La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).
La investigación fue financiada por el CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías).

Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Maestría en Ciencias
en Ciencias de la Computación

Reconocimiento continuo de la Lengua de Señas Mexicana

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de

Maestro en Ciencias

Presenta:

Ricardo Fernando Morfín Chávez

Ensenada, Baja California, México

2023

Tesis defendida por

Ricardo Fernando Morfín Chávez

y aprobada por el siguiente Comité

Dr Irvin Hussein López Nava

Director de tesis

Dr. Jesús Favela Vara

Dr. Salvador Villareal Reyes

Dr. Pedro Gilberto López Mariscal

Coordinador del Posgrado en Ciencias de la Computación

Dra. Ana Denise Re Araujo

Directora de Estudios de Posgrado

Resumen de la tesis que presenta Ricardo Fernando Morfín Chávez como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Reconocimiento continuo de la Lengua de Señas Mexicana

Resumen aprobado por:

Dr Irvin Hussein López Nava
Director de tesis

La Lengua de Señas Mexicana (LSM) es la lengua utilizada por la comunidad Sorda en México, y, a menudo, subestimada y pasada por alto por la comunidad oyente, lo que resulta en la exclusión sistemática de las personas Sordas en diversos aspectos de la vida. Sin embargo, la tecnología puede desempeñar un papel fundamental en acercar a la comunidad Sorda con la comunidad oyente, promoviendo una mayor inclusión y comprensión entre ambas. El objetivo principal de este trabajo es diseñar, implementar y evaluar un sistema de reconocimiento continuo de señas estáticas en LSM mediante, visión por computadora y técnicas de aprendizaje máquina. Se establecieron objetivos específicos, que incluyen la generación de un conjunto de datos de señas estáticas, pertenecientes al alfabeto manual de la LSM, el diseño de un modelo de reconocimiento, y la evaluación del sistema, tanto en la modalidad aislada como en la continua. La metodología involucra dos evaluaciones distintas. La primera se enfoca en el reconocimiento de señas estáticas en el dominio aislado, para ello se capturaron datos de 20 participantes realizando movimientos de la mano en múltiples ángulos. Se evaluaron diversas técnicas de aprendizaje automático, destacando que el enfoque basado en Máquinas de Soporte Vectorial (SVM) obtuvo los mejores resultados (F1-Score promedio del 0.91). La segunda evaluación se concentra en el reconocimiento continuo de señas estáticas, con datos recopilados de seis participantes con diferentes niveles de competencia en LSM, logrando un rendimiento sólido con errores cercanos al 7 %. Además, se evaluó la viabilidad del sistema en aplicaciones de tiempo real, demostrando un excelente desempeño (velocidad promedio de procesamiento de 45 cuadros por segundo). A pesar de los logros alcanzados, es importante reconocer que este proyecto se centró en el reconocimiento continuo de señas estáticas en LSM. Queda pendiente, como un desafío interesante, la exploración del reconocimiento continuo de señas dinámicas en LSM para futuras investigaciones. Se considera esencial explorar enfoques orientados a la escalabilidad y aplicaciones en tiempo real en investigaciones posteriores.

Palabras clave: Lengua de Señas Mexicana (LSM), visión por computadora, aprendizaje automático, alfabeto manual de la LSM, reconocimiento automático de señas estáticas, reconocimiento aislado de señas, reconocimiento continuo de señas, aplicaciones en tiempo real

Abstract of the thesis presented by Ricardo Fernando Morfín Chávez as a partial requirement to obtain the Master of Science degree in Computer Science.

Continuous recognition of Mexican Sign Language

Abstract approved by:

PhD Irvin Hussein López Nava
Thesis Director

This study focuses on the continuous recognition of static signs in Mexican Sign Language (Lengua de Señas Mexicana (LSM)), the language used by the Deaf community in Mexico. Despite its significance, LSM is often underestimated and overlooked, leading to the systematic exclusion of Deaf individuals in various aspects of life. The primary objective of this work is to design, implement, and evaluate a continuous static sign recognition system in LSM using computer vision and machine learning techniques. Specific goals were established, including the creation of a dataset of static signs belonging to the manual alphabet of LSM, the design of a recognition model, and the evaluation of the system in both isolated and continuous modes. The methodology involves two distinct evaluations. The first one focuses on the recognition of static signs in the isolated domain, for which data from 20 participants performing hand movements at various angles were collected. Various machine learning techniques were evaluated, with the Máquinas de Soporte Vectorial (SVM)-based approach achieving the best results (average F1-Score of 0.91). The second evaluation centers on the continuous recognition of static signs, using data collected from six participants with varying levels of competence in LSM, achieving robust performance with errors close to 7 %. Furthermore, the feasibility of the system in real-time applications was assessed, demonstrating excellent performance (average processing speed of 45 frames per second). Despite the achievements, it is important to recognize that this project focused on continuous recognition of static signs in LSM. It remains an interesting challenge to explore the continuous recognition of dynamic signs in LSM for future research. It is considered essential to explore scalability-oriented approaches and real-time applications in subsequent investigations.

Keywords: Mexican Sign Language (LSM), computer vision, machine learning, LSM manual alpahbet, automatic recognition of static signs, isolated sign recognition, continuous sign recognition, real-time aplications

Dedicatoria

A mis tres mejores amigos, Armando, Estela y Jorge Armando.

Agradecimientos

Al Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE) por brindarme el espacio, las herramientas y las facilidades necesarias para completar este trabajo.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por brindarme el apoyo económico para realizar mis estudios de maestría.

A la Asociación Regional de Sordos Ensenadenses, en especial a los profesores Carlos, Sebastián y Antonio. Gracias a sus clases, pude adquirir conocimientos fundamentales de la Lengua de Señas Mexicana, además de obtener una mayor comprensión de la cultura sorda. Su generosa enseñanza ha sido fundamental para el desarrollo de esta tesis.

A todos los voluntarios que ayudaron en la captura de datos.

A mi director de tesis Irvin Hussein López Nava por sus valiosas aportaciones, su tiempo y consejos.

A mi comité de tesis, Jesús Favela Vara y Salvador Villareal Reyes, por sus observaciones y correcciones.

Al grupo de senderismo Pelícanos Viajeros y al Club de Atletismo de CICESE. Después de tantos kilómetros recorridos y experiencias inolvidables, no puedo más que reconocer lo importante que han sido en mi camino por CICESE. Su compañía y motivación han dejado una huella imborrable en mi vida.

A mis padres, por su amor incondicional, por las deliciosas comidas y los cálidos abrazos, sin ustedes esta tesis no hubiese posible. A mi hermano, a pesar de estar a cientos de kilómetros de distancia, siento tu apoyo y cariño en cada paso que doy. ¡Simplemente gracias!

A todos los amigos que me acompañaron durante este viaje.

> "No solo no hubiéramos sido nada sin ustedes, sino con toda la gente que estuvo a nuestro alrededor desde el comienzo; algunos siguen hasta hoy. ¡Gracias... totales!"
>
> - Gustavo Cerati, 20 de septiembre de 1997.

Tabla de contenido

Página

Lista de figuras

Figura Página

Figura Página

Figura Página

Lista de tablas

Tabla Página

Capítulo 1. Introducción

1.1. Motivación

La Lengua de Señas Mexicana (LSM) es una lengua visual-gestual utilizada por la comunidad Sorda[1] mexicana. La LSM es una lengua natural y como tal cuenta con su propia sintaxis, gramática y léxico. A través del movimiento de las manos, las expresiones faciales y gestos corporales, la LSM permite a las personas Sordas comunicarse de manera efectiva entre sí y con personas oyentes que conocen esta lengua.

Según la Organización Mundial de la Salud (OMS), las personas *sordas* suelen padecer una pérdida de audición profunda, lo que significa que oyen muy poco o nada. Para comunicarse suelen utilizar alguna lengua de señas [2]. Por otro lado, se entiende por *hipoacúsicos* a las personas que presentan una pérdida parcial de la capacidad auditiva, que va desde leve hasta grave. Por lo general se comunican con lenguas orales y pueden beneficiarse de dispositivos de asistencia como audífonos o implantes cocleares[3].

Es complicado estimar el número de personas sordas en el mundo, ya que las cifras proporcionadas por la OMS indican que más del 5 % de la población global (o 430 millones de personas) sufre alguna pérdida auditiva. Sin embargo, esta cifra no distingue entre personas sordas e hipoacúsicas. Por otra parte, de acuerdo con la Federación Mundial de Sordos, conocida como World Federation of the Deaf, en el año 2016 se estimaba que existían alrededor de 70 millones de personas sordas en el mundo[4].

En el caso específico de México, se estima que 4.2 millones de personas tienen alguna limitación o discapacidad auditiva, de las cuales 2.9 millones presentan una limitación auditiva (con poca dificultad para oír) y 1.3 millones tienen una discapacidad auditiva (con mucha dificultad para oír o pérdida total) según datos del Instituto Nacional de Estadística y Geografía (INEGI) en 2020 [5].

Además, las estadísticas sobre la discapacidad auditiva a menudo no hacen distinción entre sordos prelingüísticos y sordos postlingüísticos. Los sordos prelingüísticos son aquellos que experimentaron una

[1] Por convención, *Sordo, Sorda* con letra mayúscula se emplea para referirse a la comunidad silente que comparte una lengua de señas. Por otra parte, *sordo, sorda* hace referencia a una condición auditiva (Cruz-Aldrete, 2008).

[2] Existen varias alternativas de sistemas de comunicación empleados por individuos con discapacidad auditiva, principalmente: El uso de lenguas orales, también conocido como oralización, el uso de Lenguas de Señas y la comunicación mediante señas caseras.

[3] `https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss`

[4] `https://wfdeaf.org/our-work/human-rights-of-the-deaf/`

[5] `https://www.inegi.org.mx/app/tabulados/interactivos/?pxq=Discapacidad_Discapacidad_02_6d37a7a0-9485-4f9e-85dd-5484f7786e86`

discapacidad auditiva antes de adquirir una lengua oral como su lengua materna. Por otro lado, los sordos postlingüísticos son aquellos que experimentaron una disminución o pérdida de la audición después de haber adquirido una lengua oral como lengua materna debido a diversas causas (Cruz-Aldrete, 2008).

