Vivienda (2020) realizado por el Instituto Nacional de Estadística y Geografía (INEGI), en México hay 6,179,890 personas con algún tipo de discapacidad, lo que representa 4.9 % de la población total del país. De estos el 22% sufre alguna discapacidad auditiva mientras que el 15% tiene algún problema del habla. Por lo que se estima que una población de alrededor un millón de personas tendría alguna discapacidad auditiva o del habla. Cabe mencionar que una persona puede tener una de las discapacidades o ambas.
La Lengua de Señas (LS) es la principal forma de comunicación de las personas que padecen alguna discapacidad auditiva o en el habla. Sin embargo, esta lengua de señas es desconocida por la mayoría de la población que no padece de alguna de estas discapacidades, incluso por muchos que poseen este tipo de discapacidad. Además de que no existe una LS universal y en un mismo país una misma LS puede variar de una región a otra. Por esa razón diversos investigadores de diversas áreas han tratado de dar solución a esta problemática.

Por ejemplo, en el área computacional el Reconocimiento de la Lengua de Señas (RLS) se ha convertido en un importante estudio de interés que requiere la Interacción Humano Computador (IHC). Sin embargo, lograr obtener un sistema de RLS es una tarea muy compleja debido a que implica tres aspectos fundamentales, que son: 1) el reconocimiento de la mano, 2) de los dedos y 3) de las expresiones faciales. Además, los sistemas de RLS se desarrollan en diversos campos tal como la visión computacional y el reconocimiento de patrones. Podemos decir que existen dos acercamientos principales para tratar con la implementación de un RLS: 1) basado en hardware tal como los guantes de datos y 2) basado en visión computacional. El primero de estos requieren que el signante porte un accesorio como un guante o una pulsera que permita transmitir los movimientos hacia el sistema de procesamiento que los interpretará y dará la salida correspondiente. En el caso de los sistemas implementados usando visión por computadora, se conocen como métodos no invasivos ya que no requieren que el signante utilice algún hardware en especial. Sino que se usa una cámara para capturar la información que será procesada por el sistema interprete. Dicha cámara normalmente en el sistema de procesamiento.

Por lo tanto, el reconocimiento basado en visión computacional es un enfoque más práctico y económico comparado con el basado en hardware, pero también más desafiante. Los principales desafíos son, principalmente, debidos al ruido que causan las condiciones de iluminación del entorno y sobre todo variabilidad en la representación de los símbolos de la LSM por los signantes. Esta es la razón por lo que los dispositivos de captura a utilizar representan un factor importante para alcanzar buena precisión en la implementación de un sistema de RLS.

En los últimos años, el desarrollo de sistemas de RLS basado en sensores RGB-D se ha generalizado. Lo anterior debido a que la información 3D proporcionada por los sensores de profundidad mejora la interactividad, así como la comodidad para el usuario. También se debe tener en cuenta que los precios son cada vez más asequibles para este tipo de dispositivos, como el sensor Kinect (2021) de Microsoft.

El procesamiento de datos tridimensional tiene varias ventajas sobre la información 2D que capturan las cámaras comunes. De inicio el uso de la información 3D permite estimar con mayor exactitud la posición y la orientación del objeto con respecto al sensor de captura. Otro punto importante es que el reconocimiento de objetos 3D tiende a ser más robusto en escenas con ruido. Por ejemplo, donde los objetos que se encuentran al frente obstruyen a los que se encuentran en el fondo. Cabe resaltar el hecho de que el sensor Kinect ha sido usado por una gran cantidad de investigadores en sus trabajos de RLS. Esto debido en gran medida a que el sensor Kinect cuenta con cámara de color, sensor de profundidad, un arreglo de micrófonos y es capaz de ajustarse verticalmente para una mejor captura de la información. Además, posee algoritmos integrados los cuales son capaces de calcular y proveer la posición de hasta 20 articulaciones del cuerpo humano. Para poder acceder a los algoritmos antes mencionados es necesario utilizar el Microsoft Kinect Software Development Kit (SDK, 2021) que Microsoft ofrece para programar su dispositivo. Se debe mencionar que este sensor es relativamente accesible para cualquier investigador e incluso estudiantes que deseen trabajar con este tipo de dispositivos.

Los trabajos realizados para el RLS se pueden clasificar básicamente en 3 enfoques: reconocimiento del alfabeto y números, reconocimiento de palabras y reconocimiento de oraciones. Ejemplos de trabajos donde se realiza el reconocimiento alfanumérico del LS se tiene el desarrollado por Jiménez et al. (2017), quienes usaron características de tipo Harr 3D aplicado en la LSM. Por su parte Pérez et al. (2017) emplearon Momentos de Hu y Lógica difusa para generar un sistema que fuera capaz de reconocer los símbolos estáticos de la LSM. Otro trabajo sobre el reconocimiento de símbolos estáticos lo realizó Carmona-Arrollo et al. (2021), solo que ellos usaron las características invariantes de momentos afines en 3D, como la pose, la posición y la forma entre las manos de los signantes para realizar el reconocimiento. Lahamy y Lichti (2012) también usaron características invariantes, pero en este trabajo fue en dos dimensiones. En el artículo desarrollado por Agarwal y Thakur encontramos un ejemplo en el cual solo se utilizan los simbolos en LS correspondientes a los dígitos. Finalmente, en este rubro, se comentará el trabajo realizado por Luis-Pérez et al., (2011), en el cual se reconocieron los 27 símbolos del alfabeto de LSM pero en este caso se realizó el control de un robot móvil orientado hacia la robótica de servicio.

En el área del reconocimiento de palabras comencemos citando el trabajo de Estrivero-Chavez et al. (2019), quienes usando el sensor Leap Motion (2021) lograron reconocer, además del alfabeto, una decena de palabras en LSM. Por su parte Garcí-Bautista et al. (2017) desarrollaron un sistema que usando un Kinect y el algoritmo de Dynamic Time Warping lograron reconocer 20 palabras de LSM. Tazhigaliyeva et al. (2017) aumentaron a 33 los símbolos que un sistema automático era capaz de reconocer, solo que esta vez fue para el alfabeto Cirilico.

Respecto al reconocimiento de estructuras semánticas más complejas como frases y desde luego sistemas que puedan traducir del LSM a texto o a voz se pueden comentar los trabajos siguientes. Uno de estos desarrollos lo encontramos en Hazari et al. (2017) en cuyo trabajo diseñaron un sistema basa en un kinect para traducción del lenguaje de señas Americano (ASL). Por su parte Chai et al. Implementaron un sistema para la traducción del Lenguaje de Señas Chino (CSL) también usando como dispositivo de captura un Kinect. Ghotkar y Kharate (2015) realizaron un sistema que reconocía el lenguaje de señas Indio (ISL) además de interpretar frases en dicho lenguaje.En el caso de la LSM mencionamos los trabajos de García-Bautista et al. (2016) y Sosa-Jiménez et al. (2017) quienes, cada quien por su parte, realizaron implementaciones orientadas al reconocimiento de palabras y frases simples en LSM orientado hacia el desarrollo de un traductor de LSM a voz o texto.

Como se puede imaginar por los trabajos antes mencionados, el reconocimiento de palabras de la LS implica la combinación de gestos manuales y no manuales involucrando movimiento. Para ello es necesario contar con los métodos que nos permitan realizar dicho reconocimiento. Por esa razón en la literatura podemos encontrar diversos métodos de reconocimiento de patrones que son utilizados para el RLS, como es el caso de las redes neuronales artificiales. Por ejemplo, Molchanov et al. (2015), implementaron un Sistema de reconocimiento de gestos de la mano usando redes neuronales convolucionales. Otro trabajo es el de Oyewole et al. (2018), quienes realizaron un sistema para ayudar a la comunicación entre personas oyentes y aquellos con alguna discapacidad auditiva. Un robot tutor de reconocimiento de señas orientado a la asistencia de niños con discapacidad auditiva o del habla fue desarrollado por Gürpınar et al. (2020). Jiang et al. (2021) haciendo uso de redes neuronales reconocen el esqueleto y hacen el reconocimiento del ASL. Por otro lado Zhang et al. (2019) mediante el uso de redes neuronales desarrollan un sistema en tiempo real para el reconocimiento de la lengua de señas China. Finalmente se comenta el trabajo de Fregos et al. (2021) en cuyo trabajo se usa una red neuronal convolucional y optimización de sistemas de partículas (PSO) tanto para el reconocimiento de la lengua de señas Mexicana como Americana.

