Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Desarrollo de un sistema automático de reconocimiento de la LSM para dispositivos móviles

Development of an automatic LSM recognition system for mobile devices

Mendoza-Tene, J.L.[a], Fajardo-Delgado, D.[a]*, Fajardo-Flores, S.B.[b], Puga-Nathal, M.E.[a], Sánchez-Cervantes, M.G.[a]

[a] Tecnológico Nacional de México / Instituto Tecnológico de Ciudad Guzmán, Av. Tecnológico 100, Cd. Guzmán C.P. 49100, Jal., México.
[b] Facultad de Telemática, Universidad de Colima, Av. Universidad 333, Colima C.P. 28040, Col., México.
m23291009@cdguzman.tecnm.mx; daniel.fd@cdguzman.tecnm.mx*; medusa@ucol.mx; maria.pn@cdguzman.tecnm.mx; maría.sc1@cdguzman.tecnm.mx

Innovación tecnológica: Uso de dispositivos convencionales para el reconocimiento de la LSM.
Área de aplicación industrial: Tecnologías para la inclusión social.

Recibido: 18 enero 2024
Aceptado: 25 noviembre 2024

Abstract

This paper explores the effectiveness of using mobile devices for the automatic recognition of both static and dynamic signs in Mexican Sign Language (LSM). This is due to the fact that most current works on LSM recognition typically focus on one type of sign (static or dynamic) and often involve specialized devices beyond the reach of the general population. To conduct this study, a solution was implemented, consisting of an Android application for capturing videos and sending them to a cloud-based web service hosting an automatic recognition system. The system employs the Dynamic Time Warping (DTW) algorithm to compare the features of the received video with those of a set of videos categorized by sign. It is noteworthy that this work is the first to use DTW for static signs in LSM. Finally, an experimental study of the system's performance was conducted using three mobile devices with different capabilities for both video capture and processing. Results indicate that, while camera quality affects recognition efficiency, it is not a decisive factor. Additionally, due to the video characterization method and LSM-specific attributes, the system tends to recognize static signs better than dynamic ones, even when using DTW.

Keywords: Mexican sign language, Automatic recognition system, Dynamic time warping, Mobile app.

Resumen

En este artículo se estudia la efectividad de utilizar dispositivos móviles para el reconocimiento automático de señas estáticas y dinámicas de la lengua de señas mexicana (LSM). Esto debido a que la mayoría de los trabajos actuales para el reconocimiento de la LSM, generalmente se centran en un tipo de seña (estática o dinámica) y casi siempre consideran dispositivos especializados fuera del alcance de la población. Para llevar a cabo este estudio, se implementó una solución que consta de una aplicación Android para capturar vídeos y enviarlos a un servicio Web en la nube, donde se aloja un sistema de reconocimiento automático. El sistema utiliza el algoritmo de deformación dinámica del tiempo (DTW, por sus siglas en inglés) para comparar las características del vídeo recibido con las de un conjunto de vídeos catalogados por seña. Cabe resaltar que este trabajo sería el primero en utilizar el DTW para señas estáticas de la LSM. Finalmente, se llevó a cabo un estudio experimental del desempeño del sistema utilizando tres dispositivos móviles de diferentes capacidades tanto para la captura del vídeo como en el procesamiento. Los resultados indican que, aunque la calidad de la cámara afecta la eficiencia del reconocimiento, esta no es determinante; además, debido al método de caracterización de los vídeos y por atributos propios de la LSM, el sistema tiende a reconocer mejor las señas estáticas que las dinámicas, incluso con el uso del DTW.

Palabras clave: Lengua de señas mexicana, Sistema de reconocimiento automático, Deformación dinámica del tiempo, Aplicación móvil.

1. **Introducción**

En México, alrededor de 2.9 millones de personas presentan algún grado de limitación auditiva, y aproximadamente 1.3 millones más padecen una discapacidad auditiva o sordera que dificulta o imposibilita sus capacidades de comunicación [1]. Una forma natural de expresión para las personas sordas es la lengua de señas, un sistema de comunicación visual y gestual con gramática y lingüística propia. La lengua de señas oficial en México es la lengua de señas mexicana (LSM) [2], que se compone de una amplia variedad de señas realizadas con una o ambas manos, combinadas con expresiones faciales y movimientos corporales con particularidades de una función lingüística de la comunidad sorda en el país [3]. Sin embargo, solamente el 0.2% de la población en México conoce y/o utiliza la LSM [4], lo que evidencia una brecha en la comunicación y comprensión de las necesidades básicas de las personas sordas por parte de la población oyente [5, 6].

Actualmente, se han desarrollado diversos sistemas de reconocimiento automático de la LSM en la literatura científica, e.g., [7-18]. La mayoría de ellos clasifican las señas de la LSM en dos tipos según la naturaleza temporal: estáticas y dinámicas. Las señas estáticas mantienen sus características constantes de principio a fin, mientras que las señas dinámicas implican una transición desde un estado inicial a uno final, con movimientos específicos en su configuración. La Tabla 1 muestra algunos trabajos de investigación en los que se desarrollaron sistemas de reconocimiento automático de señas estáticas o dinámicas de la LSM y la exactitud de sus modelos de reconocimiento.

Tabla 1. Comparación de la exactitud de distintos modelos de reconocimiento de la LSM con base en el tipo de seña (estática o dinámica).

Modelos de reconocimiento LSM	Señas	Exactitud
Ríos-Figueroa et al. [7]	Estáticas	99.70%
Rodríguez et al. [8]		92.72%
Fregoso et al. [9]		99.05%
García-Gil et al. [10]		99%
Morfín-Chávez et al. [11]		95.30%
Espejel-Cabrera et al. [12]		96.27%
Mejía-Peréz et al. [13]		97%
Trujillo-Romero et al. [14]	Dinámicas	90.27%
Solis-V et al. et al. [15]		95.83%
García-Bautista et al. [16]		98.57%
Sosa-Jiménez et al. [17]		99.40%
González-Rodríguez [18]		98.80%

Los sistemas de reconocimiento de la LSM de la Tabla 1 utilizan imágenes o vídeos para capturar información gestual de una o ambas manos, analizando el movimiento, orientación, forma o ubicación. Sin embargo, la mayoría de estos sistemas requiere equipo especializado como Microsoft Kinect (empleado por [7,14,16,17]) o cámaras de profundidad (utilizadas por [13,18]), lo cual limita su accesibilidad para la población en general. Además, muchas de estas soluciones están diseñadas y evaluadas bajo condiciones controladas de vestimenta, iluminación y ángulos de visión poco probables en la cotidianidad (como las propuestas por [8,9,12,15]), y rara vez se materializan en aplicaciones de software de acceso público. En general, estos sistemas suelen centrarse en señas estáticas o dinámicas de la LSM por separado [7-18] y rara vez integran ambas en el mismo modelo, lo que limita su utilidad en contextos prácticos, ya que la LSM comúnmente combina ambas modalidades.

En México, cuatro de cada cinco habitantes cuentan con un dispositivo móvil (celular) [19], por lo que generar una aplicación para este tipo de dispositivos podría facilitar la comunicación entre personas oyentes y personas sordas gracias a la versatilidad y variedad existentes en el mercado. Aunque en la actualidad existen algunas aplicaciones móviles relacionadas con la LSM, la mayoría se limita a la enseñanza de señas básicas y muy pocas integran un sistema de reconocimiento automático. Un ejemplo de ello es el trabajo de Martínez-Seis et al. [20], que presenta una aplicación móvil para el reconocimiento de señas estáticas y dinámicas de la LSM; sin embargo, dicha aplicación no está disponible para descarga en la actualidad.

En el presente trabajo, se desarrolló un sistema de reconocimiento automático de la LSM y se evaluó su desempeño en dispositivos móviles (celulares) convencionales en situaciones cotidianas. En particular, se implementó una adaptación de los trabajos de Pham et al. [21] y Guerin [22], que utilizan el algoritmo de deformación dinámica del tiempo (o DTW, *Dynamic Time Warping*) para comparar la similitud entre vídeos de señas dinámicas. Guerin utiliza un flujo de transmisión de vídeo constante para el reconocimiento de señas dinámicas de la lengua de señas francesa. A diferencia de Guerin, el sistema aquí desarrollado se alojó

en un servidor en la nube capaz de comparar vídeos cortos obtenidos desde dispositivos móviles utilizando como datos de entrenamiento tanto señas dinámicas como estáticas de la LSM.

2. Materiales y métodos

2.1 Corpus de señas

El corpus que se utilizó para el presente trabajo consiste en vídeos de señas estáticas y dinámicas de la LSM. Una seña dinámica tiene un estado inicial y una transición hacia un estado final que implica un movimiento en su configuración, mientras que la estática no genera ningún cambio. Se consideraron 37 señas en total: 16 dinámicas y 21 estáticas, las cuales se obtuvieron mediante la colaboración de cinco actores. Cada actor fue grabado al menos tres veces para una seña en particular, variando las vestimentas, los fondos y la velocidad de ejecución. Asimismo, se le pidió al actor que utilizara principalmente la mano derecha para realizar las señas. De esta forma, se grabaron 22 vídeos por cada una de las 37 señas, generando un total de 814 vídeos para el corpus. Las señas utilizadas fueron todas las del abecedario (21 estáticas y seis dinámicas), además de otras diez señas dinámicas utilizadas por [14], que son de uso habitual en la comunicación casual.

La Tabla 2 muestra las señas dinámicas seleccionadas para el corpus.