Las personas sordas enfrentan obstáculos significativos en su vida diaria debido a la falta de accesibilidad y comprensión por parte de la sociedad. La falta de conocimiento y reconocimiento de la LSM como un lenguaje completo y autónomo ha llevado a una exclusión sistemática de la comunidad sorda en ámbitos como la educación, el empleo y la atención médica. Esta situación crea barreras adicionales para su desarrollo personal, educativo y profesional.

1.2. Antecedentes

Para abordar el problema de reconocimiento de señas de la Lengua de Señas Mexicana, es importante estudiar sus componentes y enfocarnos en proponer soluciones específicas para cada uno de ellos. En particular, se explicará la diferencia entre las señas estáticas y las señas dinámicas, así como las diferencias entre el reconocimiento aislado y el reconocimiento continuo.

Diferencia entre señas estáticas y dinámicas. La diferencia entre las señas estáticas y dinámicas radica en el movimiento de las manos y el cuerpo. Las señas estáticas se refieren a gestos que se mantienen en una posición fija sin un movimiento continuo, mientras que las señas dinámicas implican un movimiento fluido y continuo de las manos y el cuerpo para expresar un significado. Ambos tipos de señas son fundamentales para tener una comunicación efectiva en la Lengua de Señas Mexicana.

Diferencia entre el reconocimiento aislado y continuo. El reconocimiento aislado se refiere a la capacidad de identificar y clasificar señas individuales de manera independiente. En este enfoque, cada seña se analiza por separado, sin considerar su contexto en una secuencia continua. Por lo tanto, el reconocimiento aislado se centra en determinar la correspondencia entre una seña específica y su significado o etiqueta correspondiente. Por otro lado, el reconocimiento continuo implica la capacidad de interpretar señas en una secuencia continua, teniendo en cuenta el contexto y la transición fluida entre señas. En lugar de analizar señas individuales de manera aislada, el reconocimiento continuo busca captar la estructura y el flujo natural de las señas en un contexto más amplio.

El desafío que aborda esta tesis es lograr el reconocimiento continuo de señas estáticas de la Lengua de Señas Mexicana, para ello se busca desarrollar un sistema de reconocimiento automatizado que pueda interpretar las señas estáticas en una secuencia continua, garantizando una comunicación fluida y efectiva en el contexto de la LSM.

En el contexto de la comunicación en Lengua de Señas Mexicana, es importante comprender las diferencias entre la dactilología y los ideogramas, así como su uso e importancia en la LSM. La dactilología es un sistema que utiliza principalmente señas estáticas para representar las letras del alfabeto, lo que permite deletrear palabras, nombres y conceptos que no tienen una seña específica en la LSM (Serafín & Pérez, 2011). Así, la palabra mamá podría representarse con cada una de sus letras, como se muestra en la Figura 1.

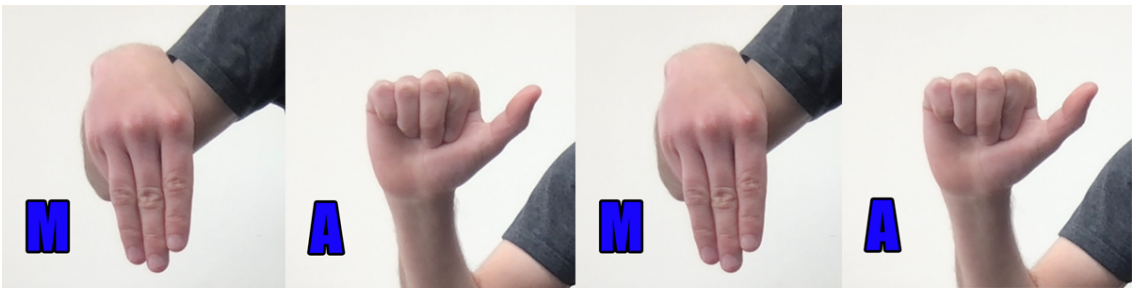


Figura 1. Deletreo manual de la palabra mamá

Por otro lado, los ideogramas son señas que representan conceptos, objetos o acciones, sin la necesidad de deletrearlas letra por letra. Estos se componen de una o varias configuraciones manuales, así como con el movimiento corporal y los rasgos no manuales. Este conjunto de señas dinámicas tienen un significado directo y se utilizan para transmitir ideas y conceptos de manera más eficiente (Serafín & Pérez, 2011). Así, el ideograma o seña particular de la palabra mamá se realiza colocando una letra m sobre la barba, tocándola con el dorso de los dedos y golpeándola ligeramente, como se observa en la Figura 2.

Tanto la dactilología como los ideogramas desempeñan un papel importante en la LSM, en conjunto, permiten una comunicación tanto precisa como eficiente. La dactilología permite deletrear palabras y nombres específicos, mientras que los ideogramas ofrecen la capacidad de transmitir conceptos y acciones de manera más directa.

La combinación de ambas formas de señas en la LSM enriquece el lenguaje y facilita la comunicación entre los usuarios de la LSM. En el marco de esta tesis, la investigación se centrará específicamente en el reconocimiento de la dactilología de la LSM.

a) Primer segmento de la seña "MAMÁ".

b) Segundo segmento de la seña "MAMÁ".

Figura 2. Ideograma para la palabra mamá.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar, implementar y evaluar un sistema que permita el reconocimiento continuo de señas estáticas de la Lengua de Señas Mexicana, utilizando visión por computadora y técnicas de aprendizaje máquina.

1.3.2. Objetivos específicos

- Generar un conjunto de datos de señas estáticas en Lengua de Señas Mexicana que incluya imágenes de manos con diversas variaciones.
- Diseñar e implementar un modelo de reconocimiento de señas estáticas utilizando técnicas de aprendizaje máquina y entrenarlo utilizando el conjunto de datos recopilado.
- Evaluar el rendimiento del sistema de reconocimiento de señas estáticas de la Lengua de Señas Mexicana en su modalidad aislada, utilizando métricas de evaluación como precisión (precision), recuerdo (recall), F1-score. Esto garantizará la capacidad del sistema para reconocer correctamente las señas con diversas variaciones.
- Evaluar la viabilidad y eficacia del sistema de reconocimiento de señas estáticas de la Lengua de

Señas Mexicana en una configuración de reconocimiento continuo.

1.4. Estructura de la tesis

La estructura del documento es la siguiente: En el Capítulo 2, se presentarán los fundamentos teóricos de las distintas técnicas de aprendizaje máquina que se utilizarán en este estudio. Se explorarán conceptos clave, así como los algoritmos relevantes para el reconocimiento de señas estáticas de lenguas de señas, sentando así las bases para el resto de los capítulos.

El Capítulo 3 se centrará en el trabajo relacionado, con el objetivo de listar y analizar los trabajos más importantes en el área de reconocimiento de señas estáticas de lenguas de señas, tanto a nivel general como específicamente en el contexto de la LSM. Se examinará el estado del arte para comprender los enfoques existentes y las contribuciones más significativas en el campo.

El Capítulo 4 se enfoca en el reconocimiento de señas estáticas en una modalidad aislada. Aquí se describe en detalle la metodología utilizada para capturar datos y el procesamiento de los mismos. De igual forma, se describen las arquitecturas, hiperparámetros y entrenamiento de los modelos de reconocimiento de señas estáticas.

En el Capítulo 5, se explora la metodología aplicada al reconocimiento de señas estáticas en la modalidad continua. Para llevar a cabo esta evaluación en el ámbito continuo, se capturan nuevos datos. Al final, se lleva a cabo una comparación entre las características de los datos empleados en el reconocimiento aislado y los datos utilizados en el contexto continuo.

En el Capítulo 6 se presentan los resultados de las modalidades aislada y continua. En la aislada, evaluamos la precisión de los modelos en el reconocimiento de señas estáticas, mientras que en la continua, además de la precisión, medimos la velocidad de reconocimiento, buscando modelos adecuados para aplicaciones en tiempo real.

Finalmente, en el Capítulo 7 se presentarán las conclusiones generales del estudio, resumiendo los hallazgos clave y discutiendo las contribuciones realizadas. Se discutirán las limitaciones encontradas durante el proceso y se plantearán las posibles direcciones para futuras investigaciones en el reconocimiento continuo de señas en LSM.

Capítulo 2. Fundamentos

En el presente capítulo, se abordarán conceptos esenciales que sustentan el desarrollo del reconocimiento continuo de señas estáticas de la Lengua de Señas Mexicana (LSM). En primer lugar, se proporcionará una introducción a conceptos clave de la LSM, como los componentes de comunicación de la LSM y el alfabeto manual.

A partir de esta base, se analizarán dos representaciones de datos: imágenes y puntos de referencia (keypoints), con el propósito de contrastar sus características distintivas. Además, se brindará una breve descripción de MediaPipe[1], una biblioteca para el procesamiento de datos y la creación de representaciones basadas en keypoints de las señas estáticas.

Posteriormente, se explorarán los cinco métodos de aprendizaje máquina seleccionados y empleados en este trabajo. Estos enfoques comprenden una Análisis Procrusteano Generalizado (GPA), Máquinas de Soporte Vectorial (SVM), Red Neuronal Profunda (DNN), Red Neuronal Convolucional (CNN) y la Memoria Asociativa Entrópica (AEM)[2].

Finalmente, se abordarán las medidas de rendimiento que se han utilizado para evaluar los modelos desarrollados, tanto en la evaluación aislada como en la evaluación continua. Permitiendo así la comparación de los diferentes modelos en ambos contextos.

2.1. La Lengua de Señas Mexicana (LSM)

Componentes de la comunicación en LSM. Liddell & Johnson (1989) introducen un sistema de transcripción que describe las señas mediante sus elementos fundamentales. El enfoque establece tres elementos principales: 1) el movimiento de la mano, 2) la forma y posición de la mano, y 3) los rasgos no manuales. En este estudio, centraremos nuestra atención exclusivamente en la forma de la mano, también denominada configuración manual.

El alfabeto manual. El alfabeto manual de la LSM se compone casi en su totalidad de señas estáticas, es decir, aquellas señas que pueden describirse únicamente a través de su configuración manual; estas

[1] `https://github.com/google/mediapipe`

[2] En este documento, vamos a usar las siglas en inglés de todos los acrónimos mencionados. Esto se debe a que estas siglas en inglés son comunes en la literatura.

señas son relevantes para nuestro estudio. En la Figura 3 se muestran imágenes para cada una de las señas que componen el alfabeto manual.