Debido a lo comentado en los párrafos anteriores es la motivación que nos mueve para que en este trabajo se presente el reconocimiento de 53 palabras de la LSM a partir del seguimiento de la trayectoria tridimensional de las manos captadas por el sensor Kinect y una red neuronal artificial de tipo Perceptrón Multi Capa y *Backpropagation* como algoritmo de aprendizaje.

Las contribuciones de este trabajo son la que se enumeran a continuación:

1. Creación de un mini-Corpus de la LSM
2. Desarrollo de un sistema que reconoce palabras en LSM.
3. Plantear las bases para la construcción tanto de un corpus de LSM más completo como para un sistema de intérprete y traducción de LSM a voz o a texto.

La estructura del artículo es la siguiente. En la sección 2 se describe a grandes rasgos, lo referente a la lengua de señas mexicana, las características del sensor Kinect, la red neuronal empleada, así como la base de datos usada en este trabajo. Posteriormente, la implementación del sistema de reconocimiento del mini corpus de LSM se presenta en la sección 3. Los resultados obtenidos al realizar las pruebas de *cross-reference* usando 5-folds se muestran en la sección 4. Finalmente, la sección 5 contiene las conclusiones de este trabajo, así como la posible continuación del mismo.

2. Métodos y materiales

En esta sección se abordará la descripción tanto de los algoritmos como de las diferentes herramientas empleadas en la realización de este trabajo. Se iniciará con una descripción referente a la Lengua de Señas Mexicana para situar el contexto. Para continuar con las características del sensor usado para capturar y crear la base de datos usada. Posteriormente, se hablará de la red neuronal que se empleo para realizar el sistema de reconocimiento para finalizar con la mención del Corpus de LSM se creo para realizar el presente trabajo.

2.1 Lengua de Señas Mexicana

La Lengua de Señas Mexicana, se deriva del antiguo lenguaje de signos francés traído a México en 1869. Para ese tiempo, en México ya existían personas sordas que usaban señas para comunicarse, estas señas se fueron incorporando al nuevo lenguaje y se complementaron ampliamente con el sistema francés (Calvo, 2004). Como la mayoría de las lenguas de señas la LSM está compuesta de la dactilología e ideogramas (Serafín y González, 2011).

El alfabeto de la LSM es el que se muestra en la Figura 1 y está formado tanto por señas estáticas como por señas dinámicas. Las señas dinámicas son aquellas cuya trayectoria de movimiento es representada por una flecha color rojo (ver Figura 1). En la ejecución del alfabeto de la LSM se hace uso de una mano base y una mano dominante. Normalmente la mano dominante es la izquierda para personas zurdas y la mano derecha para personas diestras.

La LSM, corresponde al ISO 639-2 sgn-MX (Código estándar Mex-Esp, 2021). El ISO-639-2, es el estándar internacional para los códigos de idioma, cuyo objetivo es establecer los códigos reconocidos internacionalmente para la representación de lenguajes o familias de lenguajes.

Hay que tener en cuenta que como lengua de comunicación la LSM es un lenguaje completo y distinto. Es distinto de otros lenguajes de señas tales como el Lenguaje de Señas Americano (ASL) y distinto del idioma Español (la lengua oral oficial en México).

Se debe de considerar que también existen diferencias en las diferentes lenguas de señas, aunque sea el mismo idioma oral de referencia. Es decir que la lengua de señas no es la misma en todos los países donde se hable español. Por lo tanto, existirá una lengua de señas para México, otra para España, Colombia, Perú, Chile, y así una lengua de señas diferente para país de habla hispana.

También se debe de observar que una cantidad significativa de personas sordas son mayormente monolingües en la LSM. Esto significa que ni el ASL y ni el español son adecuados para una completa comunicación entre la comunidad de sordos de México en ninguna forma, sea por video, por escrito, o por contacto personal.

Otra cosa importante es que si bien es cierto que hay diferencias significativas de la lengua de señas en los países donde se habla español, también hay variaciones dentro de la misma lengua de señas. Algunas variaciones son debidas a expresiones regionales y otras que se deben a las distintas clases de religión, a la edad e inclusive al nivel de educación.

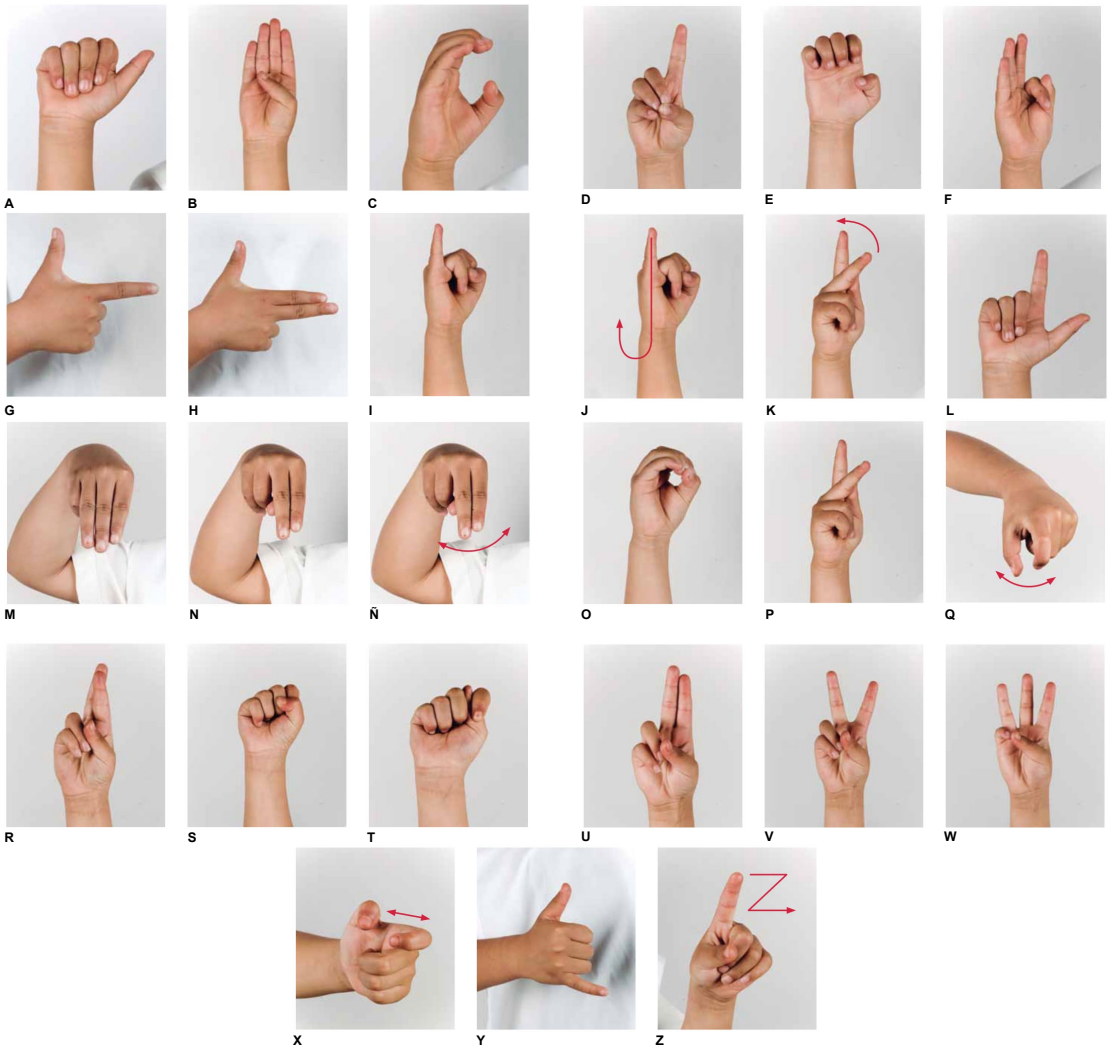


Figura 1 Alfabeto de la LSM (Serafín y González, 2011).