Tabla 2. Señas dinámicas seleccionadas.

Señas dinámicas		
Seña	**Abreviación**	**Manos usadas**
Bien	BN	Derecha
Gracias	GRA	Ambas
Hola	HLA	Derecha
No	NO	Derecha
Perdón	PDN	Ambas
Si	SI	Derecha
Tu	TU	Derecha
Yo	YO	Derecha
¿Cómo estás?	CME	Ambas
J, K, Ñ, Q, X, Z	-	Derecha

Los vídeos se grabaron utilizando un dispositivo Apple iPhone 14 Pro-Max a máxima resolución (2160×3840 píxeles). Incluso en este dispositivo, los vídeos originalmente podían tener resoluciones distintas, por lo que se llevó a cabo diferentes tareas de preprocesamiento OpenCV (*Open Computer Vision Library*) [23], una librería de código abierto con más de 2,500 algoritmos para el manejo de la visión computacional y aprendizaje automático. Estas tareas de preprocesamiento tuvieron como objetivo homogeneizar los vídeos a una velocidad de 30 fotogramas por segundo, un máximo de cuatro segundos de duración (dando un total máximo de 120 fotogramas), y un ancho y alto de fotogramas de 1280×720 píxeles, respectivamente. Posteriormente, se aplicó un difuminado a los rostros de los actores dentro de los vídeos, con el fin de proteger su privacidad.

El corpus de vídeos de señas estáticas y dinámicas de la LSM generado en este trabajo está disponible para su descarga pública en: https://github.com/RexTime007/MSL-video-corpus.

2.2 Diseño del prototipo de software

Para poder realizar pruebas con dispositivos móviles de diferentes especificaciones técnicas, fue necesario diseñar un prototipo de software que procesara de forma independiente la selección automática de señas. Inicialmente, se propuso migrar el trabajo de [22] directamente a una aplicación móvil, conjuntando la captura y procesamiento de datos en cada celular. Sin embargo, después de realizar algunas pruebas preliminares utilizando un mismo vídeo para diferentes celulares, se observó que se generaban resultados diferentes. Se conjetura que esto puede deberse a las capacidades de procesamiento y de cálculo distintas para cada dispositivo. Por esta razón, se decidió segmentar la aplicación para que los dispositivos móviles se encargaran únicamente de la captura de vídeos y la visualización de resultados, mientras que, en la nube, se ejecute de forma independiente el sistema de reconocimiento automático de la LSM. Así, la calidad de los vídeos puede ser distinta dependiendo de las capacidades de cada dispositivo móvil, pero el proceso de predicción es independiente de ellas.

La Figura 1 muestra la arquitectura general del sistema de reconocimiento automático de la LSM. Destacan tres componentes principales: la aplicación móvil, un servicio Web alojado en la nube de Google para recibir/enviar peticiones desde/hacia la aplicación, y el sistema de reconocimiento/clasificación de la LSM contenida en la misma nube.

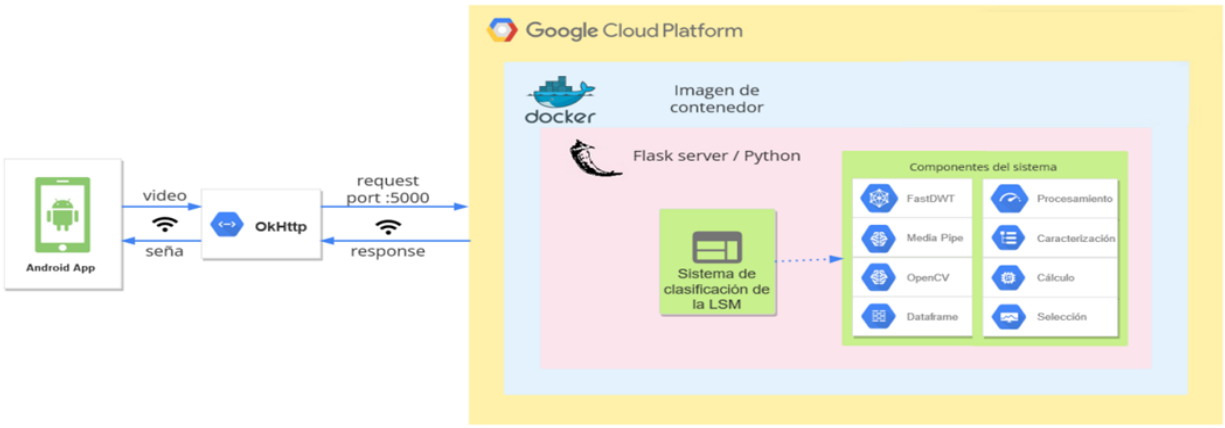

Figura 1. Arquitectura del prototipo.

La aplicación móvil está programada utilizando el kit de desarrollo estándar (SDK) de Android, garantizando la compatibilidad con una amplia gama de dispositivos que operan bajo este sistema. Desde esta aplicación, se capturan vídeos de máximo cuatro segundos conteniendo la grabación de una seña de la LSM. Esta limitación en el tiempo de grabación de los vídeos evita la comparación excesiva de imágenes (o fotogramas) en la secuencia de los mismos una vez recibida en el servidor.

La Figura 2 muestra las pantallas de inicio de la aplicación y un ejemplo del botón de captura de vídeo, mientras que la Figura 3 muestra la pantalla que contiene el vídeo grabado; éste puede ser aceptado y mandado al servidor o cancelado y grabado nuevamente hasta que se obtenga una seña satisfactoria para clasificar.

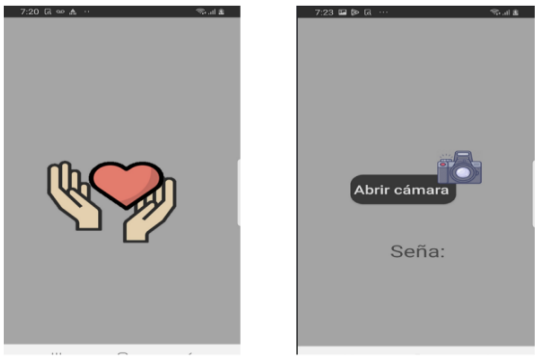

Figura 2. Pantallas del sistema.

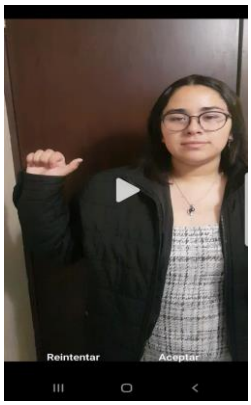

Figura 3. Pantalla de grabación del vídeo.

Una vez capturado el vídeo dentro del dispositivo móvil, se envía al servidor Web a través de OkHttp, una librería de código abierto que facilita el intercambio de archivos a través del protocolo HTTP. Con esta librería, el dispositivo móvil es un cliente listo para realizar peticiones al servidor y recibir respuesta de predicción. Para esto, se crea un objeto OkHttp al que se le asigna un localizador de recursos uniforme (o URL, por sus siglas en inglés) del servidor y un puerto disponible para manejar dicha petición. El cuerpo de la petición se conforma por los datos previos más el contenido del vídeo previamente grabado, siendo agrupado y enviado al servidor. Esta solicitud se mantiene activa hasta recibir una respuesta satisfactoria, un error o se exceda el tiempo de conexión.

El servidor Web se ejecuta en una instancia de Google Cloud Run [24], una plataforma de la nube de Google diseñada para la administración y ejecución de contenedores de forma escalable. Tanto el servicio Web como el sistema de reconocimiento automático de señas LSM se implementan en un contenedor Docker, el cual empaqueta las dependencias necesarias para la ejecución del servidor, tales como: un sistema operativo Debian, el lenguaje de programación Python y las librerías esenciales para el proceso de caracterización de los vídeos. El servicio Web se ejecuta a través de Flask, un marco de trabajo ligero basado en Python que administra el procesamiento de solicitudes y respuestas HTTP.

El proceso de selección de la seña en el servidor consiste en utilizar una seña predicha, que es aquella que muestra mayores similitudes con la que se observa en los vídeos de prueba. Finalmente, el resultado se envía al cliente, el cual despliega la seña predicha en la pantalla principal de la aplicación

2.3 Sistema de reconocimiento

2.3.1 Obtención de características

Las características utilizadas fueron puntos de referencia (o *landmarks*) obtenidos de todos los fotogramas de cada uno de los vídeos del corpus mediante MediaPipe [25], un conjunto de herramientas y librerías con modelos pre-entrenados para la detección, segmentación y/o clasificación de objetos para los campos de visión, texto y audios. En particular, se utilizó el modelo Holistics de MediaPipe, que integra una percepción simultánea de rastreo de cara, manos y pose de cuerpo en tiempo real (Figura 4). Este modelo permite identificar 543 puntos de referencia que se agrupan en diferentes regiones de interés: 468 puntos de la cara, 21 puntos para cada mano y 33 puntos del cuerpo. Para este trabajo, solo se utilizaron los puntos de referencia de dos regiones de interés: las manos y los brazos.

No se consideraron gestos faciales ni poses, presentes en algunas señas de la LSM.

Figura 4. Modelo de MediaPipe Holistic aplicado a una imagen.