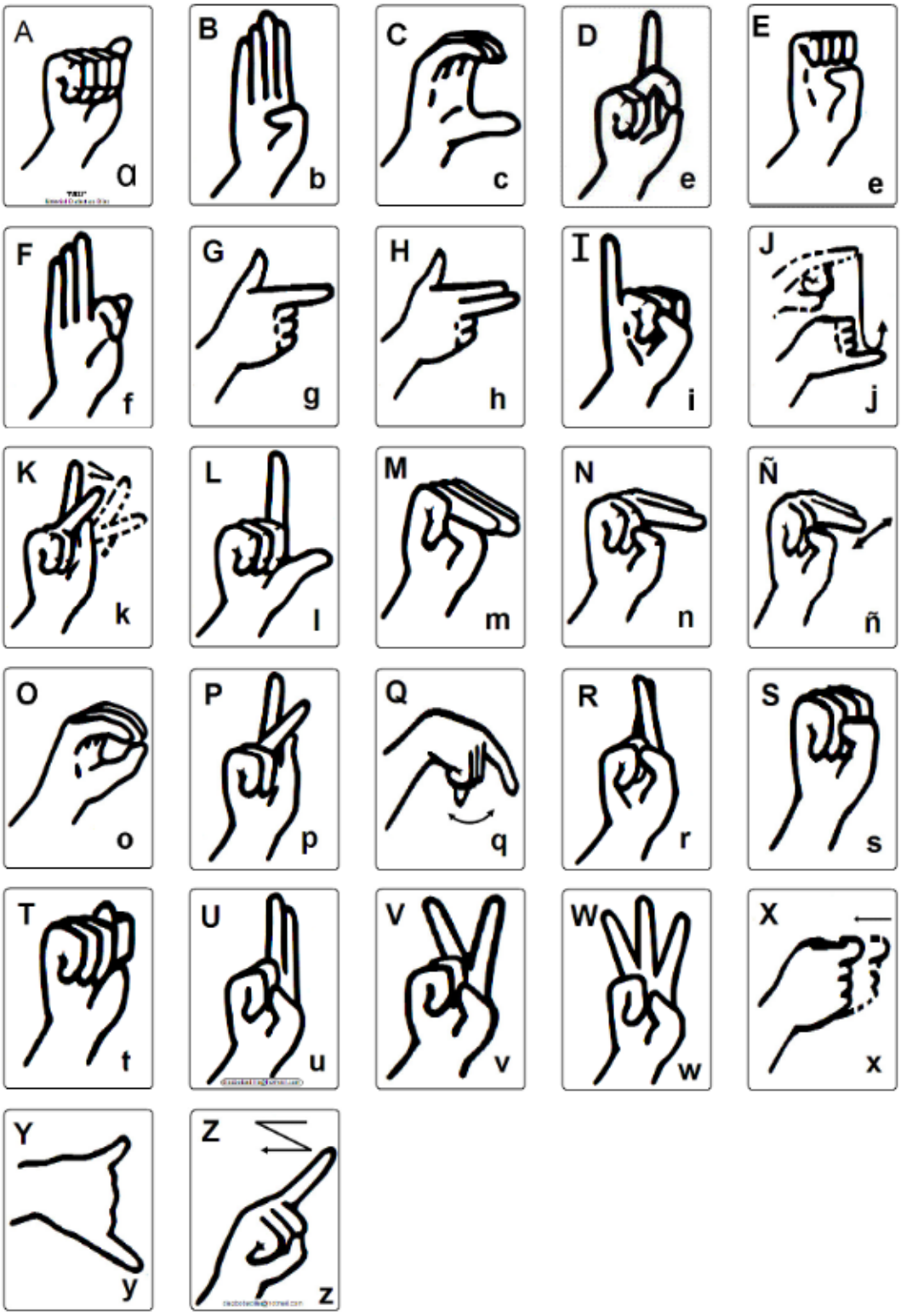


Figura 3. Alfabeto manual de la LSM. Dibujos elaborados por el artista Sordo Juan Carlos Miranda para el libro "Lenguaje Manual Aprendizaje de español signado de México" (Serafín de Fleischmann, 2014).

2.2. Representación de los datos

En esta sección se abordará un aspecto esencial del reconocimiento de señas estáticas, la representación de los datos. Se explorarán las dos representaciones utilizadas en este trabajo, la primera basada en imágenes y la segunda basada en puntos de referencia (keypoints). En ambos enfoques, se busca conservar características relevantes que permitan reconocer la forma de la mano de manera precisa.

La Figura 4a presenta un ejemplo extraído del dataset utilizado en este trabajo, antes del procesamiento. Después del procesamiento se generan dos representaciones: en la Figura 4b se encuentra la representación basada en imágenes, mientras que en la Figura 4c se muestra la representación basada en keypoints.

a) Ejemplo extraído del dataset utilizado en este trabajo, antes del procesamiento.

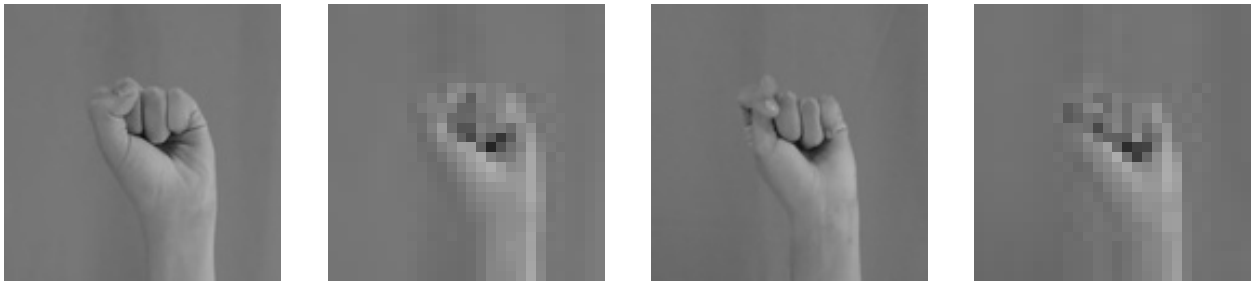
b) Ejemplo representado como una imagen, escalada y en escala de grises.

c) Ejemplo representado como un conjunto de keypoints.

Figura 4. Ejemplos de representaciones de datos utilizados en este trabajo.

2.2.1. Imágenes

Las imágenes empleadas en esta representación presentan una resolución de 128x128 píxeles. Esta elección se efectuó con el fin de equilibrar la calidad y el costo de procesamiento, considerando que una mayor resolución podría requerir más recursos sin necesariamente aportar mejoras significativas en la captura de las características esenciales de la forma de la mano. A pesar de la existencia de conjuntos de datos conocidos, como MNIST (Deng, 2012), que emplean imágenes de 28x28 pixeles para reconocer dígitos manuscritos, este tipo de imágenes no logra capturar de manera adecuada las sutilezas presentes en las letras del alfabeto manual (Figura 5).



a) Letra "S" 128x128 pixeles. b) Letra "S" 28x28 pixeles. c) Letra "T" 128x128 pixeles. d) Letra "T" 28x28 pixeles.

Figura 5. Representación de señas en LSM en diferentes resoluciones. En las subfiguras a) y c), se observan las letras "S" y "T", respectivamente, a una resolución de 128x128 píxeles. En estas imágenes, la diferencia entre ambas señas es evidente, destacando la posición del pulgar de manera clara. En contraste, las subfiguras b) y d) muestran las mismas señas a una resolución de 28x28 píxeles, donde apreciar los detalles de los dedos se torna difícil.

Es importante destacar que las imágenes utilizadas en esta representación están en escala de grises. Esta elección se realizó con el objetivo de reducir el número de parámetros a aprender en todos los métodos que involucren redes neuronales. Además, se disminuye la complejidad de los modelos, lo que puede resultar en un entrenamiento más eficiente y una menor demanda computacional. A pesar de esta reducción en la información de color, es posible mantener las características visuales relevantes que permiten determinar la forma y orientación de la mano, siendo suficiente la información de tonos y sombras en escala de grises.

2.2.2. Keypoints

Los keypoints, también conocidos como puntos de referencia, son puntos específicos en la imagen que resaltan elementos significativos de la mano, como la posición de los dedos y las articulaciones. Estos keypoints se definen mediante coordenadas (x, y, z)[3] que indican su ubicación en la imagen.

La elección de utilizar keypoints como representación de los datos ofrece múltiples ventajas. En primer lugar, permite una descripción limpia y precisa de la mano, al resaltar sus puntos cruciales. Además, la información extraída de los keypoints suele ser más compacta en comparación con la representación de la imagen completa, lo que potencialmente reduce la carga computacional durante el entrenamiento de los modelos. Adicionalmente, esta representación necesita menos espacio de almacenamiento en comparación con las imágenes completas.

[3] Ciertas bibliotecas de extracción de keypoints necesitan una cámara RGB-D para capturar la componente Z, mientras que otras pueden inferirla directamente a partir de una imagen 2D.

2.2.2.1. MediaPipe

MediaPipe es una librería de código abierto desarrollada por Google que proporciona un conjunto de herramientas y modelos preentrenados para el procesamiento de datos multimedia, como imágenes y videos. Su objetivo principal es facilitar la integración rápida de técnicas de inteligencia artificial (IA) y aprendizaje automático (ML) en diversas aplicaciones.

Esta librería es especialmente reconocida por su capacidad para detectar y seguir múltiples puntos claves en objetos en movimiento, como manos, caras y cuerpos humanos. En particular, nos enfocamos en el módulo de detección de manos, ya que es de particular interés para nuestro proyecto de reconocimiento continuo de señas estáticas.

El módulo de detección de manos de MediaPipe[4] permite inferir las coordenadas (x, y, z) de 21 puntos de referencia de la mano (keypoints), listados en la Figura 6. De acuerdo con los desarrolladores, este modelo fue entrado utilizando alrededor de 30,000 imágenes del mundo real, así como varias imágenes renderizadas de manos sobre distintos fondos.