Estas variaciones necesitan ser examinadas más detalladamente lo cual está fuera del alcance de este trabajo. Sin embargo, la mayor parte de la evidencia apunta a un alto grado de similitud, no muy diferente de las variaciones regionales encontradas en las lenguas habladas como el Inglés Americano o el Español Mexicano (Serafín y González, 2011).

2.2. Sensor RGB-D

Un sensor RGB-D, como su nombre lo dice nos permite capturar la información fotométrica en el esquema RGB. Además de poder tener los datos correspondientes a la profundidad con la que se encuentran los objetos en una escena. El sensor Kinect es un dispositivo que pertenece a este tipo de sensores RGB-D. El sensor Kinect fue creado por Microsoft principalmente para ser usado para la consola de video juegos Xbox. Sin embargo, la comunidad científica encontró en él una herramienta de bajo coste para ser usada en diversos proyectos y aplicaciones (Zhang, 2012).

Este sensor al estar compuesto por cámaras y sensores permiten el reconocimiento de movimientos corporales y gestos para facilitar la interacción humano computadora a través de la interfaz natural de usuario (Natural User Interface, NUI) provista por el fabricante.

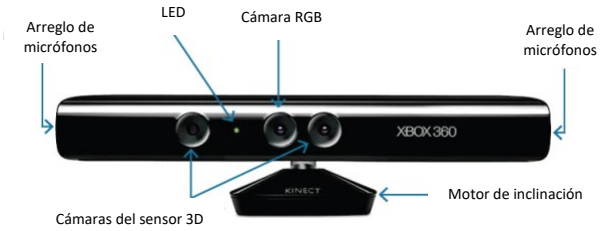


Figura 2 Componentes del dispositivo Kinect™ para Xbox.

Un dispositivo Kinect está integrado por los siguientes componentes: sensor de profundidad, cámara RGB, array de micrófonos y sensor de infrarrojos, que le permiten capturar el esqueleto humano, reconocerlo y posicionarlo en el plano, (Ver figura 2). Actualmente el Kinect cuenta con dos versiones, pero para este trabajo se utilizará la primera versión (V1). En la tabla 1 se pueden observar las características de la versión 1 del sensor Kinect.

Tabla 1 Características del Kinect (Dal Mutto et al., 2012).

Funciones	**Kinect V1**
Cámara de color	640 x 480 @30fps
Cámara de profundidad	320 x 240
Distancia máxima de profundidad	Aproximado de 4.5 mts
Distancia mínima de profundidad	40 cm en modo cercano
Campo de visión horizontal	57 grados
Campo de visión vertical	43 grados
Motor de inclinación	SI
Articulaciones definidas del esqueleto	20 articulaciones
Seguimiento completo del Esqueleto	2
Estándar USB	2.0
S.O soportados	Windows 7, 8 y 10

La captura del dispositivo Kinect inicia cuando las cámaras tanto la RGB como la de infrarrojos captan la información presente en la escena, por ejemplo, a una persona que se encuentra enfrente del dispositivo como en la Figura 3.a. En esta Figura 3.a se puede observar la información RGB del individuo presente. Al mismo tiempo y por medio de un láser se mide la distancia a la cual se encuentra el sujeto. Esto mediante el láser, el cual vuelve al lugar de origen de manera continua como el sonar de un submarino. Como se comentó, el láser solo capta la profundidad a la que se encuentra el sujeto, mientras que la cámara de infrarrojos capta los puntos que se proyectan sobre él. Lo anterior se puede observar en la figura 3.b. A partir de los datos obtenidos por el sensor infrarrojo es posible generar nubes de puntos o mapas de voxeles.

Estos datos constituyen una imagen 3D de la zona en una escala de grises. La combinación de ambas cámaras hace que el software obtenga una escena donde excluye todo lo que no sea el cuerpo, y obtiene la imagen o mapa de profundidad como el que se puede apreciar en la figura 3.c. Este mapa de profundidad es útil ya que es el elemento indispensable para la orientación y la captación de movimiento del cuerpo humano. Y es partir de este que se puede extraer el esqueleto de la persona presente en la escena como se ve en la Figura 3.d.

Este mecanismo tiene unos resultados excelentes en todo tipo de condiciones de iluminación, incluida la oscuridad total y con la misma precisión en usuarios de tez oscura. Para la identificación, el Kinect viene embebido con un sistema de reconocimiento de bípedos llamado *BodyIndex*. Este sistema evalúa varios factores como el movimiento de brazos, el rango de altura, posición de la cabeza, etc. Una vez identificado el cuerpo, el programa crea un objeto virtual independiente de la imagen.

Figura 3. Imágenes capturadas por los sensores del Kinect™.

Para poder acceder a los recursos del sensor Kinect, Microsoft proporciona el kit de desarrollo de software para Windows. Este SDK es un conjunto de controladores, de interfaces de programación de aplicaciones (APIs), interfaces de dispositivo, documentos de instalación y manuales de uso que permite la creación de aplicaciones comerciales y no comerciales para Windows (Kinect SDK, 2021).

2.3. Perceptrón multicapa

Las redes neuronales artificiales (RNA), al margen de “parecerse” a las neuronas biológicas que se encuentran en el cerebro humano, presentan una serie de características propias del mismo. Como, por ejemplo, las RNA son capaces de aprender de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos.

Lo que se considera la primera red neuronal y que se le llamó “Perceptrón” fue desarrollado por Rosenblatt (1957). Este primer perceptrón era capaz de aprender algo y también era robusto, ya que su comportamiento solo se veía afectado si resultaban dañados los componentes que lo conformaban. Este perceptrón es solo una neurona, por lo que su capacidad para aprender y resolver problemas está limitada. Por esa razón se comenzó a desarrollar otros tipos de arquitecturas de redes, pero teniendo como base al perceptrón.

El primer desarrollo para entrenamiento de una red neuronal multicapa fue descrito por Werbos (1974), donde se presentaba un algoritmo de entrenamiento para redes neuronales de contexto general. Sin embargo, fue hasta mediados de los años 80´s, cuando Rumelhart et al. (1986) retomarón la investigación y comenzó a popularizarla en diferentes publicaciones, hasta lo que hoy se conoce como algoritmo *backpropagation*.

El objetivo de entrenar de una red neuronal es conseguir que una aplicación determinada, para un conjunto de entradas, produzca un conjunto de salidas deseadas. El proceso de entrenamiento consiste del ajuste en los pesos de las interconexiones según un procedimiento predeterminado. En cada iteración, los pesos convergen gradualmente hacia los valores que hacen que cada entrada produzca la salida deseada. Los algoritmos de entrenamiento se pueden clasificar en dos grupos: supervisados y no supervisados. Los primeros consisten en presentar un vector de entrada a la red, calcular la salida de la misma, compararla con la salida deseada, y el error o diferencia resultante se utiliza para realimentar la red y cambiar los pesos de acuerdo con un algoritmo que tiende a minimizar el error. Los segundos modifican los pesos de la red de forma que produzcan vectores de salida consistentes, extraen las propiedades estadísticas del conjunto de vectores de entrenamiento y agrupa en clases los vectores similares.

El método de entrenamiento *backpropagation* (entrenamiento hacia atrás), es un sistema de entrenamiento con capas ocultas perfeccionado en la década de los 80"s (Rumelhart et al., 1986). A grandes rasgos, el sistema de entrenamiento mediante *backpropagation* consiste en:

1. Inicializar los pesos de la red de manera aleatoria.
2. Introducir datos de entrada de entre los que se van a usar para el entrenamiento.
3. Dejar que la red genere un vector de datos de salida (propagación hacia adelante).
4. Comparar la salida de la red, con la salida deseada.
5. La diferencia entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos en la capa de salida.
6. El error se propaga hacia atrás (*backpropagation*), hacia la capa de neuronas anterior, y se usa para ajustar los pesos de esa capa.
7. Se continúa propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entrada.