2.3.2 Caracterización de las manos

Para la caracterización de las manos, se utilizan los ángulos entre los 21 puntos de referencia de cada mano observable y se almacenan en un vector. Después, se utilizan las posiciones de *x* y *y* de cada punto de referencia de las manos normalizadas a un rango entre 0 y 1 del ancho y alto de la imagen o fotograma. Dado que en el presente trabajo no se utilizan cámaras de profundidad o 3D, la librería MediaPipe con su modelo Hands asigna al eje *z* de cada punto de referencia la proximidad de ellos con respecto a la cámara, calculado a través de un mapa de profundidad de distintas superficies observables en el fotograma (ver Figura 5).

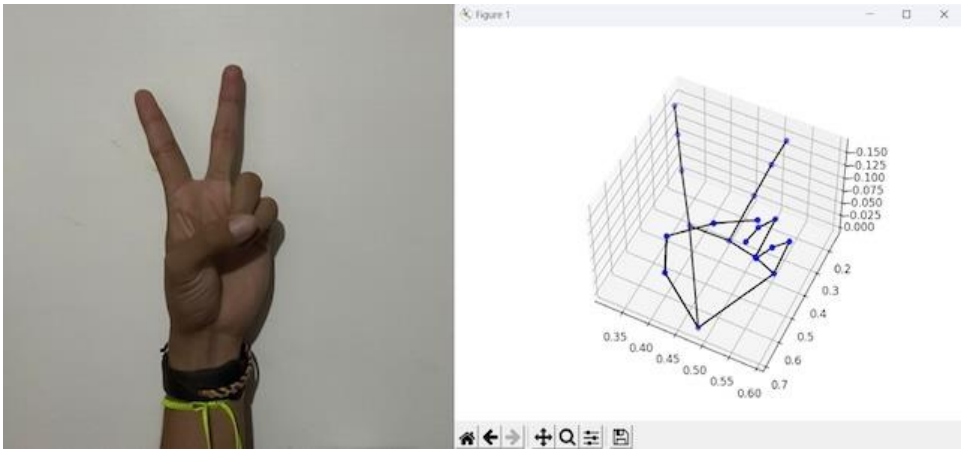

Figura 5. Ejemplo del cálculo de los puntos de referencia de MediaPipe para una mano.

Dadas las coordenadas *x*, *y*, *z* generadas por los puntos de referencia, éstas se agrupan en una tupla y se almacenan en una matriz. Después, se consideran todas las combinaciones de pares de elementos de dicha matriz para calcular los ángulos en el espacio en tres dimensiones. En Ec. (1) se presenta la fórmula para el cálculo de los ángulos relativos entre pares de vectores en tres dimensiones.

$$\theta = \arccos \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\|\|\vec{v}\|}, \qquad \textbf{(Ec. 1)}$$

Donde θ denota el ángulo obtenido entre un par de vectores, que se calcula mediante la función inversa del coseno, aplicada a la razón entre el producto punto de los vectores $\vec{u}$ y $\vec{v}$ y el producto de las magnitudes de ambos vectores. El uso de ángulo relativos, permite que el sistema tenga propiedades invariantes al tamaño, posición absoluta y a la rotación de las señas dentro del proceso de clasificación.

Los ángulos son almacenados en arreglos que diferencian entre mano izquierda y derecha, y que pueden variar de longitud dependiendo del número de apariciones de dicha mano en el vídeo, en donde considerando los 21 puntos de referencia, existe un total de 441 ángulos de pares de vectores calculados.

2.3.3 Creación del conjunto de entrenamiento

Una vez obtenidas las características, estas se almacenan en archivos Pickle, que son archivos binarios de Python que contienen una versión serializada de las estructuras con información para las manos y los brazos. Cada archivo se nombra en función de la seña y el vídeo correspondiente. El conjunto de archivos Pickle constituye el conjunto de entrenamiento para la clasificación.

2.3.4 Similitud entre señas

Se utiliza una métrica de similitud que permite comparar cualquier vídeo que contenga una seña de la LSM con los vídeos de señas considerados en el conjunto entrenamiento. Entre mayor similitud arrojada por la métrica para un vídeo en particular, mayor la probabilidad de que la seña grabada en el vídeo corresponda entre sí. Sin embargo, la longitud del nuevo vídeo puede variar respecto a los utilizados en el conjunto entrenamiento, así como el número de apariciones de una o ambas manos por fotograma y la velocidad en la que se grabó la seña. En particular, cada vídeo se representa como una *serie de tiempo*, i.e., un conjunto ordenado $T = t_1, \ldots, t_m$ de variables de valores [26].

Sea m_1 el número de fotogramas de cualquier vídeo considerado en el conjunto de entrenamiento, y sea m_2 el número de fotogramas del vídeo de prueba. Dado que puede ocurrir que $m_1 \neq m_2$, se requiere de la implementación de un método que permita comparar series de tiempo de longitud variable, en este caso $m_1 \times 441$ con $m_2 \times 441$. En particular, en este trabajo se utiliza el algoritmo de deformación dinámica del tiempo.

El algoritmo de deformación dinámica del tiempo, o DTW [27, 28], es una técnica que permite determinar la similitud entre dos series de tiempo, incluso cuando existe una diferencia en la longitud o un desfase temporal entre ellas (ver Figura 6). Para ello, el DTW crea una matriz de distancias entre los puntos de las dos series y busca la ruta óptima que minimice la distancia acumulada. La distancia mínima acumulada a lo largo de dicha ruta óptima, denominada aquí como distancia-DTW, es el valor que indica cuán similares son las dos series de tiempo.

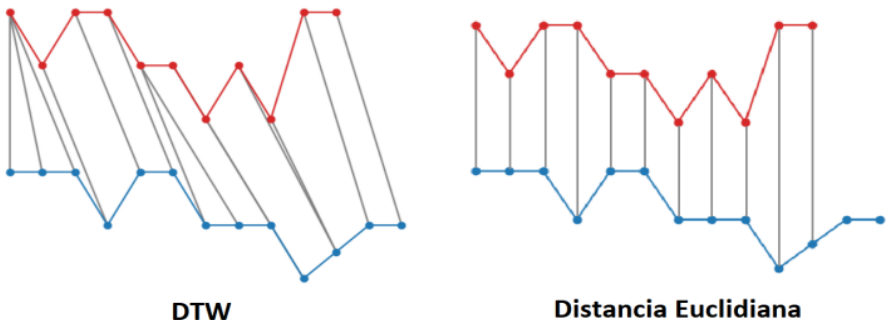

Figura 6. Similitud entre pares de series de tiempo del DTW versus el uso.

La implementación del DTW tiene un alto costo computacional que, en este caso, está dictado por el número de fotogramas de los 814 vídeos convertidos a series de tiempo y la serie tiempo que será sometida para su clasificación. Para menguar con estos costos computacionales en tiempo de ejecución, se hace uso de una implementación de la transformada rápida de Fourier disponible en Python.

El sistema propuesto para el reconocimiento de señas de la LSM utiliza el DTW para procesar y comparar el vídeo de una seña en particular con cada vídeo del conjunto de entrenamiento.

2.3.5 Selección de señas

Como resultado de la comparación entre vídeos de señas utilizando DTW, se obtiene un valor de distancia-DTW para cada vídeo de seña del conjunto de entrenamiento. Las distancias-DTW resultantes se ordenan de forma ascendente, donde un valor menor indica una mayor similitud entre dos señas. En particular, se consideran únicamente las primeras 13 coincidencias encontradas, las cuales se agrupan por seña, y se cuenta el número de instancias de cada agrupación. Este número impar de coincidencias fue determinado mediante pruebas preliminares realizadas con el prototipo. Posteriormente, se selecciona la distancia-DTW mínima de cada conjunto y se divide entre el número de instancias de cada grupo de señas, generando una "distancia-DTW grupal". La seña cuya distancia-DTW grupal sea la menor después de este procedimiento se considera como dominante de ese caso y es retornada como la seña identificada en el vídeo. Todos los procesos previamente mencionados son encapsulados en los módulos del prototipo de software generado.

Las Tablas 3 y 4 ilustran el procedimiento de selección de seña. Por ejemplo, suponga un vídeo donde la seña grabada es la letra G. El sistema procesa este vídeo y los resultados de similitud obtenidos se muestran en la Tabla 3, donde las coincidencias con las señas son las siguientes: cinco coincidencias para G, cinco para H y tres para L. Estas incidencias se ordenan de forma ascendente según la distancia-DTW. Enseguida, se toma la distancia-DTW correspondiente a la primera incidencia de una seña en la tabla (o distancia mínima) de acuerdo con sus posiciones: uno para G, cinco para la H y diez para L. Posteriormente, cada distancia-DTW se divide entre el número de incidencias para cada seña, generando así una distancia-DTW grupal. Finalmente, las distancias obtenidas se ordenan de forma ascendente, siendo la primera de estas, la seña G. De este modo, G se considera la seña dominante y constituye el resultado de la predicción final (ver Tabla 4).

Tabla 3. Ejemplo de las primeras 13 distancias-DTW.

Posición	Seña	Distancia-DTW
1	**G**	1239.93
2	G	1430.32
3	G	1902.30
4	G	1920.41
5	H	1960.32
6	H	1979.21
7	H	2013.96
8	G	2030.10
9	H	2145.88
10	L	2450.48
11	H	2462.41
12	L	2740.01
13	L	3036.89

Tabla 4. Agrupación de señas.