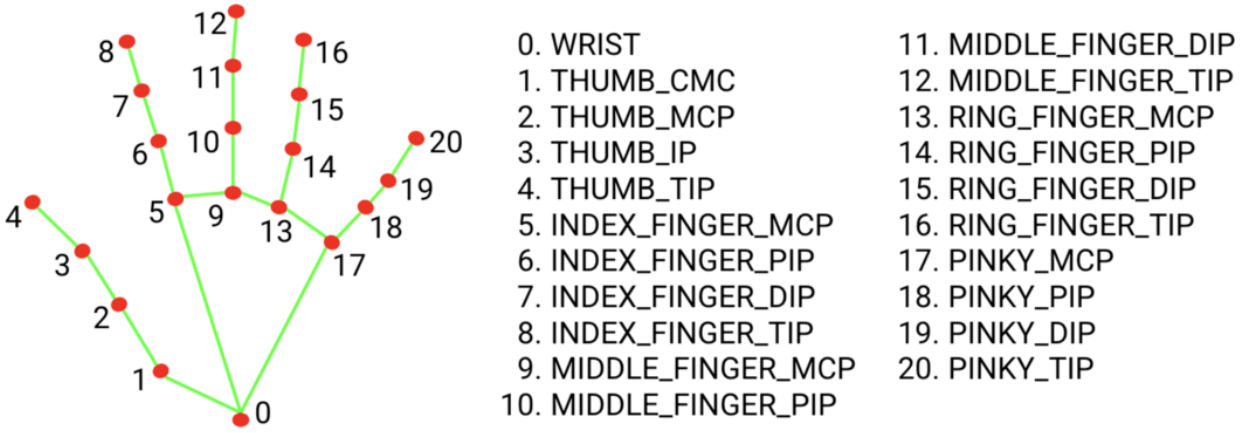


Figura 6. Arreglo de keypoints de la mano que MediaPipe es capaz de extraer a partir de imágenes.[5]

Una de las ventajas distintivas al elegir MediaPipe en comparación con otras librerías es su capacidad para realizar inferencias utilizando únicamente la unidad central de procesamiento (CPU) con un buen rendimiento. Esto marca una diferencia notable respecto a otras herramientas que requieren el uso de una unidad de procesamiento gráfico (GPU) para lograr una ejecución en tiempo real.

[4] `https://developers.google.com/mediapipe/solutions/vision/hand_landmarker`

[5] Las líneas que enlazan los keypoints son exclusivamente elementos visuales de la figura. MediaPipe no ofrece detalles sobre la conectividad entre los keypoints.

2.3. Análisis Procrusteano

El Análisis Procrusteano (Gower, 1975) es una técnica que se utiliza para comparar o alinear dos conjuntos de datos, generalmente matrices, en un espacio común. El objetivo principal es ajustar uno de los conjuntos de datos al otro aplicando transformaciones geométricas, como rotaciones, escalados y traslaciones. De manera que los dos conjuntos de datos sean lo más similares posible en términos de su forma y estructura. En la Figura 7 se muestra una visión general de las transformaciones que se realizan en el análisis Procrusteano simple, a lo largo de esta sección se abordarán cada una de estas transformaciones.

Supongamos que tenemos dos conjuntos de n puntos bidimensionales

$$A = \{(x_{1a}, y_{1a}), (x_{2a}, y_{2a}), \ldots, (x_{na}, y_{na})\} \quad \text{y} \quad B = \{(x_{1b}, y_{1b}), (x_{2b}, y_{2b}), \ldots, (x_{nb}, y_{nb})\}$$

.

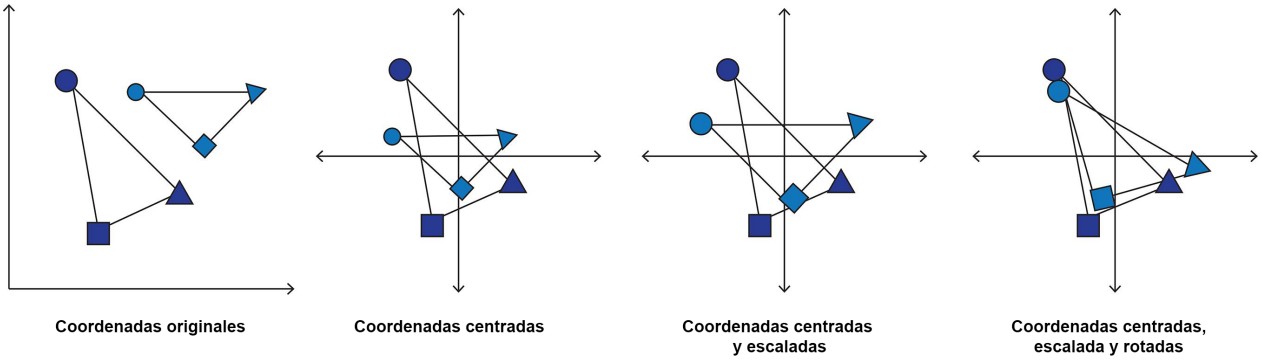


Figura 7. Transformaciones realizadas en el análisis Procrusteano. Figura adaptada de (del Medico et al., 2020)

El proceso del análisis Procrusteano involucra los siguientes pasos:

Traslación. Restamos las coordenadas x y y de cada conjunto de datos por su respectiva media para centrar los puntos en el origen:

$$X_{\text{centrado}} = X - \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{y} \quad Y_{\text{centrado}} = Y - \frac{1}{n}\sum_{i=1}^{n} y_i$$

donde X y Y son matrices que contienen las coordenadas x e y de los puntos en los conjuntos A y B.

Escalamiento uniforme. Después de centrar los datos, escalamos los puntos para que las normas de los vectores sean iguales a 1:

$$X_{\text{escala}} = \frac{X_{\text{centrado}}}{\|X_{\text{centrado}}\|} \quad \text{y} \quad Y_{\text{escala}} = \frac{Y_{\text{centrado}}}{\|Y_{\text{centrado}}\|}$$

Rotación. Mediante técnicas de álgebra lineal como la descomposición en valores singulares (SVD), se calcula una matriz de rotación R que minimiza la discrepancia entre los conjuntos X_{escala} y Y_{escala}:

$$R = V \cdot U^T$$

donde U y V son las matrices obtenidas de la SVD de $X_{\text{escala}}^T \cdot Y_{\text{escala}}$.

Una vez encontrada la mejor matriz de rotación, se aplica esta transformación al conjunto de datos que se está ajustando.

$$X_{\text{rotado}} = X_{\text{escala}} \cdot R \quad \text{y} \quad Y_{\text{rotado}} = Y_{\text{escala}} \cdot R$$

Cálculo de la discrepancia. Finalmente, calculamos alguna métrica de discrepancia, como la suma de los cuadrados de las diferencias entre los puntos de X_{rotado} y Y_{rotado}, para evaluar la similitud entre los conjuntos de datos transformados:

$$\text{Suma de los Cuadrados} = \sum_{i=1}^{n} \|x_i - y_i\|^2$$

2.4. Máquinas de Soporte Vectorial (SVM)

Las Máquinas de Soporte Vectorial (Cortes & Vapnik, 1995) son algoritmos de aprendizaje máquina utilizados para clasificación y regresión. Las SVM son particularmente útiles para problemas en los que los datos son linealmente separables en un espacio multidimensional. Además, cuentan con la habilidad

de transformar los datos a un espacio de mayor dimensión, donde podrían volverse linealmente separables, incluso si en el espacio original no lo son.

La idea fundamental detrás de las SVM es encontrar un hiperplano (una especie de "línea" en dimensiones superiores) que maximice el margen entre diferentes clases de datos. El margen se define como la distancia entre el hiperplano y los puntos más cercanos de cada clase, estos puntos son llamados vectores de soporte. La intuición detrás de esto es que un hiperplano con un mayor margen generaliza mejor y es menos propenso al sobreajuste en nuevos datos.

En la Figura 8 se presenta un ejemplo de cómo las Máquinas de Soporte Vectorial se usan para resolver problemas de clasificación. En esta ilustración, las clases están representadas por los símbolos (+) y (-). El hiperplano visible separa estos dos conjuntos maximizando el margen entre ellos.

En su forma más básica, las SVM se aplican principalmente a problemas de clasificación binaria, es decir, cuando se trata de dividir datos en dos clases diferentes. Sin embargo, cuando se enfrenta a problemas multiclase, donde hay más de dos clases a clasificar, las SVM pueden ser utilizadas mediante dos enfoques principales: uno contra uno (OvO, One-vs-One) y uno contra todos (OvA, One-vs-All).

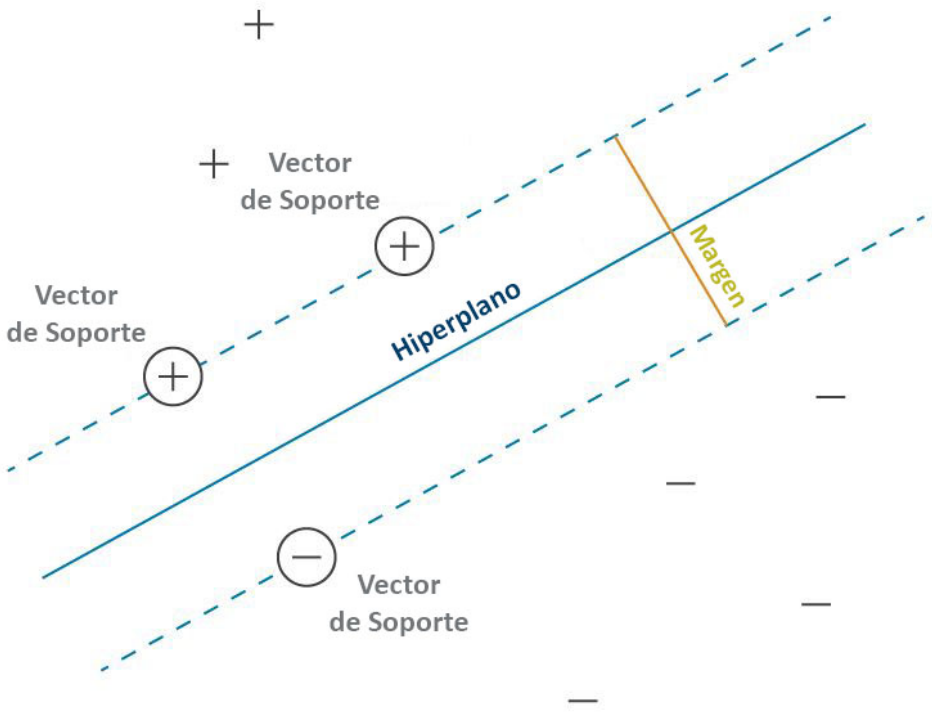


Figura 8. Ejemplo de las SVM para resolver un problema de clasificación binaria[6].