2.3 Corpus de la Lengua de Señas Mexicana

Un corpus es una base de datos que contiene la información necesaria para poder desarrollar herramientas o aplicaciones de reconocimiento de la lengua de señas. El corpus es, entonces, el conjunto de muestras de diferentes palabras en lengua de señas que se procesará y servirá de entrada para el sistema de reconocimiento. Actualmente existen muchos corpus de la LS tal como el Corpus de la Lengua de Señas Británica (BSL) (BSL Corpus, 2021), el Corpus de la Lengua de Señas Alemana (GSL) (GSL Corpus, 2021), el Corpus de la Lengua de Señas Española (LSE Corpus, 2021), el Corpus de la Lengua de Señas Peruana (LSP Corpus, 2021) entre otros. En México existen algunos diccionarios representativos del LSM y estos son el DIELSEME (Calvo, 2004) y Manos con Voz (Serafín y González, 2011). Sin embargo, a pesar de contar con una base rica en información, los diccionarios existentes en México para la LSM no constituyen un corpus y carecen de suficiente información para trabajar en el reconocimiento de la LSM. Cabe mencionar que hasta el presente día en México no existe de manera formal y oficial un Corpus de la Lengua de Señas Mexicana (CLSM).

Por esta razón se procedió a generar la construcción de un pequeño CLSM con el objetivo de tener los datos necesarios para poder lograr un sistema de reconocimiento de un grupo de palabras en LSM específicas.

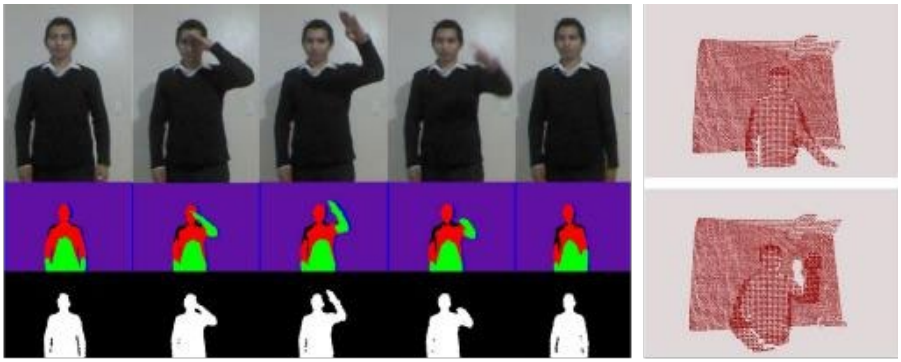


Figura 4 Muestras de color, profundidad y nubes de puntos contenidas en el CLSM.

Para iniciar con la obtención de este corpus, se recolectaron un total de 1590 muestras de 53 palabras interpretadas por 30 personas que utilizan la LSM para comunicarse. Del total de participantes 16 fueron mujeres y 14 hombres, con rango de edad entre 12 y 51 años. En la figura 4, se muestra un ejemplo de la palabra HOLA en LSM que está contenida en el corpus. Los datos que se almacena en este corpus a partir de las palabras de la LSM capturadas contienen la información de profundidad, color, las nubes de puntos, así como las posiciones de 14 articulaciones del esqueleto humano que nos proporciona el sensor Kinect.

En la Tabla 2, se puede observar los 11 campos semánticos seleccionados, así como las palabras en cada uno de ellos, considerando la variabilidad en el movimiento de ambas manos. Además, en la Tabla 2, se presenta el número de muestras de cada seña, las manos que se utilizan para realizar cada seña y como dato adicional una abreviación que se utilizará para referirnos a una seña en particular.

Tabla 2. Palabras que conforman el CLSM utilizado.

Campo Semántico	Palabra	Abreviación	Manos implicadas	No. Muestras
Saludos	Hola	HLA	DER	30
	Adiós	ADS	DER	30
	Gracias	GRA	DER	30
	Nos vemos	NVE	DER	30
	¡Buenos días!	BNS	IZQ	30
Familia	Mamá	MAM	DER	30
	Papá	PAP	DER	30
	Hijo	HJO	DER	30
	Hija	HJA	DER	30
	Familia	FAM	IZQ, DER	30
Pronombres	Yo	YO	DER	30
	Tú	TU	DER	30
	Él/Ella	EL	DER	30
	Nosotros	NOS	DER	30
	Ellos	ELL	DER	30
Lugares	Calle	CAL	IZQ, DER	30
	Edificio	EDI	IZQ, DER	30
	Casa	CAS	IZQ, DER	30
	Baño	BAÑ	DER	30
	Escuela	ESC	DER	30
Alimentos	Agua	AGU	DER	30
	Fruta	FRU	IZQ, DER	30
	Huevo	HUE	IZQ, DER	30
	Café	CAF	IZQ, DER	30
	Galleta	GAL	DER	30
Preguntas	¿Cómo estás?	CME	IZQ, DER	30
	¿Qué haces?	QHA	DER	30
	¿Cuál?	CUA	IZQ, DER	30
Cocina	Comida	COM	IZQ, DER	30
	Cuchara	CUC	IZQ, DER	30
	Mesa	MES	IZQ, DER	30
	Plato	PLA	IZQ, DER	30
	Silla	SIL	IZQ, DER	30
Colores	Blanco	BLA	IZQ, DER	30
	Negro	NEG	IZQ, DER	30
	Amarillo	AMA	IZQ, DER	30
	Azul	AZU	IZQ, DER	30
	Rojo	ROJ	DER	30
Estados de Ánimo	Aburrido	ABU	IZQ, DER	30
	Enojado	ENO	DER	30
	Feliz	FEL	DER	30
	Miedo	MIE	DER	30
	Triste	TRI	DER	30
Medios de Transporte	Avión	AVI	IZQ, DER	30
	Automóvil	AUT	IZQ, DER	30
	Taxi	TAX	IZQ, DER	30
	Autobús	AUB	IZQ, DER	30
	Motocicleta	MOT	IZQ, DER	30
Misceláneos	Bien	BIE	DER	30
	Mal	MAL	DER	30
	Trabajo	TRA	IZQ, DER	30
	Problemas	PRO	IZQ, DER	30
	Abrazos	ABR	IZQ, DER	30

3. Implementación del sistema de reconocimiento

El proceso realizado en este trabajo se muestra en la Figura 5, y comienza con la captura de los datos que conformarán el corpus de la LSM. Posteriormente, el patrón de la trayectoria de la mano es reconstruido y pre-procesado. A partir del nuevo patrón se genera el vector de características que se introduce a la red neuronal. Finalmente el método *K-Fold Cross Validation* es usado para el entrenamiento y la validación de la red neuronal.

Para poder realizar el sistema de reconocimiento usando el corpus de LSM generado se requiere de procesar previamente los datos de este para extraer las características que nos interesan y que van a servir para el entrenamiento del sistema y la subsecuente evaluación del reconocedor de palabras de LSM. De esta manera entonces vamos a revisar la implementación del sistema de reconocimiento de LSM empezando por el acondicionamiento de los datos del corpus.

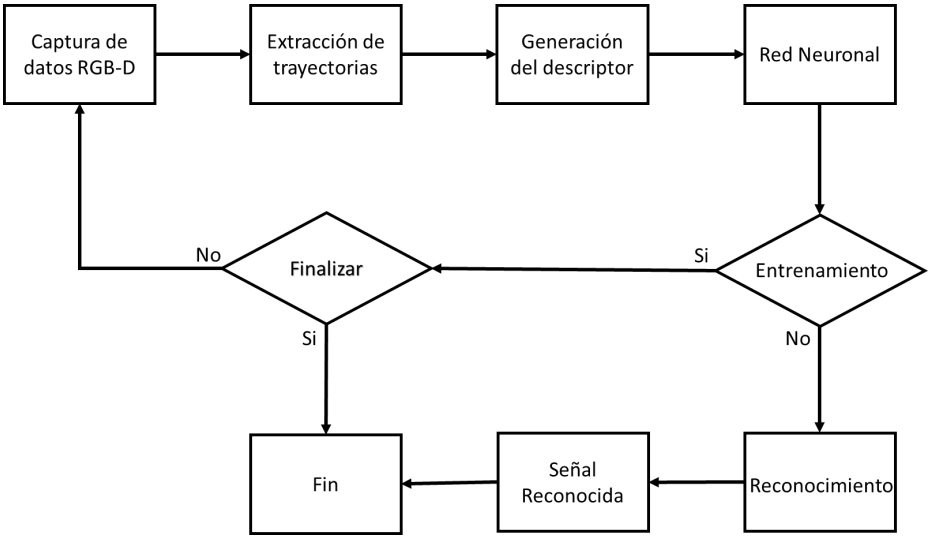


Figura 5 Proceso del sistema reconocedor de palabras de la LSM.