Seña	Distancia-DTW mínima	Número de incidencias	Distancia-DTW grupal
G	**1239.93**	5	247.97
H	1960.32	5	392.06
L	2450.47	3	816.83

3. Experimentación

Para la ejecución de los experimentos, se consideraron los siguientes tres dispositivos móviles: Samsung Galaxy A32 (A32), Huawei P40 Lite (P40) y Samsung Galaxy S8 (S8), listados en orden por su rendimiento y calidad de las cámaras (ver Tabla 5). Todos cuentan con 4GB de memoria RAM y 64GB de almacenamiento.

Tabla 5. Características de los dispositivos móviles.

Modelo	**Calidad del vídeo – *frames* por segundo (fps)**	**Procesador**
A32	1080p@30fps	Mediatek Helio
P40	2160p@30fps	Kirin 710F
S8	2160p@60fps	Snapdragon 835

Las pruebas se llevaron a cabo con la colaboración de cinco personas, sin conocimiento previo de la LSM, a las que se les iba instruyendo la forma en la que se debían realizar las señas frente al celular. Las cinco personas utilizaron los tres celulares para una misma seña, ubicados a metro y medio de distancia y en la misma posición. Los vídeos obtenidos para el conjunto de prueba fueron generados en entornos no controlados, con diferentes escenarios, vestimentas, niveles de iluminación y velocidad de ejecución de la seña.

Se evaluaron cada una de las 37 señas utilizadas en el corpus, donde cada persona realizó una única vez cada seña. Así, se generaron un total de 111 vídeos de prueba por persona, de los cuales cada tercio pertenece a uno de los dispositivos utilizados, quedando 37 vídeos por dispositivo móvil para cada persona; i.e., se generó un total de 185 vídeos por cada dispositivo móvil, y un total general de 555 vídeos grabados para el conjunto de prueba.

3.1 Métricas de rendimiento

Se utilizaron las siguientes métricas de rendimiento de sistemas de clasificación para comparar objetivamente los resultados: exactitud, que mide las señas correctamente clasificadas respecto al total de señas evaluadas; precisión, que es el porcentaje de señas clasificadas correctamente respecto al total de señas catalogadas como correctas; exhaustividad, la cual mide qué tan bien identifica y clasifica correctamente todas las señas positivas o en el conjunto de entrenamiento; y el Valor F, que evalúa la efectividad general del clasificador combinando la precisión y la exhaustividad.

3.2 Pruebas estadísticas

Se utilizaron las pruebas estadísticas no paramétricas de Kruskal & Wallis [29] y de Dunn [30] para determinar las diferencias significativas entre las distancias-DTW obtenidas para cada dispositivo móvil. Por otro lado, para la comparación entre las distancias obtenidas para tipos de señas de la LSM, se utilizó la prueba t-student [31] para dos muestras. En todas las pruebas se consideró un nivel de significancia $\alpha = 0.05$ y se calcularon utilizando el paquete scipy.stats de Python 3.6.

4. Resultados y discusión

Los resultados de la predicción del sistema utilizando el conjunto de prueba se concentra en una sábana de datos almacenada en un archivo con valores separados por comas (de formato CSV) que consta de 185 filas, donde se incluye: la seña esperada contra la predicha de cada una de las personas para los tres celulares, el número de incidencias encontradas por el DTW, la distancia-DTW media de dicha seña, las señas con las que encontró similitud para dicha prueba y la clasificación de la seña (estática o dinámica).

4.1 Evaluación entre dispositivos móviles

El sistema, independientemente del dispositivo móvil, tiene una exactitud promedio cercana al 0.8072, lo que es aceptable dado que se utilizaron dispositivos móviles convencionales (ver Tabla 6).

Tabla 6. Rendimiento general del modelo de reconocimiento.

Métrica/ Celular	P40	S8	A32	Promedio
Precisión	0.8457	0.8265	0.8280	0.8334
Exhaustividad	0.8216	0.7999	0.8	0.8071
Exactitud	0.8216	0.8	0.8	0.8072
Valor F	0.8053	0.7865	0.7828	0.7915

Se tomaron los valores de distancia-DTW bajo dos condiciones principales: cuando todos los dispositivos móviles acertaron en sus respuestas y cuando al menos uno de ellos falló. A partir de las tres muestras obtenidas, una por cada dispositivo, se buscó determinar si existe una diferencia significativa en el cálculo de las distancias-DTW para cada celular. Para verificar esto, se aplicó la prueba estadística de Kruskal & Wallis, la cual obtuvo un valor de probabilidad de 0.000974 con un nivel de significancia $\alpha = 0.05$. Estos resultados indican que existe una diferencia estadísticamente significativa en el cálculo de las distancias-DTW en al menos uno de los dispositivos móviles.

Posteriormente, se aplicó la prueba de Dunn (ver Tabla 7) para realizar comparaciones por pares entre cada dispositivo móvil, donde se observó que la diferencia significativa se presenta entre el A32 y el S8. Es importante notar que, aunque el desempeño en los resultados de clasificación fue muy similar entre ambos dispositivos, el cálculo de las distancias-DTW fue considerablemente diferente. De hecho, en el S8 se obtuvieron valores de distancias-DTW mucho menores en comparación con el P40 y el A32, lo que indica que el criterio utilizado para la selección de la seña fue decisivo para predicción correcta o incorrecta.

Tabla 7. Valores de probabilidad de la prueba Dunn para resultados unánimes.

	P40	S8	A32
P40	1	0.0536	0.1373
S8	0.0536	1	0.0006
A32	0.1373	0.0006	1

Se utilizó la prueba de Kruskal & Wallis en resultados donde al menos uno de los dispositivos falló en su predicción, obteniendo un valor de probabilidad de 0.4130. Esto puede estar relacionado a que los valores de distancia-DTW son altos cuando los vídeos contienen señas distintas.

Con base en los resultados de las pruebas estadísticas presentadas en la Tabla 7, se puede concluir que la caracterización de las señas obtenidas de un vídeo depende de las capacidades técnicas del dispositivo móvil. Sin embargo, no es posible determinar la naturaleza de la distinción, ya que no se evalúan de manera independiente los componentes del dispositivo (como la cámara o el procesador), sino como una unidad. Se generaron matrices de confusión bajo diferentes condiciones de la distribución de las señas esperadas y las obtenidas en cada dispositivo móvil. En todas ellas, el eje y muestra las predicciones realizadas y el eje x los resultados esperados.

La Figura 7 presenta los resultados de las predicciones en general para todos los dispositivos móviles. Allí se aprecia que el sistema clasificó de manera correcta mayormente las señas estáticas, mientras que para las dinámicas abundaron más los errores. Si bien algunas señas tienden a ser erróneamente clasificadas para una seña en particular, este comportamiento varía dependiendo del dispositivo móvil utilizado.

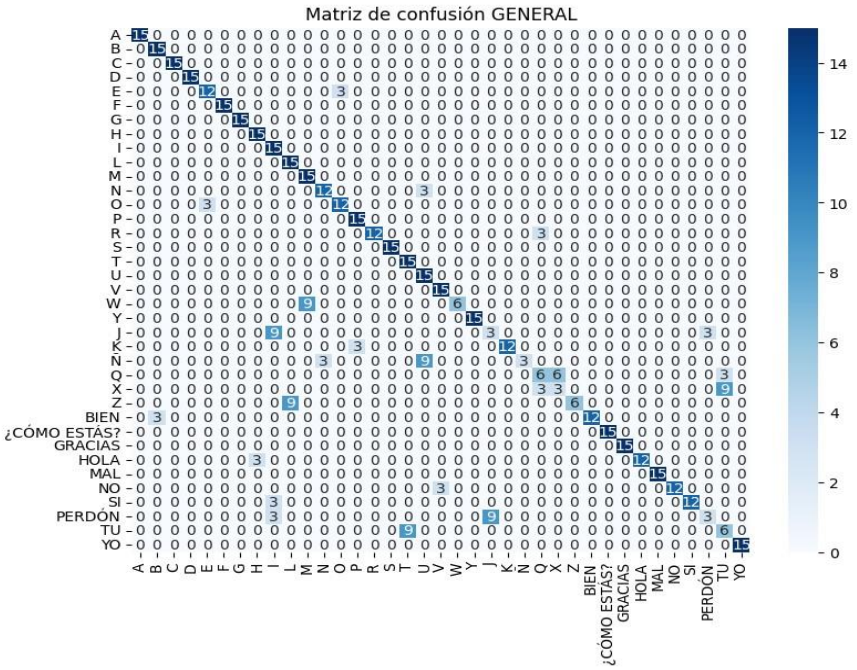

Figura 7. Matriz de confusión general.

La Figura 8 muestra los resultados de las predicciones generadas por el sistema para P40. Aquí se observa que este dispositivo tuvo un mal desempeño, a comparación de los otros dos, al momento de clasificar las señas estáticas. Por otro lado, P40 tuvo un mejor desempeño clasificando las señas dinámicas a comparación del S8 y A32. Cabe destacar que este dispositivo no logró clasificar correctamente una señal en cada una de las pruebas: la letra Ñ.