[6]Figura adaptada de `https://www.mathworks.com/discovery/support-vector-machine.html`.

En el enfoque OvO, se construye un clasificador binario para cada par único de clases. Por ejemplo, si tienes tres clases (A, B y C), crearías tres clasificadores: uno para A vs B, otro para A vs C y otro para B vs C. Luego, cuando quieras clasificar un nuevo dato, cada clasificador vota por su clase correspondiente, y la clase con más votos se elige como la predicción final.

En el enfoque OvA, se construye un clasificador para cada clase individualmente, considerando esa clase como la clase positiva y el resto de las clases como la clase negativa. Similar al enfoque OvO, cuando se realiza la predicción, cada clasificador vota y la clase con más votos se selecciona como la predicción final.

Después de comprender el funcionamiento básico de las SVM, es importante explorar una técnica clave conocida como el "kernel trick". Esta técnica permite lidiar con datos que no son linealmente separables en su forma original. En lugar de modificar directamente los datos, los kernels calculan las similitudes entre pares de datos en un espacio de mayor dimensión.

En la Figura 9, se muestra un problema de clasificación, las clases están representadas por los colores rojo y verde. En la imagen de la izquierda se observa que los puntos no son linealmente separables en el espacio bidimensional. Sin embargo, al aplicar un kernel con función de base radial, se muestra cómo es posible encontrar un hiperplano que separa las clases en un espacio tridimensional.

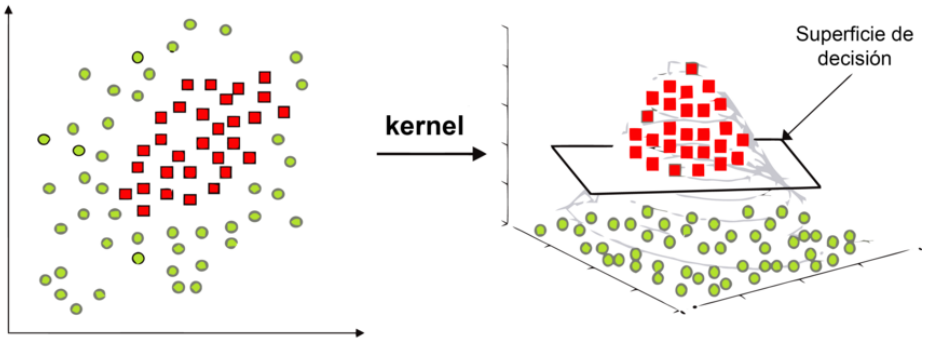


Figura 9. Impacto del "kernel trick" en un problema de clasificación no linealmente separable en su forma original[7].

Algunos de los kernels más utilizados en las SVM con sus correspondientes fórmulas matemáticas son:

- **Kernel Lineal:**

  $K(x, y) = x^T y$

[7]Figura adaptada de https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d.

- **Kernel Polinómico:**
  $K(x, y) = (x^T y + c)^d$
  donde c es un término constante y d es el grado del polinomio.

- **Kernel Función de Base Radial (RBF):**
  $K(x, y) = \exp\big(-\gamma||x - y||^2\big)$ donde γ (gamma) es un parámetro de ajuste que determina la influencia de cada punto de datos en el cálculo de similitud.

- **Kernel Sigmoide:**
  $K(x, y) = \tanh\big(\alpha x^T y + c\big)$
  donde α y c son constantes ajustables.

Estas fórmulas representan cómo se calcula la similitud entre dos puntos "x" e "y" en el espacio original o en un espacio de mayor dimensión. Al elegir un kernel, se ajustan los parámetros relevantes (como "c", "d" o "γ") para adaptarlo al comportamiento de los datos y al problema que se está abordando.

2.5. Redes Neuronales

Las redes neuronales han emergido como una poderosa herramienta en el campo de la inteligencia artificial. Estas estructuras computacionales, inspiradas en las interconexiones de neuronas biológicas, han revolucionado la forma en que abordamos tareas de reconocimiento de patrones. A lo largo de esta sección se examinarán los conceptos fundamentales de un solo perceptrón hasta las arquitecturas más sofisticadas, como las redes neuronales convoluciones y los autoencoders, y cómo su entrenamiento y ajuste de parámetros son cruciales para alcanzar un rendimiento óptimo.

2.5.1. El Perceptrón

Para comprender plenamente el funcionamiento de las redes neuronales, es esencial explorar el concepto fundamental del perceptrón (Rosenblatt, 1958). El perceptrón, es un tipo de modelo de neurona artificial diseñada para emular la forma en que las neuronas biológicas procesan la información en el cerebro.

En términos simples, un perceptrón es una unidad computacional que toma una serie de entradas, realiza

cálculos en función de esos datos y genera una salida. En la Figura 10 se muestra un esquema general del perceptrón (Jurafsky & Martin, 2008). Posteriormente, se realizará una descripción detallada de cada uno de sus componentes.

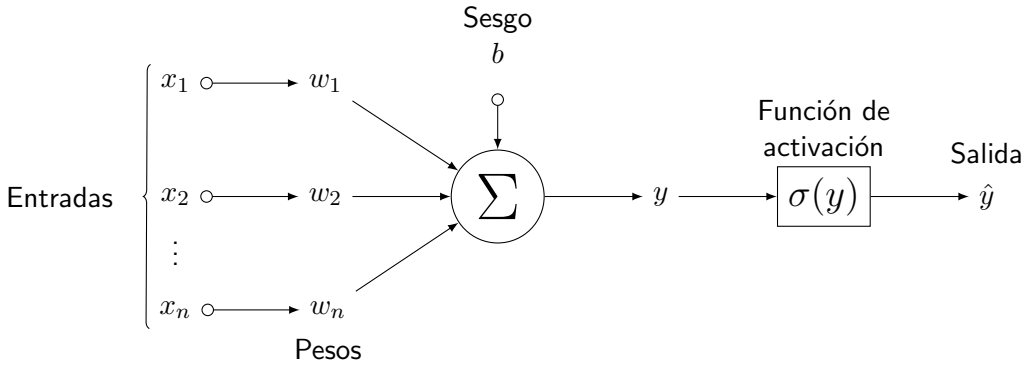


Figura 10. Ilustración de un perceptrón, una unidad básica en las redes neuronales[8].

Dado un conjunto de características de entrada $x_0, x_1, \ldots, x_n$, pesos $w_0, w_1, \ldots, w_n$ y un término de sesgo b, el perceptrón lleva a cabo una operación que implica sumar los productos de las características de entrada (x) con sus pesos correspondientes (w), y luego agrega el término de sesgo (b). Esta suma ponderada resultante produce una salida denominada y.

$$y = \sum_{i=1}^{n} (w_i \cdot x_i) + b$$

La salida (y) es luego sometida a una función de activación, representada como $(\sigma(y))$. Esta función de activación toma el valor de salida obtenido en el paso previo (y) como entrada y ejecuta algún tipo de normalización, generando una nueva salida $(\hat{y})$. Siendo este $(\hat{y})$ el valor final de salida del perceptrón.

La función escalón (Figura 11) es una de las funciones más simples utilizadas en los perceptrones y otros modelos de redes neuronales. Si la entrada excede un cierto umbral (comúnmente este umbral es 0), la salida es 1; de lo contrario, la salida es 0.

No obstante, es importante destacar que el perceptrón original, con su función de activación escalón, tiene limitaciones notables. Este tipo de función de activación solo puede resolver problemas que son linealmente separables, es decir, problemas en los que se puede trazar una línea recta (o hiperplano) para dividir claramente las clases de datos. Los problemas más complejos, como aquellos que involucran

[8] Figura adaptada de `https://tex.stackexchange.com/questions/505741/architecture-neural-network-with-weights`.

relaciones no lineales entre los datos, no pueden ser resueltos por un solo perceptrón de activación escalón.

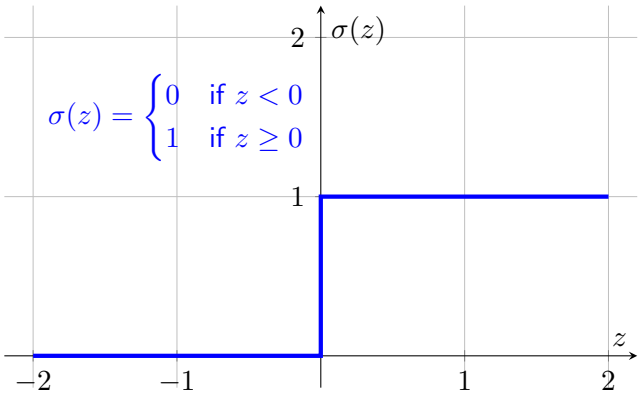


Figura 11. Función escalón.

Para superar estas limitaciones se pueden utilizar múltiples perceptrones (o neuronas artificiales) interconectados en capas para formar una red neuronal artificial, lo que da lugar a arquitecturas más avanzadas como las redes neuronales profundas. Además, se suelen incorporar diversas funciones de activación no lineales, las cuales juegan un papel importante en la capacidad de aprendizaje de las redes neuronales.

2.5.2. Del perceptrón a las redes neuronales

Una red neuronal se organiza en capas, cada una compuesta por varias neuronas. Aunque un solo perceptrón puede tomar decisiones simples, las redes neuronales con múltiples capas y conexiones, como se muestra en la Figura 12, permiten abordar tareas más complejas.

Estas capas procesan y transforman gradualmente la información, lo que les permite aprender y representar relaciones complejas en los datos de manera eficiente. Las capas se pueden dividir en tres tipos principales:

- **Capa de entrada:** Es la primera capa de la red y recibe los datos de entrada, que pueden ser características de un problema específico, como pixeles de una imagen o palabras de un texto.
- **Capas ocultas:** Estas capas intermedias procesan la información a medida que pasa a través de

[8] Figura adaptada de `https://tex.stackexchange.com/questions/153957/drawing-neural-network-with-tikz`.

la red. Cada capa oculta está compuesta por múltiples neuronas que realizan transformaciones en los datos de entrada.