3.1 Acondicionamiento del corpus de LSM

En este paso se hará uso de un filtro pasa-banda con la finalidad de eliminar toda información que no pertenezca al signante. Además, la aplicación de este filtro pasa-banda nos permitirá realizar una reducción del tamaño de los datos lo cual hará que se procesen más rápido.

El funcionamiento de este filtro consiste en remover todos los puntos que se encuentren fuera de un rango dado para una dimensión específica. Considerando que todas las muestras fueron capturadas en un entorno controlado con respecto a la distancia entre el signante y el Kinect de alrededor de 1.7 metros con la posición del usuario se encuentra centrada respecto al sensor. Además, el sensor está colocado a una altura de 1.3 metros. Por tal razón se consideró factible aplicar este filtro en base a un umbral de 1.9 metros respecto al eje central del sensor.

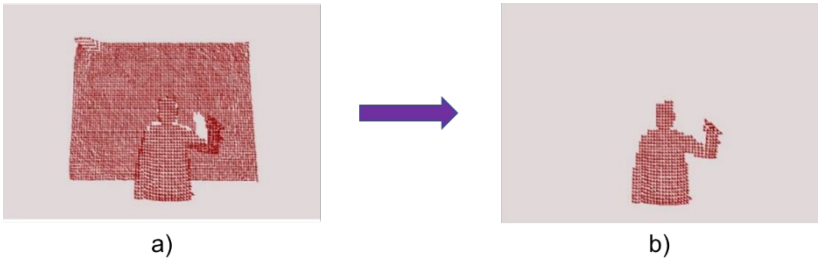


Figura 6. a) Datos de entrada, b) datos filtrados

La figura 6a) muestra la nube de puntos que se obtiene directamente del sensor Kinect mediante el uso del SDK que proporciona Microsoft. Mientras que en la Figura 6b) se puede observar la nube de puntos ya filtrada.

Figura 7. Segmentación de la mano

Continuando con el procesamiento de la información, lo que nos interesa es localizar las manos del signante para poder seguir su trayectoria. Por lo tanto, el movimiento de la mano durante la ejecución de una palabra de la LSM involucra una trayectoria en un plano tridimensional. Así que es necesario realizar una segunda etapa de filtrado, pero esta vez para reconocer y segmentar la mano del signante con la finalidad de seguir y trazar solo la trayectoria de esta parte del cuerpo. La figura 7a) muestra la nube de puntos después de quitar la información que no pertenece al signante. A partir de esta información se segmenta la mano para poder seguir su trayectoria. La figura 7b) se puede observar la mano segmentada de la nube de puntos de la figura 6a).

Una vez ubicada y segmentada la mano haremos el seguimiento con ayuda del sensor Kinect, el cual nos permite almacenar esta trayectoria como un conjunto ordenado de posiciones 3D también llamados puntos (ver Figura 8a). Tanto el número de puntos como la precisión en el cálculo de las posiciones dadas por el Kinect SDK están directamente ligados a la velocidad con la que un gesto es ejecutado frente al sensor y a la respuesta de este a los cambios en la entrada. Cabe mencionar que un error en la precisión del cálculo de una posición puede generar un punto fuera de contexto y ocasionar ruido.

3.2 Reconstrucción del nuevo patrón

Por lo tanto, dada la variabilidad en el número de puntos de una muestra a otra, la posición de captura respecto al sensor y el posible ruido existente es necesario generalizar y simplificar el patrón de la trayectoria de la mano. Para ello se realizó una etapa de reconstrucción del patrón de puntos, en el cual se incluyeron las fases de adición de nuevos puntos intermedios y filtrado de puntos.

La adición de nuevos puntos intermedios busca dar mayor consistencia al patrón de la trayectoria al completar espacios vacíos formados por la rápida ejecución del signante y evitar que el método de filtrado elimine algún dato importante. Decimos que un punto nuevo está conformado de la siguiente manera:

$$p_{nuevo} = (x_{nuevo}, y_{nuevo}, z_{nuevo}) \quad (1)$$

Donde:

x_{nuevo} es la nueva posición en el eje de la absisa,

y_{nuevo} es la nueva posición en el eje de la ordenada y

z_{nuevo} es la nueva posición en el eje de la cota.

El cálculo de una nueva posición para cada eje es dado por la ecuación (2), considerando al patrón de la trayectoria como un arreglo ordenado donde:

pos_{nueva} = el valor de la nueva posición,

pos_{actual}= es el valor de la posición actual y

$pos_{destino}$= es el valor de la posición siguiente.

$$pos_{nueva} = \min(pos_{actual}, pos_{destino}) + \frac{\sqrt{\left(\frac{pos_{actual} - pos_{destino}}{2}\right)^2}}{2} \quad (2)$$

Este método es utilizado para calcular las nuevas posiciones de x, y, z. Este proceso fue iterado 5 veces para dar un aspecto uniforme en el espacio entre cada punto.

Con la finalidad de eliminar aquellos puntos que se encuentran fuera de contexto y que pueden ocasionar ruido se aplica el algoritmo de K-vecinos más cercanos (KNN). Utilizando el SDK del Kinect se convirtieron las posiciones dadas por el Kinect a posiciones en pixeles de una imagen de 640x480 pixeles. De esta forma el algoritmo KNN busco los puntos que en un espacio de K=4 no tuvieran ninguna conexión y de esta forma proceder a su eliminación.

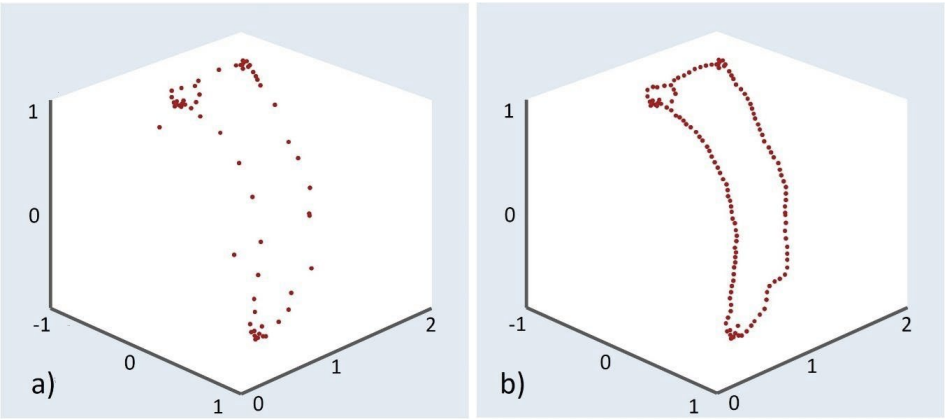


Figura 8. a) Patrón de puntos de la palabra **Hola** y b) nuevo patrón generado: filtrado y adición.

A partir del nuevo patrón reconstruido se realizó el cálculo de la Distancia Euclidiana usando la ecuación 3, en cada punto respecto al punto inicial y el etiquetamiento de este valor a su punto correspondiente. En la figura 9a) se puede ver las distancias calculadas para un punto en particular.

Decimos que $P = \{p_1, \cdots, p_n\}$ es el conjunto de posiciones de la trayectoria de la mano ejecutadas en un plano tridimensional. Sea q la posición inicial de la mano. Para cada posición p_i se calcula la distancia Euclidiana respecto a q.

$$d = \sqrt{(p_x - q_x)^2 + \left(p_y - q_y\right)^2 + (p_z - q_z)^2} \tag{3}$$

Posteriormente, se realizó el redondeo de cada distancia a su valor entero más cercano para poder después reducir el tamaño del patrón de acuerdo a la eliminación de los números repetidos secuencialmente dejando un solo valor (véase Figura 9b).