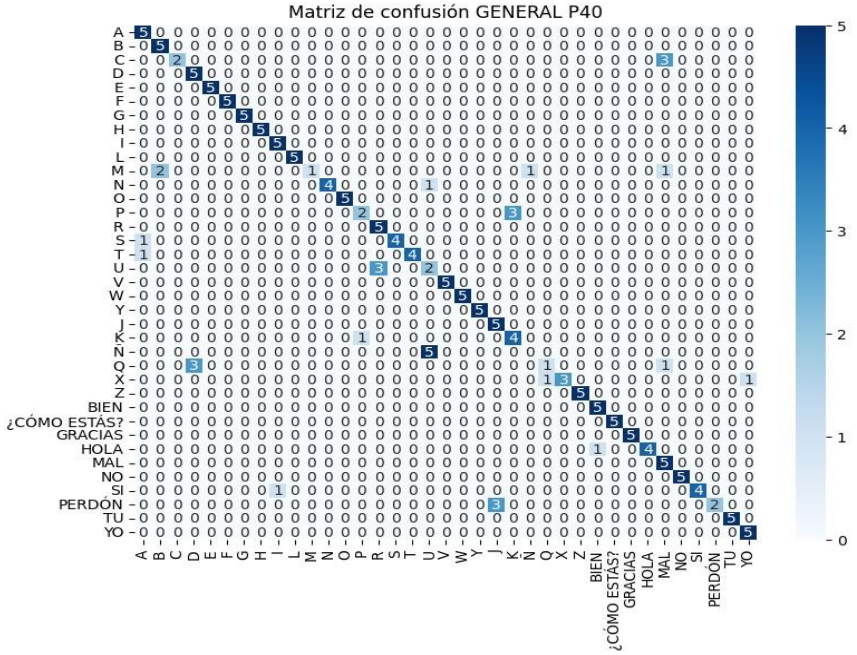

Figura 8. Matriz de confusión P40.

La Figura 9 muestra los resultados de las predicciones generadas por el sistema para A32. Aunque menos evidente que en el caso de P40, el A32 también tuvo un mejor desempeño para señas estáticas que para dinámicas. Asimismo, el A32 tampoco logró clasificar correctamente la letra Ñ.

La Figura 10 muestra los resultados de las predicciones generadas por el sistema para S8. Es importante destacar que este dispositivo fue el único clasificar correctamente al menos una vez cada seña en las pruebas.

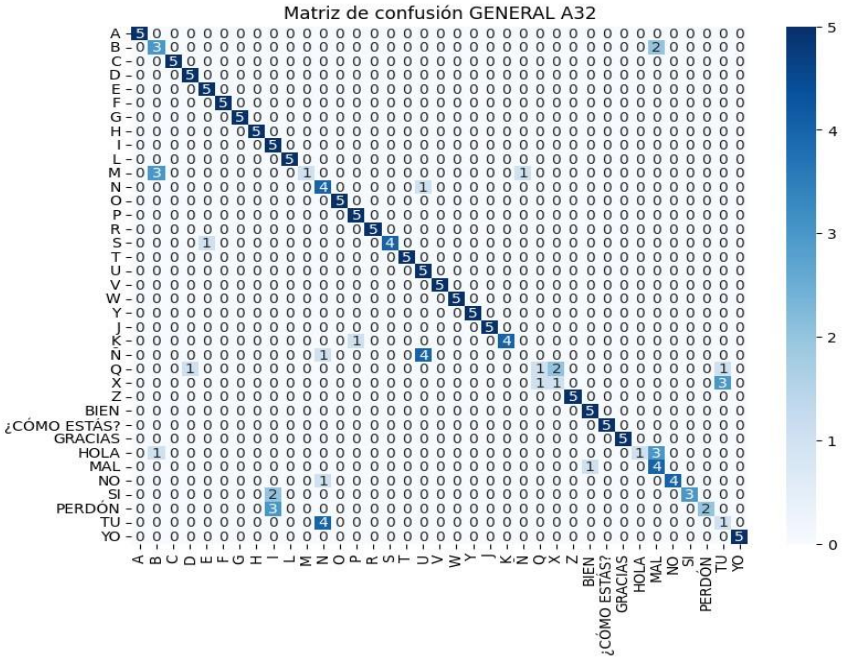

Figura 9. Matriz de confusión A32.

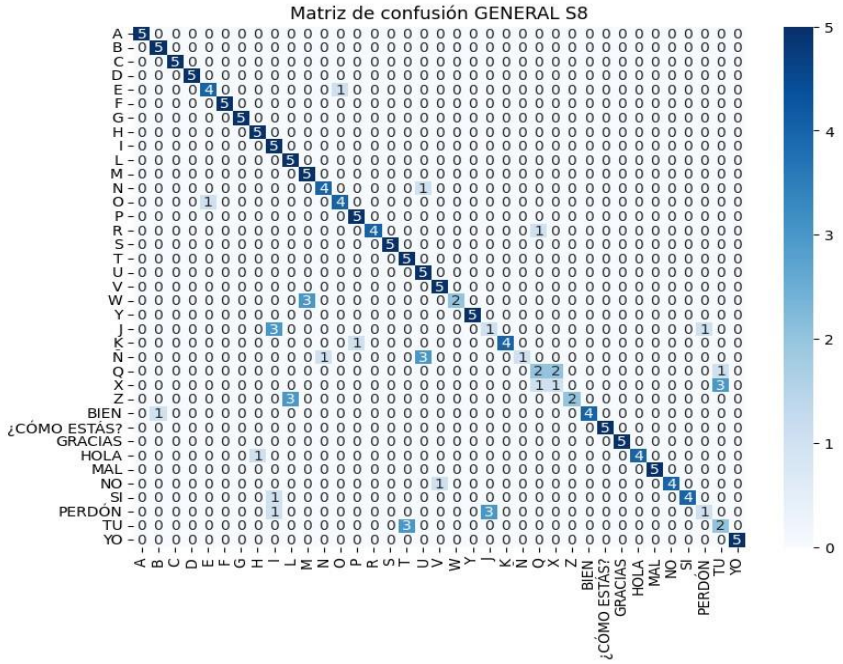

Figura 10. Matriz de confusión S8.

En conclusión, se puede observar que la diferencia de capacidades técnicas entre los dispositivos sí determina la calidad de caracterización del vídeo (mediante el cálculo de las distancias-DTW). Sin embargo, esto no fue tan evidente en los resultados de desempeño de clasificación del sistema propuesto de reconocimiento automático de señas de la LSM, debido al criterio de agrupamiento en la selección de señas.

4.2 Señas estáticas versus dinámicas

Para determinar cuáles señas de la LSM, estáticas o dinámicas, fueron más fáciles de diferenciar por el sistema de reconocimiento propuesto, se elaboraron diversas tablas comparativas. Las Tablas 8 y 9 muestran los resultados en el desempeño en clasificación para las señas estáticas y dinámicas de la LSM, respectivamente, utilizando cada uno de los dispositivos móviles. En ellas se observa que S8 presenta una mayor exactitud para las señas estáticas, seguido por A32 y después P40. Cabe destacar que la exactitud de 0.9333 obtenida por S8, es incluso superior a la de otros sistemas diseñados exclusivamente para las señas estáticas de la LSM, e.g., [8]; y no tan significativamente distante del mejor resultado para este tipo de señas (ver Tabla 1).

En contraste, se tiene que S8 tiene un peor desempeño para las señas dinámicas, siendo el P40 el que reportó una mejor exactitud. Esta diferencia de desempeño podría deberse a que S8 graba a una tasa de 60fps, por lo que al momento de procesar se generan más comparaciones dando cabida a que las distancias-DTW sean mayores. Además, la exactitud de 0.7875 obtenida por P40 en señas dinámicas dista del desempeño reportado por otros trabajos de reconocimiento para este tipo de señas, como [14-18]. Esto puede deberse a las condiciones variables de captura de los vídeos respecto a la iluminación, los ángulos de visión y los fondos y colores de vestimenta no controlados. También, se observó en algunos vídeos que la grabación incluía un movimiento adicional de las manos previo a la transición inicial de la seña, lo que ocasionaba similitudes con otras señas.

Tabla 8. Rendimiento general del modelo por dispositivo para señas estáticas.

Métrica/ Celular	**P40**	**S8**	**A32**	**Promedio**
Precisión	0.8216	0.9117	0.8768	0.8700
Exhaustividad	0.7416	0.8909	0.8434	0.8253
Exactitud	0.8476	0.9333	0.9238	0.9015
Valor F	0.7532	0.8921	0.8467	0.8306

Tabla 9. Rendimiento general del modelo por dispositivo para señas dinámicas.

Métrica/ Celular	**P40**	**S8**	**A32**	**Promedio**
Precisión	0.6812	0.5233	0.5653	0.5833
Exhaustividad	0.6300	0.4000	0.4636	0.4978
Exactitud	0.7875	0.625	0.6375	0.6833
Valor F	0.6385	0.4388	0.4883	0.5218

Cabe resaltar que se esperaba que el sistema obtuviera mejores resultados en la clasificación de señas dinámicas, debido a que el método DTW para la comparación de series de tiempo es empleado en otros trabajos de reconocimiento para este tipo de señas. Sin embargo, los resultados muestran un mejor desempeño para las señas estáticas y esto se cree que ocurre debido a los siguientes factores:

1) El poco o nulo movimiento en los vídeos de las señas estáticas refleja una baja complejidad y menor variabilidad temporal en comparación con las señas dinámicas. Esto sugiere que las distancias-DTW entre vídeos de una misma seña son menores en las señas estáticas que en las dinámicas.
2) Una seña dinámica requiere un movimiento desde un estado inicial de la configuración de la seña hasta un estado final. Sin embargo, la transición desde el inicio del vídeo hacia dicho estado inicial puede incluir otros movimientos similares a los de otras señas, lo que aumenta el "ruido" y la posibilidad de errores.
3) Dado que la caracterización de las señas es invariante a tamaño, posición absoluta y rotación (ver Sección 2.3.2), algunas señas dinámicas pueden compartir configuraciones similares a otras señas durante un periodo de tiempo determinado. Un caso particular es la seña dinámica Ñ, que el sistema confundió con la seña estática U en nueve ocasiones. Se conjetura que esta confusión ocurrió porque los vídeos generados para la U capturaron el movimiento hacia la configuración inicial de la seña, lo que pudo provocar el error.