- **Capa de salida:** Esta capa produce los resultados finales de la red neuronal. Dependiendo de la tarea, la capa de salida puede tener una o varias neuronas que generan las predicciones o clasificaciones deseadas.

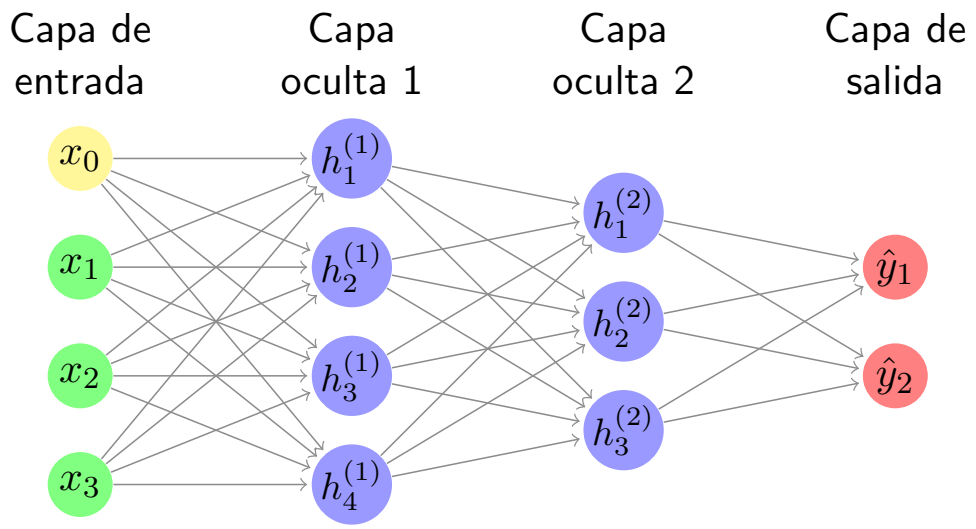


Figura 12. Arquitectura de red neuronal profunda con múltiples capas[9].

Conexiones. Las conexiones en una red neuronal representan las relaciones entre las neuronas de diferentes capas. Cada conexión tiene un peso asociado que determina la influencia que tiene la salida de una neurona en la entrada de otra. Durante el entrenamiento de la red, estos pesos se ajustan para que la red aprenda a generar las salidas deseadas a partir de las entradas.

Feedforward. Cuando los datos ingresan a la red neuronal a través de la capa de entrada, las neuronas de esta capa se activan en función de los valores de entrada y los pesos de las conexiones. Estas activaciones se propagan a través de las capas ocultas hasta llegar a la capa de salida, donde se generan las predicciones o clasificaciones. A este proceso se le llama *feedforward*.

Retropropagación. Durante el entrenamiento, se compara la salida predicha con la salida real y se calcula un error con ayuda de una *función de pérdida*. Luego, el algoritmo de *retropropagación* ajusta los pesos de las conexiones en todas las capas para minimizar este error, lo que permite a la red aprender de los datos y mejorar su rendimiento con el tiempo.

2.5.3. Arquitecturas

Las arquitecturas en las redes neuronales se refieren a la estructura y organización de una red compuesta por neuronas artificiales interconectadas. Estas arquitecturas están diseñadas para extraer características y encontrar patrones útiles de los datos de entrada. Las arquitecturas puede variar en términos de su topología, esto es, en la cantidad de capas, el tipo de conexiones entre las neuronas y las funciones de activación utilizadas. Cada una de estas arquitecturas se han desarrollado para abordar una amplia gama de problemas, desde el reconocimiento de imágenes hasta la traducción automática. A continuación se realizará una breve introducción a las arquitecturas utilizadas en este trabajo.

2.5.3.1. Redes Neuronales Convolucionales (CNNs)

Las Redes Neuronales Convolucionales son una clase de redes neuronales profundas diseñadas específicamente para el procesamiento eficiente de datos con estructura de cuadrícula, como imágenes y datos espaciales. Su arquitectura se inspira en cómo el sistema visual biológico procesa información. A diferencia de las redes neuronales tradicionales, las CNN se centran en preservar la relación espacial entre los datos.

En la Figura 13 se muestra un esquema general de la arquitectura básica de una CNN, a lo largo de esta sección se abordarán cada uno de los conceptos fundamentales de las CNN.

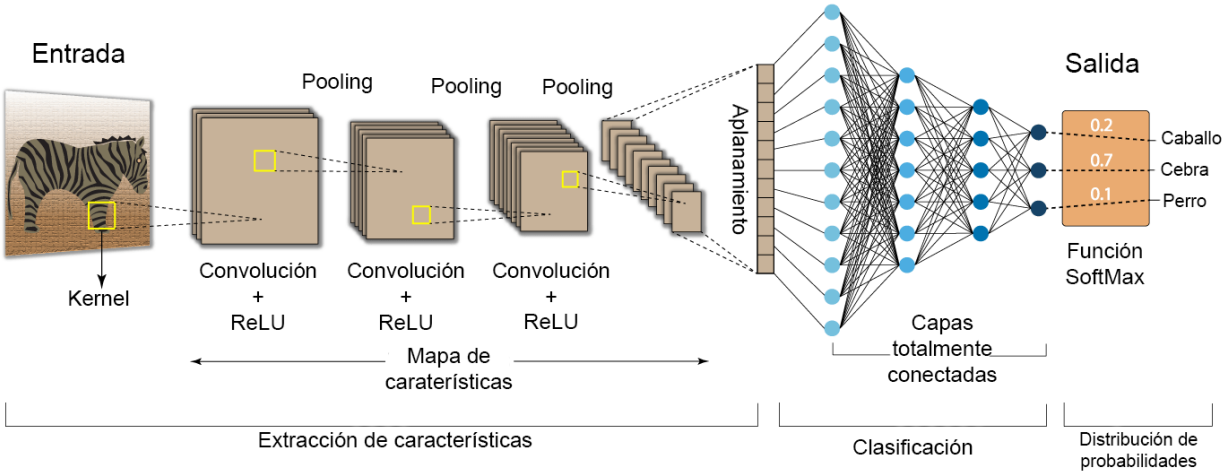


Figura 13. Arquitectura básica de una CNN[10].

Convoluciones. Uno de los conceptos más importantes de las CNN son las convoluciones. Estas consisten en aplicar un filtro (kernel) a una pequeña región de la imagen de entrada y calcular la suma ponderada de los valores de pixeles en esa región. Estos filtros permite detectar características específicas, como bordes, equinas y texturas. A medida que se desplaza el filtro por la imagen, se obtienen mapas de características que resaltan áreas de interés.

En la Figura 14 se muestra un ejemplo de la convolución entre una imagen de dimensiones 4x4 con un filtro de tamaño 2x2, las dimensiones y los valores del kernel pueden variar de acuerdo con las características específicas que se deseen capturar.

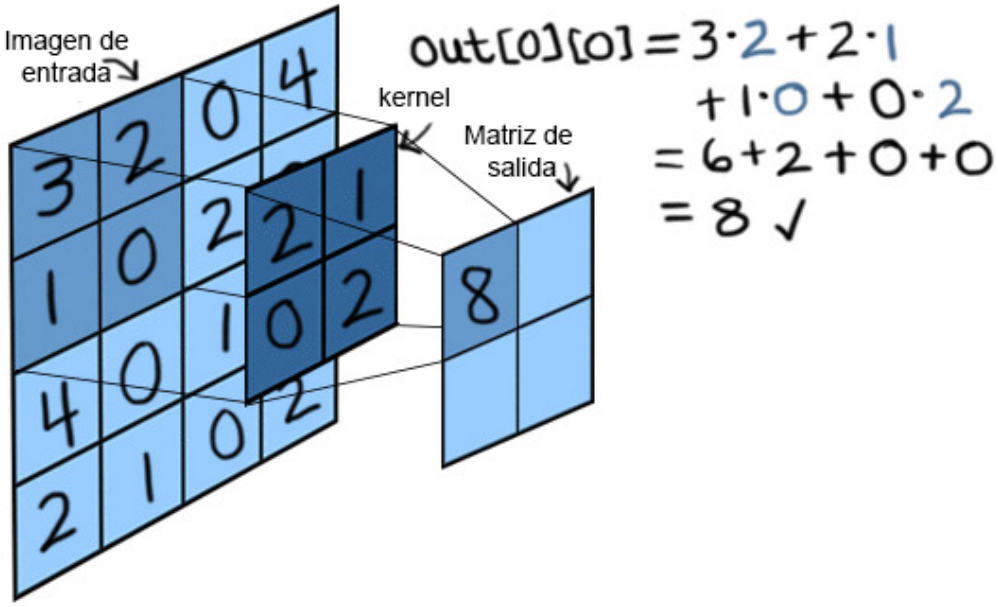


Figura 14. Ejemplo de convolución entre una imagen y un filtro (kernel)[11].

Capas convolucionales. Las capas convolucionales en una CNN están compuestas por múltiples filtros convolucionales. Los mapas de características generados por cada filtro se apilan para formar un volumen de salida, que luego se convierten en la entrada para la siguiente capa.

Función de activación. Después de aplicar la operación de convolución, se aplica una función de activación no lineal, como la función Rectified Linear Unit (ReLU) (Figura 15), que introduce no linealidad a la red y permite capturar relaciones complejas a los datos.

[10] Figura adaptada de `https://shorturl.at/cflE7`.

[11] Figura adaptada de `https://blog.insightdatascience.com/convolutional-neural-networks-explained-with-american-ninja-warrior-c6649875861c`.

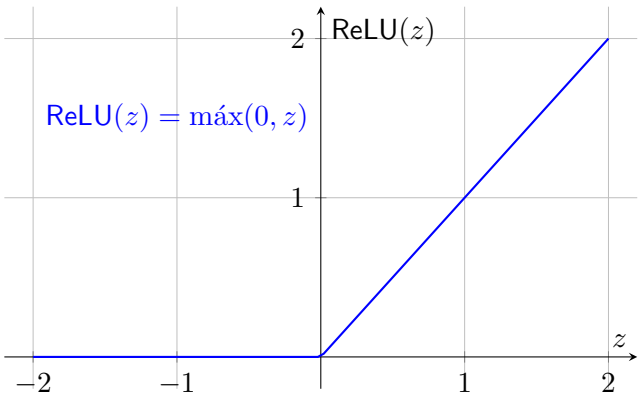


Figura 15. Función ReLU.