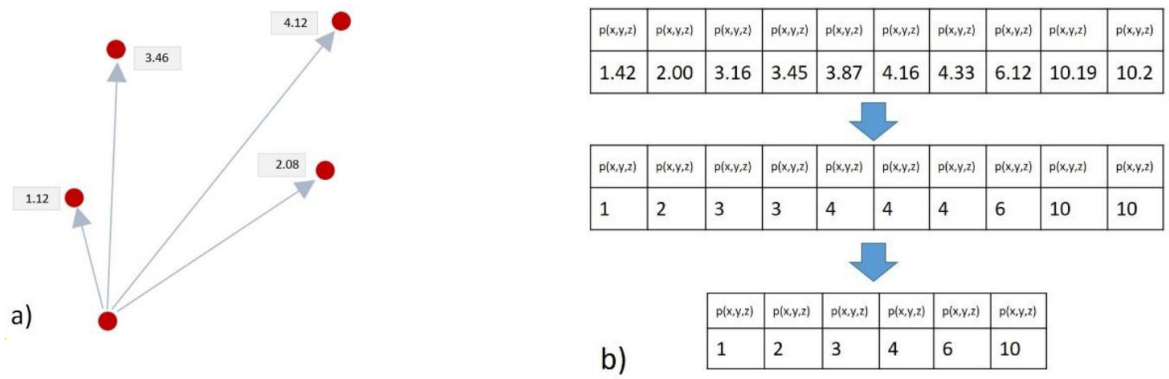


Figura 9. a) Distancia Euclidiana de cada punto respecto a su punto inicial y b) redondeo y reducción del patrón de puntos.

A partir de los datos reducidos se describe el patrón de movimiento de acuerdo a su geometría y a su trayectoria. Para describir al patrón de acuerdo a su geometría se consideró la anchura (4), altura (5), profundidad (6) y orientación del patrón (7).

$$anchura = x_{max} - x_{min} \tag{4}$$

$$altura = y_{max} - y_{min} \tag{5}$$

$$profundidad = z_{max} - z_{min} \tag{6}$$

$$\overrightarrow{orientacion} = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2 + (z_{max} - z_{min})^2} \tag{7}$$

Donde:

$x_{max} =$
$y_{max} =$
$z_{max} =$
$x_{min} =$
$y_{min} =$
$z_{min} =$
Se calculan estos valores tanto para la mano izquierda como para la mano derecha.

De acuerdo a la observación hecha durante la realización de este trabajo se pudo constatar que las palabras que hemos usado y además un gran número de palabras de la LSM culminan en la posición más alta de su ejecución para después volver a la posición inicial velozmente. Por lo tanto se realizó la descripción de la trayectoria de la mano basada en la división del patrón en dos secciones respecto a la distancia Euclidiana máxima alcanzada. En la figura 10, se puede observar el procedimiento mediante el cual se obtiene la media aritmética (7) de las distancias Euclidianas y de las profundidades por cada sección y de todo el patrón. Este procedimiento se repite para el patrón de la mano izquierda y derecha.

$$\mu = \frac{1}{n}\sum_{i-1}^{n} d_i = \frac{d_1 + d_2 + \cdots + d_n}{n} \tag{8}$$

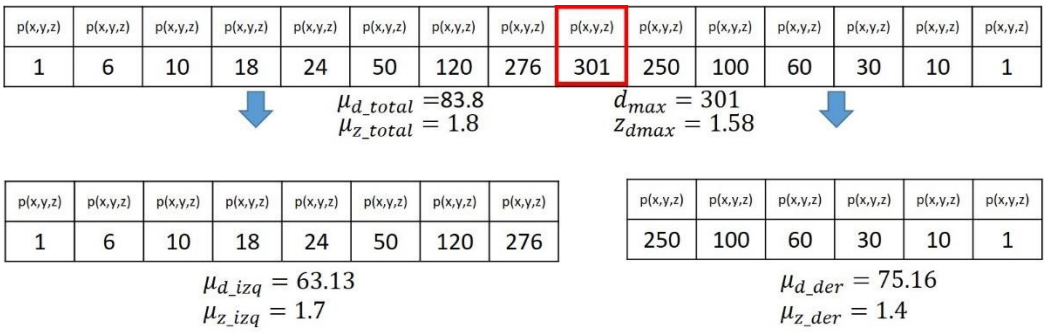


Figura 10. Representación del procedimiento para dividir el patrón y obtener 8 elementos característicos.

Finalmente, se crea un vector de 12 elementos mediante la concatenación de las características de altura, anchura, profundidad, ángulo, μ_{d_izq}, μ_{z_izq}, μ_{d_total}, μ_{z_total}, d_{max}, z_{d_max}, μ_{d_der} y μ_{z_der}. Entonces se forma un vector de 12 elementos por cada mano para generar un arreglo con un total de 24 características. Este arreglo es el que sirve de entrada a la red neuronal tanto para el aprendizaje de las señas del corpus como para su posterior reconocimiento de estas.

3.3. Sistema de Reconocimiento

Como se ha comentado en párrafos anteriores, el método de reconocimiento presentado en este trabajo utiliza una red neuronal de tipo perceptrón multicapa y *Backpropagation* como método de entrenamiento.

Si bien es cierto que se usó Backpropagation para el entrenamiento de la red, no fue el algoritmo base el utilizado sino una de las variantes que existen para este algoritmo. En este caso se utilizó la variante que hace uso de la función *Levenberg-Marquardt* para dicho entrenamiento. Además de la función del Gradiente Descendiente para el ajuste de los pesos de la red y el error cuadrático medio para evaluar su desempeño.

La ventaja de usar dicha variante de *Levenberg-Marquardt* es, principalmente, una convergencia más rápida del algoritmo de entrenamiento. Sin embargo, esta rapidez en la convergencia tiene un alto costo computacional ya que se requieren más recursos para su implementación. Como por ejemplo se requiere más memoria y la capacidad de cómputo del sistema en el cual se realiza el entrenamiento.

Tabla 3. Parámetros de la Red neuronal.

Parámetro	Valor
Tipo de red	MLP
Algoritmo de Entrenamiento	Backpropagation
Variante	*Levenberg-Marquardt*
Num. Capas ocultas	3
Razón de aprendizaje,	0.2
Num. Épocas Máximo	10000
Error mínimo	1e-10

La arquitectura de esta red neuronal consiste en una capa de entrada, 3 capas intermedias con 10, 30, y 10 neuronas respectivamente y una capa de salida. Además de la cantidad de capas y de neuronas por capa se consideran otros parámetros que se presentan en la tabla 3.

4. Resultados

En esta sección se presentan los resultados obtenidos a partir de las pruebas realizadas. Una vez validado el algoritmo de entrenamiento se procedió a entrenar el sistema de reconocimiento mediante el método *KFold Cross Validation (KFCR)*. Este método de validación cruzada permite evaluar el desempeño y la robustez del sistema de reconocimiento implementado. Lo anterior al evaluar el sistema mediante conjuntos de entrenamiento tomados al azar y validando con las imágenes restantes.

Entonces, si se considera que solo se cuenta con 30 muestras de cada palabra se debe de asignar un valor de *kfold* máximo. En este caso se asignó un valor máximo de $K = 6$, dado que el número de muestras por palabra es de 30. Esto significa que se realizaron 6 pruebas diferentes de K_1 hasta K_6 y en cada una de ellas se tomaron 5 imágenes de las 30 muestras de cada seña en el corpus. Cada grupo de 5 imágenes era diferente para cada *kfold*. Por lo tanto, en cada iteración se tomaron 265 muestras considerando 5 de cada palabra. Estas muestras fueron separadas para la validación y las otras 1,325 muestras (25 de cada palabra) se utilizaron para el entrenamiento.