Los resultados de las distancias-DTW para cada seña, estática o dinámica se compararon utilizando la prueba de Kruskall & Wallis para los dispositivos móviles cuando estos acertaron en todas sus predicciones, i.e., cuando se tienen "resultados unánimes". Los valores de probabilidad fueron 0.0049 y 0.0480 para señas estáticas y dinámicas, respectivamente. Se obtuvo como resultado que al menos uno de los dispositivos provee distancias-DTW significativamente diferentes para ambos tipos de señas. También se aplicó la prueba Dunn, donde se identificó que la diferencia significativa radica entre S8 y A32.

Por otro lado, se compararon las distancias-DTW entre predicciones de señas estáticas versus dinámicas independientemente de los dispositivos móviles. Aplicando una prueba de t-student para ambas muestras, se obtiene un valor de prueba de -3.2497 y un valor de probabilidad de 0.0012. Esto indica una diferencia significativa entre el cálculo de las distancias-DTW para cada seña. El promedio de las distancias-DTW es de 96,978.56 y de 114,627.91 para señas estáticas y dinámicas, respectivamente.

Por otro lado, se compararon las distancias-DTW entre predicciones de señas estáticas versus dinámicas independientemente de los dispositivos móviles. La Figura 11 muestra los resultados de las predicciones de señas dinámicas para P40. En este caso, el sistema fue más preciso al momento de clasificar los vídeos de entrenamiento, ya que solamente toma cuatro señas estáticas cuando predice de manera incorrecta las letras D, I, P y U. Las señas no pertenecientes al alfabeto tuvieron un mejor resultado que aquellas que sí pertenecen. Las incidencias en señas que se predijeron incorrectamente ocurrieron más en letras del alfabeto que en palabras.

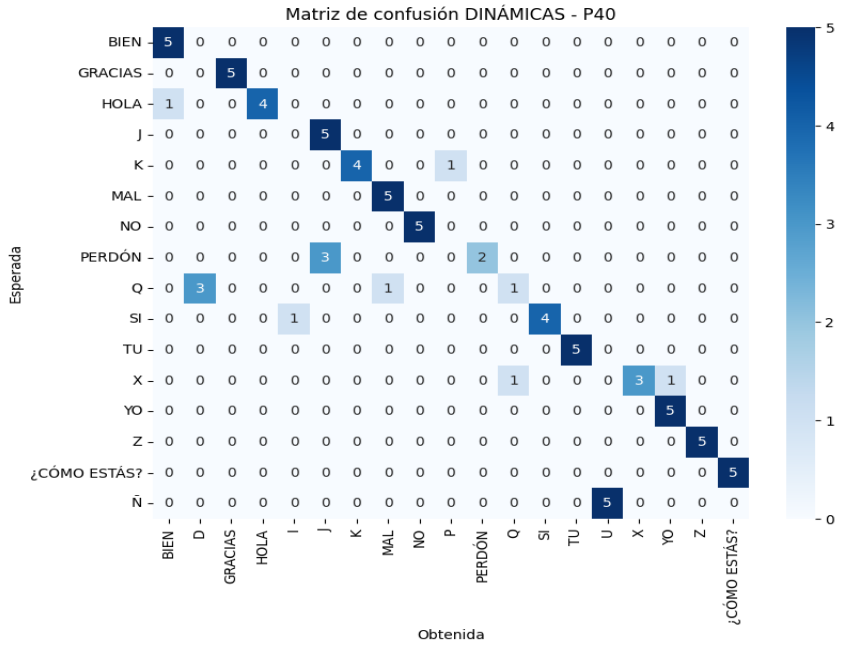

Figura 11. Matriz de confusión de señas dinámicas para P40.

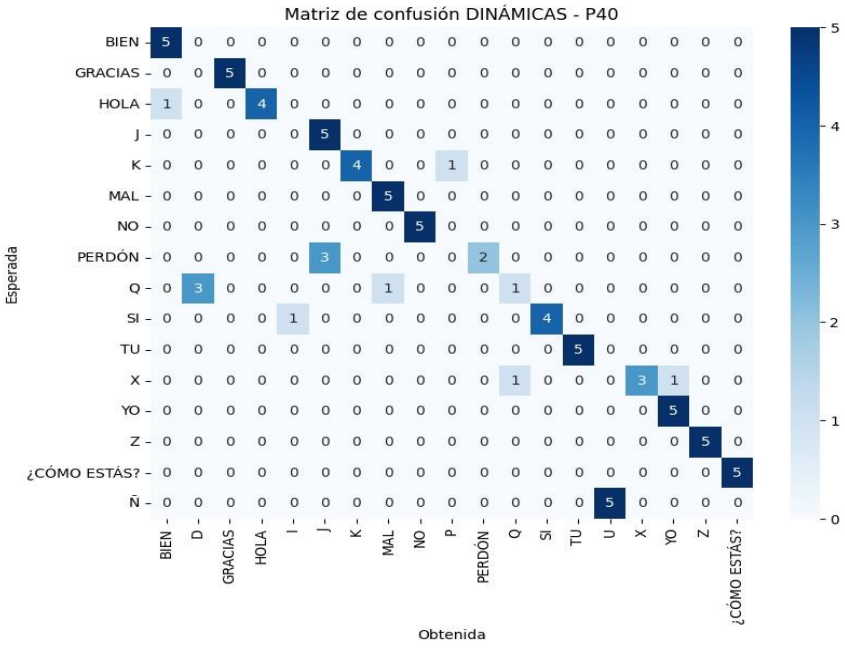

Figura 12. Matriz de confusión de señas estáticas de P40.

La Figura 12 muestra los resultados de las predicciones de señas estáticas para P40. En este caso, el sistema utilizó más señas dinámicas cuando clasificaba erróneamente: K, MAL y Ñ. En esta instancia, se observa que el sistema empleó un mayor número de señas dinámicas al clasificar de manera incorrecta, específicamente las letras K, MAL y Ñ. En la mayoría de las pruebas realizadas con este dispositivo, las señas incorrectamente clasificadas coincidieron mayormente con aquellas que comparten ciertas características de configuración.

La Figura 13 muestra los resultados de las predicciones de señas dinámicas A32. En este escenario, al realizar la clasificación de los videos de entrenamiento, el sistema identifica erróneamente seis señas estáticas: B, D, I, N, P y U. De manera análoga a lo observado en el caso de P40, las señas que no forman parte del alfabeto presentaron un rendimiento superior en comparación con aquellas que sí lo conforman. Además, las instancias de predicciones incorrectas también se inclinaron más hacia letras del alfabeto que hacía palabras.

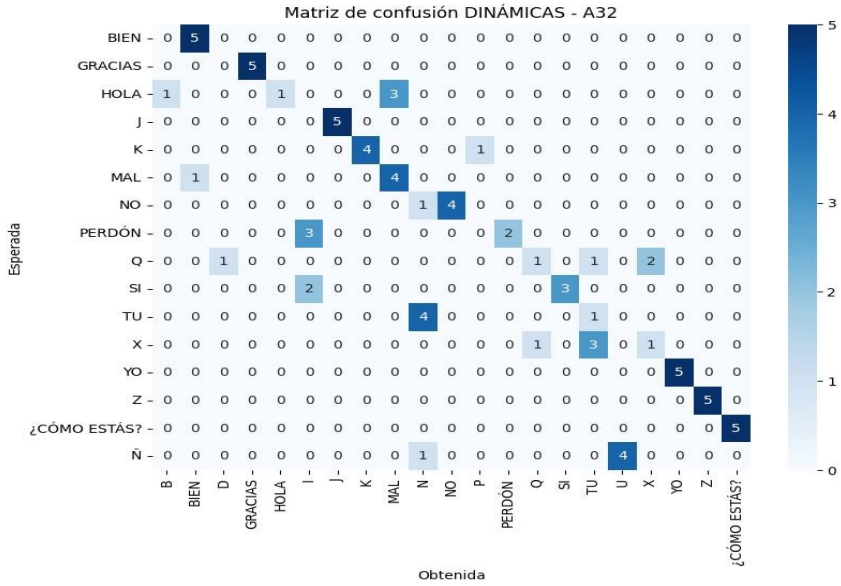

Figura 13. Matriz de confusión de señas dinámicas de A32.

La Figura 14 muestra los resultados de las predicciones de señas estáticas para A32. En este caso, el sistema utilizó menos señas dinámicas cuando clasificaba erróneamente: MAL y Ñ. De forma similar a P40, las señas erróneas en general fueron predichas con aquellas cuyos ángulos en los puntos de referencia son similares; e.g., las señas U, N y Ñ tienen ángulos similares en diferente perspectiva.