Capas de pooling. Las capas de pooling reducen el tamaño espacial de las representaciones y disminuyen la cantidad de parámetros de la red. Una capa de pooling comúnmente utilizada es la MaxPooling, esta capa selecciona el valor máximo de un área determinada en el para de características, lo que ayuda a conservar las características más importantes.

Capa de aplanamiento. La capa de aplanamiento (flatten) es una capa utilizada para transformar los datos bidimensionales o multidimensionales en un formato unidimensional. Esta capa es necesaria cuando se pasa de las capas convoluciones y de pooling a las capas totalmente conectadas.

Capas totalmente conectadas. Las capas totalmente conectadas realizan la clasificación. Estas capas toman las características generadas por las capas anteriores y ayudan a tomar decisiones finales sobre la clase objetivo.

Función Softmax. Finalmente, se calcula una distribución de probabilidades con ayuda de una función Softmax y con ello determina a qué clase pertenece la imagen de entrada de la red.

2.5.3.2. Autoencoders

Los Autoencoders son una clase de modelos de redes neuronales que se utilizan para aprender representaciones eficientes de datos de entrada en forma de vectores de menor dimensión. Un autoencoder consta de dos partes principales: el codificador (*encoder*) y el decodificador (*decoder*) (Figura 16). El

codificador toma la entrada original y la convierte en una *representación latente* de menor dimensión. Luego, el decodificador toma esta representación latente y trata de reconstruir la entrada original a partir de ella. El objetivo es que la reconstrucción sea lo más cercana posible a la entrada original.

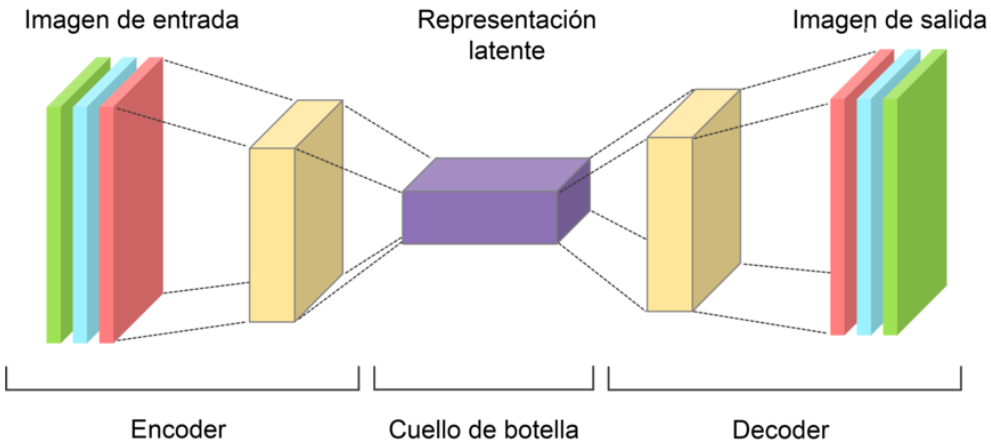


Figura 16. Arquitectura básica de un autoencoder[12].

La representación latente aprendida por el autoencoder puede ser utilizada como características para entrenar un clasificador. Este enfoque aprovecha las capacidades del autoencoder para aprender características significativas y reducir la dimensionalidad de los datos.

No obstante, estas características son útiles y efectivas si las entradas pertenecen al mismo dominio que los datos de entrenamiento, ya que están específicamente adaptadas a ese contexto. Aplicar estas características a un dominio diferente puede no resultar útil. Por lo tanto, es esencial considerar la coherencia del dominio al usar las representaciones latentes para tareas adicionales, como la clasificación.

2.6. Memoria Asociativa Entrópica (AEM)

La Memoria Asociativa Entrópica (Pineda & Morales, 2023) es un modelo computacional que se basa en un enfoque de memoria distribuida y asociativa, diseñado para simular ciertas propiedades de la memoria humana. Este modelo utiliza tres niveles de representación: el primero corresponde a imágenes concretas de entrada y salida, el segundo a representaciones abstractas amodales y el tercero a la representación distribuida almacenada en la memoria.

[12]Figura adaptada de `https://github.com/ghnmqdtg/Deep-Learning-Based-Noise-Reduction-and-Speech-Enhancement-System`.

En la Figura 17 se muestra una ilustración con los tres niveles de representación que se utilizan en la AEM. Para transformar los datos de entrada (imágenes) a una representación amodal abstracta (funciones) se utilizó un codificador (la primera parte de un autoencoder) y un proceso de cuantificación.

Del mismo modo, es factible reconstruir una imagen a partir de esa representación abstracta utilizando la operación inversa de decuantificación, facilitada por un decodificador (la segunda parte de un autoencoder). Sin embargo, en este trabajo no se abordará la recuperación de imágenes.

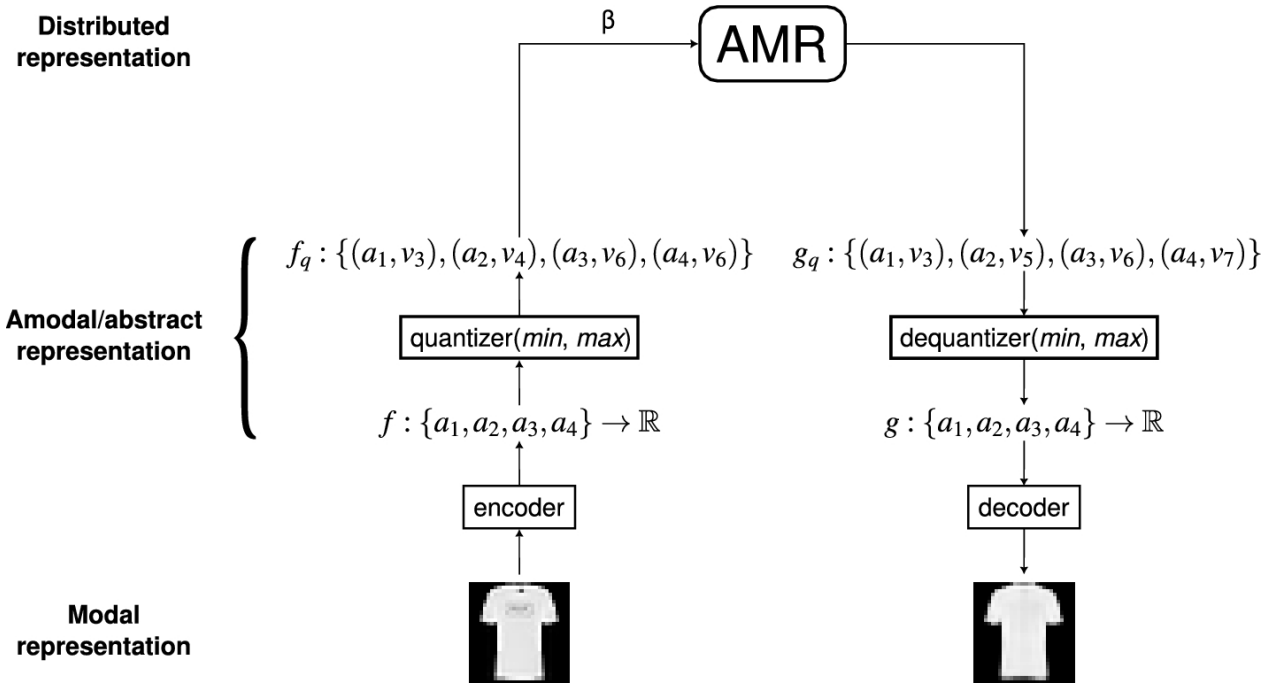


Figura 17. Niveles de representación en la AEM[13].

En el contexto de la AEM, las operaciones de memoria se definen mediante operaciones formales, como el registro de memoria (λ), el reconocimiento de memoria (η) y la recuperación de memoria (β). El proceso de recuperación implica la selección de objetos basados en una entrada de estímulo (cue) y la relación de los objetos almacenados en la memoria.

En la Figura 18 se muestra una representación gráfica de la operación de registro λ. El estímulo (cue) se simboliza como una función $\{(a1, v3), (a2, v1), (a3, v6), (a4, v7)\}$ y se presenta en forma de tabla en el registro auxiliar (Aux-Reg). Después se ingresa en un Registro de Memoria Asociativa (AMR) que ya contiene información en el estado t. El valor de cada celda en el registro auxiliar se suma al valor de su celda correspondiente en el AMR en el estado t, lo que genera el estado $t + 1$.

[13]Todas las figuras presentadas en esta sección fueron extraídas del artículo (Pineda & Morales, 2023).

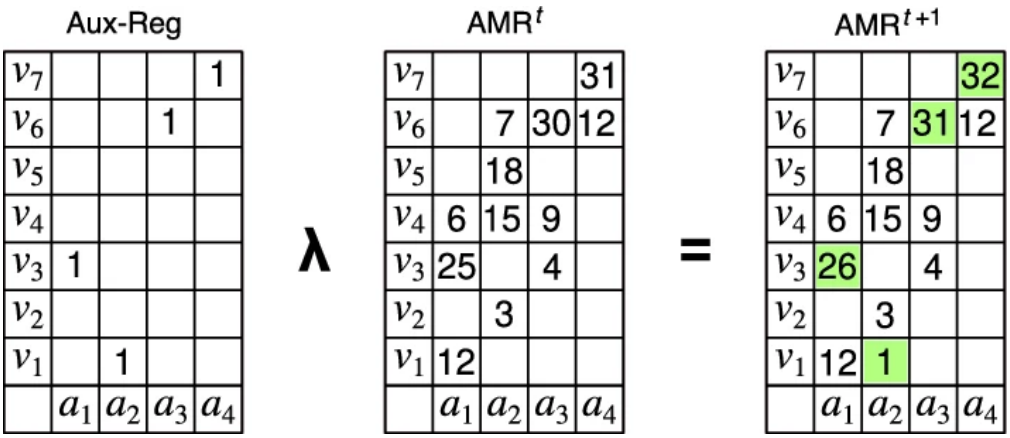


Figura 18. Operación de registro λ.