Tabla 3. Precisión obtenida en las pruebas realizadas

Fold	Precisión
K_1	94.34%
K_2	93.96%
K_3	98.87%
K_4	90.19%
K_5	92.83%
K_6	90.57%

En la gráfica de la Figura 11, se muestran los valores de reconocimiento promedio obtenidos después de la ejecución de cada uno de los 6 experimentos uno para cada uno de los *6-folds*. En esta Figura 11, cada barra representa el porcentaje promedio obtenido para cada una de las palabras de LSM evaluadas.

La tasa de reconocimiento obtenida durante cada iteración o para cada *kfold* se muestra en la tabla 3, aquí es posible observar que los porcentajes de reconocimiento obtenido para cada *kfold* no es muy diferente y que oscilan entre 90% y 99%. Al calcular la desviación estándar a partir de los resultados generados al evaluar cada *kfold* se obtiene una desviación de 0.03153, lo cual es una dispersión de los datos muy pequeña. Esto representa que el sistema de reconocimiento implementado es robusto ya que no hay gran variación entre los resultados obtenidos en cada uno de los experimentos realizados. Cabe resaltar el hecho de que en cada experimento se realizó la evaluación del sistema con imágenes diferentes. Es decir, en cada experimento se usaron imágenes de prueba diferente del anterior.

La precisión del clasificador en cada *kfold* es determinada por la ecuación 9, donde:

TP son los verdaderos positivos y

FP son los falsos positivos.

Los verdaderos positivos son los patrones clasificados correctamente y los falsos positivos los patrones clasificados de manera incorrecta. Finalmente se obtuvo una precisión media del 93.46% la cual fue determinada por la ecuación 10.

$$precisionK_n(\%) = \frac{TP}{(TP + FP)} * 100 \tag{9}$$

$$precision\ media(\%) = \frac{1}{n}\sum_{i-1}^{n} K_i = \frac{K_1 + K_2 + \cdots + K_n}{n} \tag{10}$$

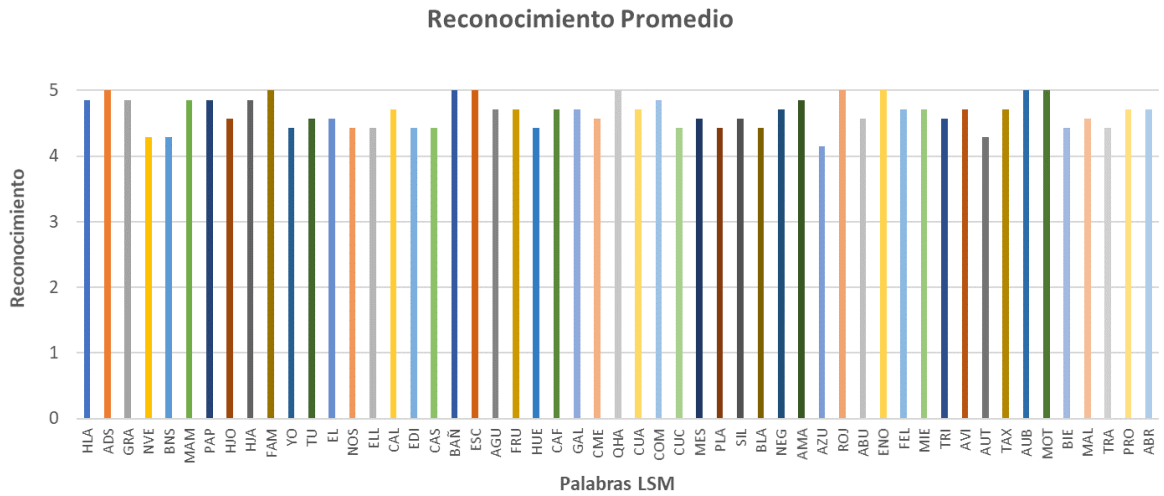
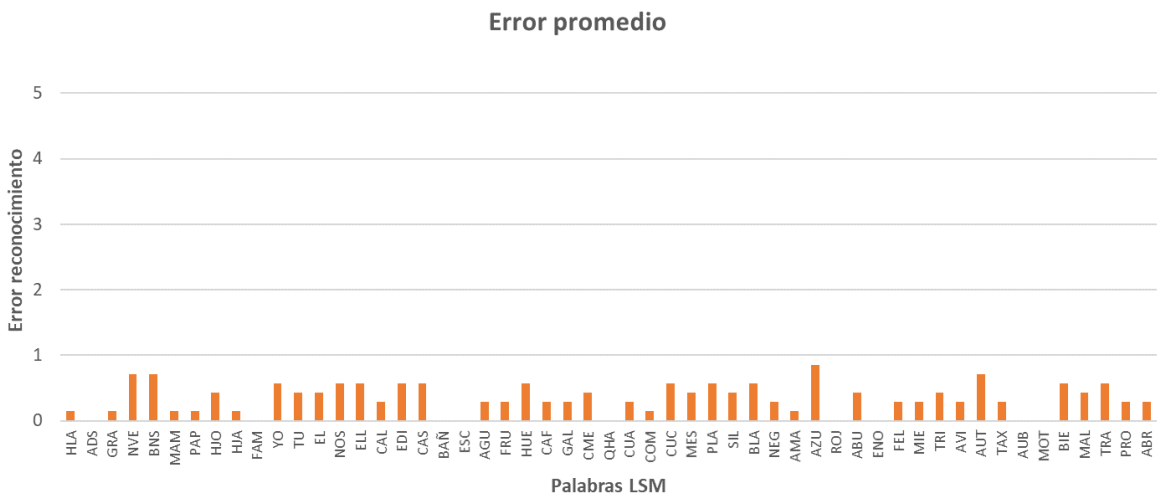


Figura 11. Matrices de confusión de cada iteración.

En la Figura 12, se observa el error promedio resultante de la diferencia del valor esperado y el valor obtenido por el sistema de reconocimiento en cada *kfold*. En esta gráfica se puede apreciar de mejor manera que la palabra que más se confunde al momento de realizar el reconocimiento es AZUL la cual se identifica como AZU en la gráfica. En el otro lado estaría que el sistema reconoce sin problemas 9 palabras, las cuales son: ADIOS, FAMILIA, BAÑO, ESCUELA, ¿QUÉ HACES?, ROJO, ENOJADO, AUTOBUS y MOTOCICLETA. Identificadas en la gráfica como: ADS, FAM, BAÑ, ESC, QHA, ROJ, ENO, AUB y MOT, respectivamente.



5.- Conclusiones y Perspectivas

De acuerdo con los resultados obtenidos en este trabajo se concluye que este modelo tiene como principal ventaja la descripción de la trayectoria del movimiento de la mano a partir de un procedimiento simple y rápido.

La precisión media alcanzada del 93.46% muestra que el modelo presenta una buena eficiencia al reconocer las palabras de la LSM aprendidas con el método de redes neuronales. Además, al existir un alto número de patrones diferentes, se considera que este modelo es capaz de reconocer un número mucho mayor de palabras y puede ser utilizado para el reconocimiento de gestos en otras áreas tal es el caso de la robótica de servicio, en donde se controlaría un robot mediante el lenguaje natural de la lengua de señas.

En trabajos futuros se plantea ampliar tanto el número de palabras como la cantidad de muestras del corpus que nos permita contar con más información para continuar realizando experimentos en el reconocimiento de la LSM. Además, se pretende implementar este enfoque en tiempo real para validar su desempeño y dar cavidad a pruebas con fines educativos y experimentales en instituciones de apoyo a personas con discapacidad en la escucha o en el habla.

Finalmente, se puede afirmar que este trabajo representa una buena aproximación para la construcción de un traductor de la LSM. Por lo que también se propone en trabajos futuros realizar el procesamiento de todos los datos que se extraen mediante el sensor Kinect. Es decir utilizar las nubes de puntos, imágenes de color y profundidad que se encuentran en el corpus creado para robustecer este trabajo.

Referencias

INEGI Censo población 2020. (2021). Población. Discapacidad. Cuentame.inegi.org.mx. Recuperado en 5 de mayo de 2021, de: http://www.cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P.

Sensor Kinect. (2021). Microsoft Kinect for Windows Specs y Prices. CNET. Retrieved 3 June 2021, from https://www.cnet.com/products/microsoft-kinect-for-windows/.