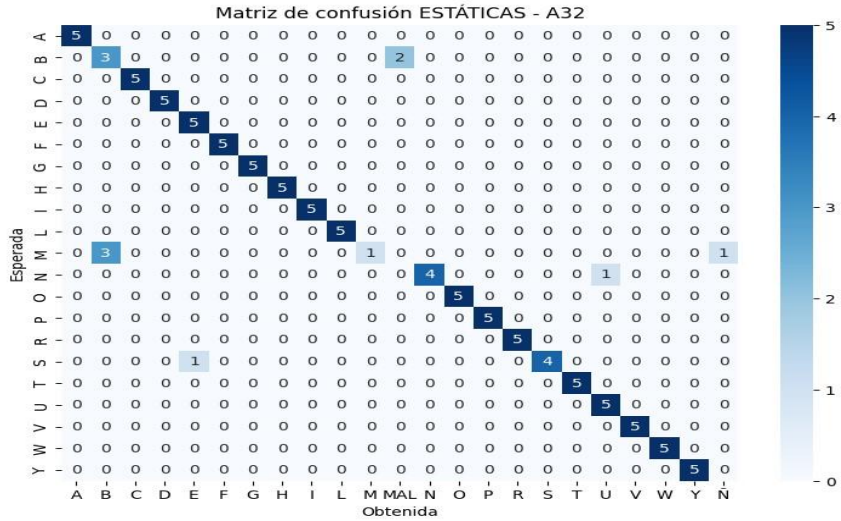

Figura 14. Matriz de confusión de señas estáticas de A32.

La Figura 15 muestra los resultados de las predicciones de señas dinámicas para S8. En este caso, se predicen más señas estáticas de manera incorrecta, en particular: B, H, I, L, N, P, U y V. Al igual que en los casos de P40 y A32, las señas que no forman parte del alfabeto mostraron un desempeño superior en comparación con aquellas que sí pertenecen. Las instancias de predicciones incorrectas también se inclinaron más hacia letras del alfabeto que hacía palabras.

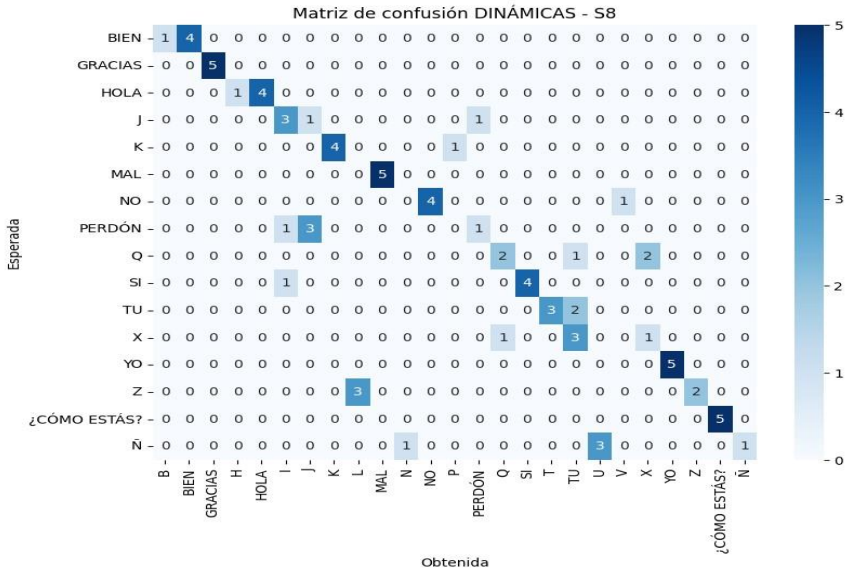

Figura 15. Matriz de confusión de señas dinámicas de S8.

La Figura 16 muestra los resultados de las predicciones de señas estáticas para S8. En este caso, el sistema registró un menor número de clasificaciones erróneas, con excepción de la seña Q. Siguiendo una pauta similar a los resultados obtenidos en P40 y A32, en la mayoría de las pruebas realizadas con S8 se evidencia que las señas erróneas fueron asociadas mayormente con aquellas que comparten ángulos similares en sus puntos de referencia.

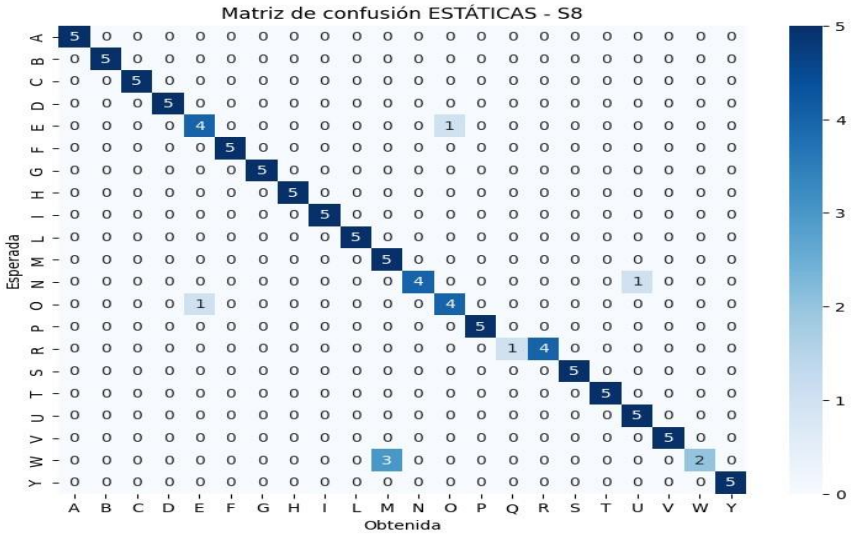

Figura 16. Matriz de confusión de señas estáticas de S8.

Tabla 10. Exactitud individual para cada seña.

Seña	Exactitud
A, F, G, H, I, L, V, Y, GRACIAS, YO, ¿CÓMO ESTAS?	100%
E, O, R, T, BIEN, MAL	93%
B, S, NO	87%
C, N, P, U, W, K, Z	80%
D	79%
J, SI	73%
HOLA	60%
TU	53%
M	47%
X, PERDON	33%
Q	27%
Ñ	7%

Se observa que, en general, la exactitud en la predicción de cada seña está influenciada principalmente por la diferencia entre los ángulos de los puntos de referencia, así como también por la dificultad en la configuración de la seña; e.g., la seña ¿CÓMO ESTÁS? utiliza ambas manos y contrasta muy bien con el resto de las señas, lo que no ocurre con las señas U, N y Ñ. Adicionalmente, también influye la exactitud con la que se generaron los vídeos de entrenamiento y las capacidades de los sujetos de prueba para realizar las señas. La Tabla 10 muestra la exactitud en la predicción de cada seña.

Finalmente, cabe mencionar que, en algunos casos, la lista generada durante el proceso de selección de la seña muestra una diferencia mínima entre las distancias-DTW de las dos primeras opciones. Además, la segunda opción por lo general representa la seña correcta. Esto indica que el sistema de reconocimiento propuesto puede mejorar su desempeño de clasificación utilizando otras políticas de selección de señas.

5. Conclusiones

En el presente trabajo, se desarrolló e implementó un sistema de reconocimiento automático de señas de la LSM para dispositivos móviles. El objetivo fue evaluar la viabilidad de un sistema de este tipo utilizando equipos convencionales en situaciones cotidianas. Además, se analizó el desempeño del sistema considerando las diferencias en el reconocimiento de señas estáticas y dinámicas de la LSM.

Se llevó a cabo un estudio experimental con tres dispositivos móviles (P40, S8 y A32), los cuales presentan distintas capacidades y calidades de cámara, para recolectar un conjunto de vídeos de señas de la LSM. Los resultados experimentales muestran que, en general, las capacidades técnicas de los dispositivos impactan en la caracterización de las señas de la LSM, medida a través de las distancias-DTW. Aunque los resultados en general son inferiores en desempeño en comparación con otros trabajos (ver Tabla 1), la exactitud promedio de 0.8072 obtenida indica la viabilidad de este tipo de sistemas en dispositivos convencionales. Sin embargo, persisten diversos desafíos que deben considerarse y superarse, como las condiciones de iluminación, el ruido de fondo y las limitaciones de procesamiento y energía en estos dispositivos. Este estudio es un precedente en el reconocimiento de señas de la LSM en el contexto de estas restricciones.

Una observación destacable fue que el sistema reconoció con mayor exactitud las señas estáticas en comparación con las dinámicas (ver Sección 4.2). Esto se debe, en parte, a la mayor perturbación en la configuración espacial de las señas dinámicas, como el ángulo de las regiones de interés durante la captura. Las señas estáticas, al no involucrar movimiento, presentaron menores distancias-DTW, lo que sugiere una mayor consistencia en su caracterización. Aunque el DTW se emplea comúnmente para la comparación de secuencias temporales y es particularmente útil en señas dinámicas, los resultados sugieren que aplicar este enfoque a las señas estáticas podría mejorar la precisión de los sistemas actuales.

Este estudio también reveló áreas de oportunidad en el proceso de selección de señas. Se utilizó una política de agrupamiento que, aunque adecuada, podría ser refinada para mejorar la exactitud de las predicciones. Un enfoque más preciso en la política de selección de señas, que tome en cuenta tanto las distancias-DTW como otras características específicas de las señas, tiene el potencial de optimizar el sistema.

Como trabajo futuro, se propone implementar el uso de la posición y perspectiva de las señas, lo que permitiría una diferenciación más eficiente de las señas capturadas durante el proceso de caracterización. También sería interesante profundizar en la comparación entre señas estáticas y dinámicas utilizando técnicas adicionales al DTW. Además, se sugiere desarrollar un nuevo procedimiento de selección de señas con diferentes políticas en relación con las distancias-DTW, que se ajusten mejor a los casos. Finalmente, se propone paralelizar el proceso de comparación para reducir el tiempo de ejecución y lograr que la aplicación tenga un comportamiento aceptable en contextos cotidianos.