Microsoft Kinect SDK. (2021). Download Kinect for Windows SDK v1.0 from Official Microsoft Download Center. Microsoft.com. Retrieved 8 July 2021, from https://www.microsoft.com/en-us/download/details.aspx?id=28782.

Jimenez, J., Martin, A., Uc, V. y Espinosa, A. (2017). Mexican Sign Language Alphanumerical Gestures Recognition using 3D Haar-like Features. IEEE Latin America Transactions, 15(10), 2000–2005. https://doi.org/10.1109/TLA.2017.8071247

Pérez, L. M., Rosales, A. J., Gallegos, F. J., y Barba, A. V. (2017). LSM static signs recognition using image processing, 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2017), pp. 1-5, doi: 10.1109/ICEEE.2017.8108885.

Carmona-Arroyo, G., Rios-Figueroa, H. V., y Avendaño-Garrido, M. L. (2021). Mexican Sign-Language Static-Alphabet Recognition Using 3D Affine Invariants. In M. Malarvel, S. R. Nayak, P. K. Pattnaik, y S. N. Panda (Eds.), Machine Vision Inspection Systems, Volume 2 (1st ed., pp. 171–192). Wiley. https://doi.org/10.1002/9781119786122.ch9

Lahamy, H., y Lichti, D. (2012). Towards Real-Time and Rotation-Invariant American Sign Language Alphabet Recognition Using a Range Camera. Sensors, 12(11), 14416–14441. MDPI AG. Retrieved from http://dx.doi.org/10.3390/s121114416

Agarwal, A. y Thakur, M. K. (2013). Sign language recognition using Microsoft Kinect. 2013 Sixth International Conference on Contemporary Computing (IC3), 181–185. https://doi.org/10.1109/IC3.2013.6612186

Luis-Pérez, F. E., Trujillo-Romero, F., y Martínez-Velazco, W. (2011). Control of a Service Robot Using the Mexican Sign Language. Advances In Soft Computing, 419-430. https://doi.org/10.1007/978-3-642-25330-0_37

Estrivero-Chavez, C., Contreras-Teran, M., Miranda-Hernandez, J., Cardenas-Cornejo, J., Ibarra-Manzano, M., y Almanza-Ojeda, D. (2019). Toward a Mexican Sign Language System using Human Computer Interface. 2019 International Conference On Mechatronics, Electronics And Automotive Engineering (ICMEAE). https://doi.org/10.1109/icmeae.2019.00010

LeapMotion. (2021). LeapMotion Datasheet. Ultraleap.com. Retrieved 5 June 2021, from https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf.

Garcia-Bautista, G., Trujillo-Romero, F., y Caballero-Morales, S. (2017). Mexican sign language recognition using kinect and data time warping algorithm. 2017 International Conference On Electronics, Communications And Computers (CONIELECOMP). https://doi.org/10.1109/conielecomp.2017.7891832

Tazhigaliyeva, N., Kalidolda, N., Imashev, A., Islam, S., Aitpayev, K., Parisi, G., y Sandygulova, A. (2017). Cyrillic manual alphabet recognition in RGB and RGB-D data for sign language interpreting robotic system (SLIRS). 2017 IEEE International Conference On Robotics And Automation (ICRA). https://doi.org/10.1109/icra.2017.7989526

Hazari, S., Asaduzzaman, Alam, L., y Goni, N. (2017). Designing a sign language translation system using kinect motion sensor device. 2017 International Conference On Electrical, Computer And Communication Engineering (ECCE). https://doi.org/10.1109/ecace.2017.7912929

Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., y Zhou, M. (2013, April). Sign language recognition and translation with kinect. In IEEE Conf. on AFGR (Vol. 655, p. 4).

Ghotkar, A., y Kharate, G. (2015). Dynamic Hand Gesture Recognition for Sign Words and Novel Sentence Interpretation Algorithm for Indian Sign Language Using Microsoft Kinect Sensor. Journal of Pattern Recognition Research, 10(1), 24–38. https://doi.org/10.13176/11.626

Garcia-Bautista, G., Trujillo-Romero, F., y Diaz-Gonzalez, G. (2016). Advances to the development of a basic Mexican sign-to-speech and text language translator (A. G. Tescher, Ed.; p. 99713E). https://doi.org/10.1117/12.2238281

Sosa-Jimenez, C., Rios-Figueroa, H., Rechy-Ramirez, E., Marin-Hernandez, A., y Gonzalez-Cosio, A. (2017). Real-time Mexican Sign Language recognition. 2017 IEEE International Autumn Meeting On Power, Electronics And Computing (ROPEC). https://doi.org/10.1109/ropec.2017.8261606

Molchanov, P., Gupta, S., Kim, K. y Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1–7. https://doi.org/10.1109/CVPRW.2015.7301342

Oyewole, O. G., Nicholas, G., Oludele, A., y Samuel, O. (2018). Bridging communication gap among people with hearing impairment: An application of image processing and artificial neural network. International Journal of Information and Communication Sciences, 3(1), 11.

Gürpınar, C., Uluer, P., Akalın, N. et al. Sign Recognition System for an Assistive Robot Sign Tutor for Children. Int J of Soc Robotics 12, 355–369 (2020). https://doi.org/10.1007/s12369-019-00609-9

Zhang, Z., Su, Z., y Yang, G. (2019). Real-Time Chinese Sign Language Recognition Based on Artificial Neural Networks*. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 1413–1417. https://doi.org/10.1109/ROBIO49542.2019.8961641

Jiang, S., Sun, B., Wang, L., Bai, Y., Li, K., y Fu, Y. (2021). Skeleton aware multi-modal sign language recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3413-3423).

Fregoso, J., Gonzalez, C. I., y Martinez, G. E. (2021). Optimization of Convolutional Neural Networks Architectures Using PSO for Sign Language Recognition. Axioms, 10(3), 139. MDPI AG. Retrieved from http://dx.doi.org/10.3390/axioms10030139

Calvo, M.T. (2004). Diccionario Español - Lengua de Señas Mexicana (DIELSEME): estudio introductorio. Dirección de Educación Especial: México.

Serafín de Fleischmann, M., González Pérez, R. (2011). Manos con voz, Diccionario de Lenguaje de Señas Mexicana. Primera edición, Libre Acceso, A.C., ISBN 978-607-9134-0I-3

Código estándar Mex-Esp. (2021). ISO 639 — Language codes. ISO. Retrieved 14 Mayo 2021, from https://www.iso.org/iso-639-language-codes.html.

Zhang, Z. (2012). Microsoft Kinect Sensor and Its Effect. IEEE MultiMedia, 19(2), 4–10. https://doi.org/10.1109/MMUL.2012.24

Dal Mutto, C., Zanuttigh, P., y Cortelazzo, G. (2012). Time-of-flight cameras and Microsoft Kinect. Springer.

Rossenblatt, F. (1957). The perceptron, a perceiving and recognizing automation. Cornell Aeronautical Laboratory. Report No. 85-460-1.

Werbos, P. (1974). Beyond Regression: New tools for prediction and analysis in the behavioral sciences (Ph.D). Harvard University.

Rumelhart, D. E., Hinton, G., y Williams, R. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536. https://doi.org/10.1038/323533a0

BSL Corpus. (2021). Home | BSL Corpus Project. British Sign Language Corpus Project. Retrieved 13 March 2021, from http://www.bslcorpusproject.org/.

GSL Corpus. (2021). German Sign Language Korpus. Retrieved 14 March 2021, from: http://www.sign-lang.unihamburg.de/dgs-korpus/index.php/welcome.html

LSE Corpus. (2021). Corpus de la lengua de signos española. Corpuslse.es. Retrieved 14 July 2021, from https://www.corpuslse.es/.

LSP Corpus. (2021). Repositorio Digital de la Lengua de Señas Peruana - Grupo Señas Gramaticales. Grupo Señas Gramaticales. Retrieved 10 March 2021, from https://investigacion.pucp.edu.pe/grupos/senasgramaticales/proyecto/repositorio-digital-de-la-lengua-de-senas-peruana/.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-Compartirlgual 4.0 Internacional.