6. Agradecimientos

La presente investigación fue financiada parcialmente por: CONAHCYT a través de proyecto 319143 bajo la dirección de la Dra. Silvia Berenice Fajardo Flores, y por el Tecnológico Nacional de México (TecNM) a través del proyecto 18205.23-P bajo la dirección del Dr. Daniel Fajardo Delgado.

Abreviaciones

Las siguientes abreviaciones se utilizan a lo largo del documento:

LSM – Lengua de señas mexicana.
DTW (*Dynamic Time Wrapping*) – Deformación dinámica del tiempo.
OpenCV (*Open Computer Vision Library*) – Librería de código abierto para visión por computadora.
HTTP (*Hypertext Transfer Protocol*) – Protocolo de transferencia de hipertexto.
SDK (*Estándar Development Kit*) – Kit de desarrollo estándar.
URL (*Uniform Resource Locator*) – Localizador de recursos uniforme.
CSV (Comma-Separated Values) – Formato de archivo con valores separados por comas.

7. Referencias

[1] INEGI (2020). Población con discapacidad o limitación en la actividad cotidiana por entidad federativa y tipo de actividad realiza según sexo, 2020. https://www.inegi.org.mx/app/tabulados/interactivos/?pxq=Discapacidad_Discapacidad_02_2c111b6a-6152-40ce-bd39-6fab2c4908e3&idrt=151&opc=t.

[2] Diario Oficial de la Federación (2005). Ley general de las personas con discapacidad, publicada el 10 de Junio de 2005, modificada el 1 de Agosto de 2008 y abrogada el 30 de Mayo de 2011. *Diario Oficial de la Federación, Órgano del Gobierno Constitucional de los Estados Unidos*

Mexicanos. https://www.dof.gob.mx/nota_detalle.php?codigo=2044351&fecha=10/06/2005#gsc.tab=0

[3] Cruz-Aldrete, M. (2014). Hacia la construcción de un diccionario de lengua de señas mexicana. *Revista de Investigación*, 38(83), pp. 57–80.

[4] Cámara de Diputados (2021). Aprueban reformas para que personas con discapacidad auditiva reciban educación bilingüe en lengua de señas. *Boletín* No.5854 de la Cámara de Diputados del Congreso de la Unión de México.

[5] Pérez-Castro, J., & Cruz-Cruz, J.C. (2021). Inclusion-exclusion experiences of a group of deaf people users of the Mexican sign language. *Revista latinoamericana de educación inclusiva*, 15(1), pp. 39–54. https://dx.doi.org/10.4067/S0718-73782021000100039

[6] Flores-Saldaña, M. C., Cruz-Aldrete, M., Guajardo-Ramos, E., & Moreno-Aguirre, A. J. (2022). Percepción de inclusión educativa y atención a la salud de personas sordas en México. *Revista ConCiencia EPG*, 7(2), pp. 16–29. https://doi.org/10.32654/CONCIENCIAEPG.7-2.2

[7] Ríos-Figueroa, H. V., Sánchez-García, A. J., Sosa-Jiménez, C. O., y Solís-González Cosío, A. L. (2022). Use of spherical and cartesian features for learning and recognition of the static Mexican sign language alphabet. *Mathematics*, 10(16), 2904. https://doi.org/10.3390/math10162904

[8] Rodríguez, R. F., Rosas, F. J. P., Zuñiga-Madrid, L. Á., & Arguijo, P. (2021) Reconocimiento de las señas estáticas del LSM con características basadas en aprendizaje profundo. *Research in Computing Science*, 150(6), pp. 303–311.

[9] Fregoso, J., Gonzalez, C. I., & Martinez, G. E. (2021). Optimization of convolutional neural networks architectures using PSO for sign language recognition. *Axioms*, *10*(3), 139. https://doi.org/10.3390/axioms10030139

[10] García-Gil, G., López-Armas, G. D. C., Sánchez-Escobar, J. J., Salazar-Torres, B. A., & Rodríguez-Vázquez, A. N. (2024). Real-time machine learning for accurate Mexican sign language identification: A distal phalanges approach. *Technologies*, *12*(9), 152. https://doi.org/10.3390/technologies12090152

[11] Morfín-Chávez, R. F., Gortarez-Pelayo, J. J., & Lopez-Nava, I. H. (2024). Fingerspelling recognition in Mexican sign language (LSM) using machine learning. En: Calvo, H., Martínez-Villaseñor, L., Ponce, H. (eds) *Advances in Computational Intelligence*. MICAI 2023. Lecture Notes in Computer Science, vol 14391, pp. 110–120. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-47765-2_9

[12] Espejel-Cabrera, J., Cervantes, J., García-Lamont, F., Ruiz Castilla, J. S., & Jalili, L. D. (2021). Mexican sign language segmentation using color based neuronal networks to detect the individual skin color. *Expert Systems with Applications*, 183, 115295. https://doi.org/10.1016/j.eswa.2021.115295

[13] Mejía-Peréz, K., Córdova-Esparza, D. M., Terven, J., Herrera-Navarro, A. M., GarcíaRamírez, T., & Ramírez-Pedraza, A. (2022). Automatic recognition of Mexican sign language using a depth camera and recurrent neural networks. *Applied Sciences*,

12(11), 5523. https://doi.org/10.3390/app12115523

[14] Trujillo-Romero, F., & Caballero-Morales, S.-O. (2013). 3D Data sensing for hand pose recognition. En: *CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing, Cholula, Puebla, Mexico,* pp. 109–113. https://doi.org/10.1109/CONIELECOMP.2013.6525769

[15] Solis-V, J. F., Toxqui-Quitl, C., Martínez-Martínez, D., & Margarita, H. G. (2014). Mexican sign language recognition using normalized moments and artificial neural networks. En: Awwal A.A.S, Iftekharuddin, K.M, Matin, M.A., Márquez, A. (eds), Optics and Photonics for Information Processing VIII, *SPIE Proceedings*, Vol. 9216, 92161A. https://doi.org/10.1117/12.2061077

[16] García-Bautista, G., Trujillo-Romero, F., & Caballero-Morales, S. O. (2017). Mexican sign language recognition using kinect and data time warping algorithm. En: *CONIELECOMP 2017, 23rd International Conference on Electronics, Communications and Computing, Cholula, Puebla, Mexico,* pp. 1–5. https://doi.org/10.1109/CONIELECOMP.2017.7891832

[17] Sosa-Jiménez, C.O., Ríos-Figueroa, H.V., & Solís-González-Cosío, A.L. (2022). A prototype for Mexican sign language recognition and synthesis in support of a primary care physician, *IEEE Access*, 10, pp. 127620–127635. https://doi.org/10.1109/ACCESS.2022.3226696

[18] González-Rodríguez, J. R., Córdova-Esparza, D. M., Terven, J., & Romero-González, J. A. (2024). Towards a bidirectional Mexican sign language–spanish translation system: A deep learning approach. *Technologies*, 12(1), 7. https://doi.org/10.3390/technologies12010007

[19] Instituto Federal de Telecomunicaciones (2021). Encuesta nacional sobre disponibilidad y uso de tecnologías de la información en los hogares. https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/OtrTemEcon/ENDUTIH_2020.pdf. Recuperado el día 21 de Junio de 2023.

[20] Martinez-Seis, B., Pichardo-Lagunas, O., Rodriguez-Aguilar, E., & Saucedo-Diaz, E.-R. (2019). Identification of static and dynamic signs of the Mexican sign language alphabet for smartphones using deep learning and image processing. *Research in Computing Science*, 148(11), pp. 199–211.

[21] Pham, C. H., Le, Q. K., & Le, T. H. (2014). Human action recognition using dynamic time warping and voting algorithm. *VNU Journal of Science: Computer Science and Communication Engineering*, 30(3).

[22] Guerin, G. (2021). Sign language recognition - using mediapipe dtw. https://www.sicara.fr/blog-technique/sign-language-recognition-using-mediapipe. Recuperado el día 21 de Junio de 2023.

[23] Howse, J., & Minichino, J. (2020). Learning OpenCV 4 computer vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning. Packt Publishing Ltd.

[24] CloudRun (2023). ¿Qué es cloud run? https://cloud.google.com/run/docs/overview/what-is-cloud-run?hl=es-419. Recuperado el día 21 de Junio de 2023.

[25] MediaPipe (2023). Mediapipe solutions guide. https://developers.google.com/mediapipe/. Recuperado el día 21 de Junio de 2023.

[26] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. *Data mining and knowledge discovery*, 33(4), pp. 917–963. https://doi.org/10.1007/s10618-019-00619-1

[27] (2007). Dynamic time warping, En: Information Retrieval for Music and Motion. Springer, Berlin, Heidelberg. pp. 69–84. https://doi.org/10.1007/978-3-540-74048-3_4

[28] Salvador, S. & Chan, P. (2004). FastDTW: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5), pp. 561–580. https://doi.org/10.3233/IDA-2007-11508

[29] Kruskal, W. H. & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260), pp. 583–621. https://doi.org/10.2307/2280779

[30] Dunn, O. J. (1964). Multiple comparisons using rank sums. *Technometrics*, 6(3), pp. 241–252. https://doi.org/10.2307/1266041

[31] Student (1908). The probable error of a mean. *Biometrika*, 6(1), pp. 1–25. https://doi.org/10.2307/2331554