En la Figura 19 se muestra una ilustración de la operación de reconocimiento η. En la parte superior del diagrama, un estímulo (cue) se considera aceptada solamente si todas las celdas correspondientes en el Registro de Memoria Asociativa (AMR), relacionadas con las celdas utilizadas por la pista en el Registro Auxiliar, tienen valores diferentes a cero.

En contraste, en la parte inferior, una pista se considera rechazada cuando una o más celdas en el Registro de Memoria Asociativa (AMR), relacionadas con las celdas utilizadas por la pista en el Registro Auxiliar, tienen valores iguales a cero.

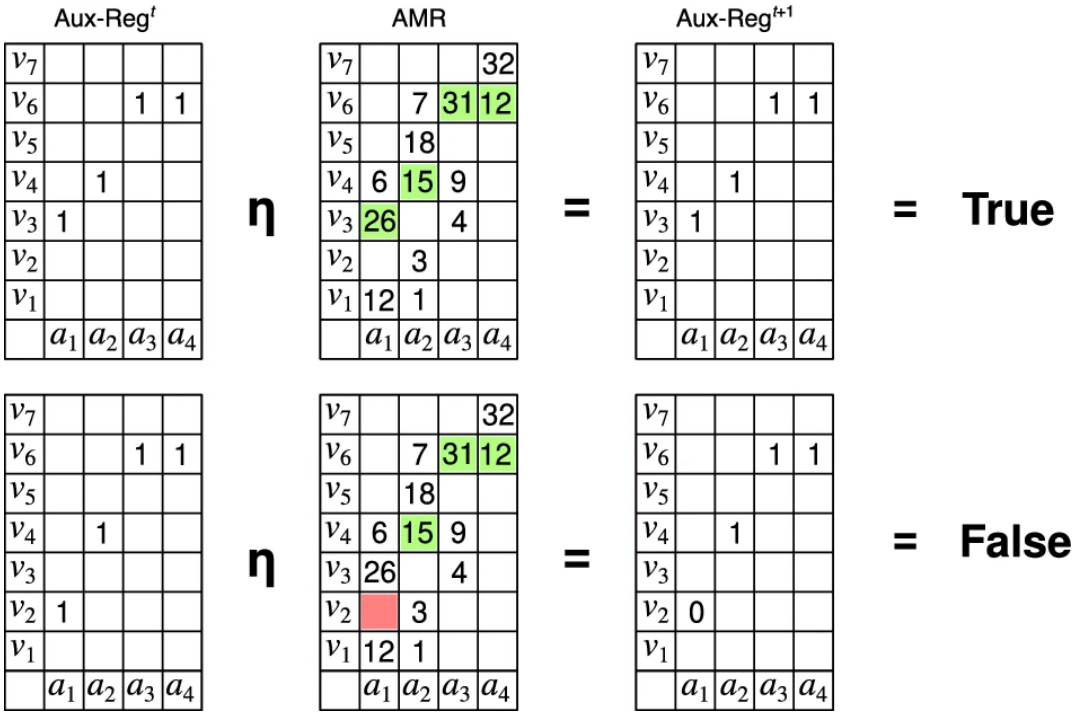


Figura 19. Operación de reconocimiento η.

Finalmente, en la Figura 20 se muestra la operación de recuperación β. Cada columna del registro auxiliar se trata como una distribución de probabilidad normal que tiene su centro en la celda utilizada por el estímulo (cue). La desviación estándar σ, que es un parámetro ajustable, define la forma de esta distribución. Similarmente, las columnas en el Registro de Memoria Asociativa (AMR) también son vistas como distribuciones de probabilidad. Para obtener el valor de cada columna en el AMR, se realiza un proceso de selección aleatoria multiplicando la distribución que representa el estímulo (cue) con la distribución específica de la columna en el AMR.

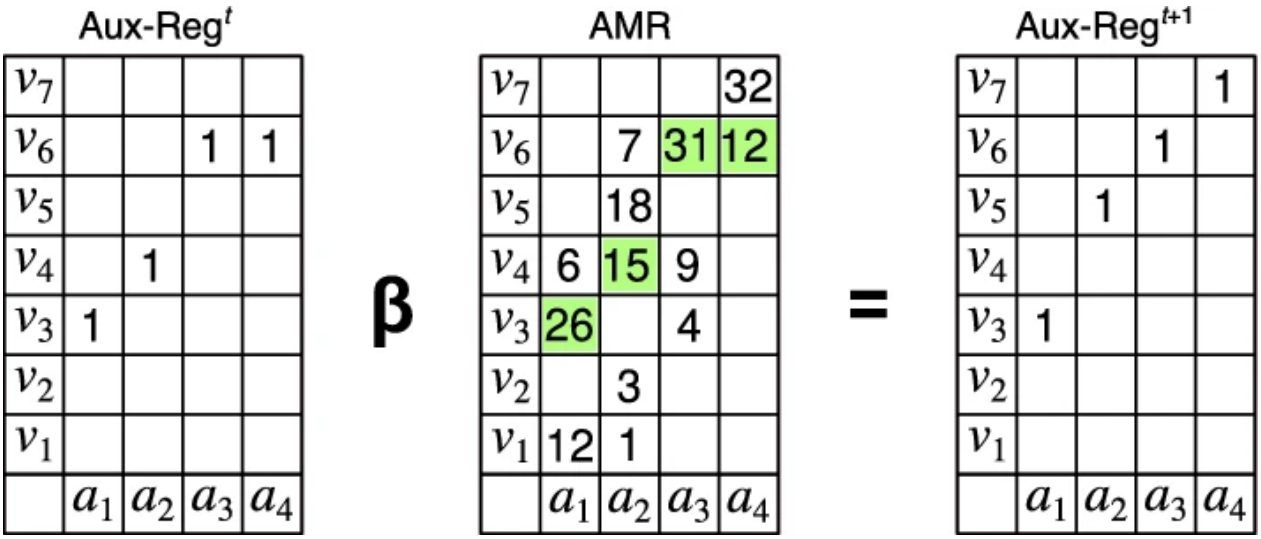


Figura 20. Operación de recuperación β.

Una característica clave de la AEM es su capacidad para manejar la indeterminación y entropía. La indeterminación se refiere a la superposición de unidades de contenido almacenadas, lo que permite la recuperación de objetos relacionados y la construcción de nuevos objetos. La entropía se utiliza para medir la indeterminación y la ambigüedad en la recuperación de objetos de la memoria.

2.7. Métricas de evaluación

En este capítulo se explorarán las métricas de evaluación utilizadas para medir el rendimiento de los modelos de reconocimiento de señas en LSM, tanto a nivel aislado como a nivel continuo. Estas métricas brindan una comprensión cuantitativa del desempeño y la generalización de los modelos. Se presentan tanto las métricas utilizadas en este trabajo como las reportadas en el Capítulo 3 "Trabajo relacionado". A continuación, se presentarán las métricas junto con sus fórmulas correspondientes.

2.7.1. Métricas para la evaluación aislada

Exactitud (Accuracy). La exactitud evalúa la proporción de predicciones correctas en relación con el número total de predicciones. Se define de la siguiente manera:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

donde:

- VP (Verdaderos Positivos) son los casos correctamente clasificados como positivos.
- VN (Verdaderos Negativos) son los casos correctamente clasificados como negativos.
- FP (Falsos Positivos) son los casos clasificados incorrectamente como positivos.
- FN (Falsos Negativos) son los casos clasificados incorrectamente como negativos.

Precisión (Precision). La precisión calcula la proporción de verdaderos positivos (VP) en relación con el número total de predicciones positivas (VP + FP). Su fórmula es:

$$Precision = \frac{VP}{VP + FP}$$

Sensibilidad (Recall), La sensibilidad mide la proporción de verdaderos positivos (VP) en relación con el número total de casos reales positivos (VP + Falsos negativos, FN). Su fórmula es:

$$Recall = \frac{VP}{VP + FN}$$

F1-Score. El F1-Score combina la precisión y el recall en un solo valor, proporcionando una evaluación equilibrada del modelo. Su fórmula es:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Mientras que la exactitud brinda una visión general del porcentaje de predicciones correctas, la precisión se centra en la calidad de las predicciones positivas, la sensibilidad destaca la capacidad del modelo para

identificar correctamente las instancias positivas y el F1-Score combina precisión y sensibilidad en un solo valor, proporcionando una evaluación equilibrada del modelo.

2.7.2. Métricas para la evaluación continua

Distancia de Levenshtein. La distancia de Levenshtein (Levenshtein, 1966) es una métrica que mide la similitud entre dos secuencias, como las secuencias de señas predichas y las secuencias reales. La distancia de Levenshtein se basa en el número mínimo de operaciones (inserciones, eliminaciones o sustituciones) requeridas para transformar una secuencia en otra, ver Figura 21. Su fórmula es:

$$LevenshteinDistance = D[n, m]$$

donde $D[n, m]$ representa la distancia de Levenshtein entre las secuencias de longitud n y m.

Cadena Inicial	Cambios
monitor	-
contitor	Reemplazo: m por c
con tor	Eliminación: i
contaor	Inserción: n por t
contador	Inserción: d

Figura 21. Ejemplo de distancia de Levenshtein para las palabras "monitor" y "contador".

Word Error Rate (WER). También es importante mencionar la Word Error Rate (WER), una métrica ampliamente empleada en la evaluación de sistemas de traducción automática. WER se deriva de la distancia de Levenshtein, pero con una diferencia clave: mientras que la distancia de Levenshtein opera a nivel de caracteres, WER opera a nivel de palabras. La fórmula para calcular la Word Error Rate es:

$$WER = \frac{S + D + I}{N}$$

donde:

- S es el número de sustituciones (palabras incorrectas).
- D es el número de eliminaciones (palabras omitidas).

- I es el número de inserciones (palabras adicionales).
- N es el número total de palabras en la referencia.

En el transcurso de este capítulo, se exploraron aspectos fundamentales que sientan las bases para la comprensión integral del trabajo en cuestión. Desde los fundamentos de la comunicación en la Lengua de Señas Mexicana (LSM) y su alfabeto manual, hasta las representaciones de datos mediante imágenes y keypoints. Además, se examinaron a fondo los cinco métodos de aprendizaje máquina aplicados en esta investigación. Finalmente, presentamos medidas importantes para evaluar el desempeño, no solo para el presente estudio, sino también como un preámbulo al próximo capítulo, donde se ahondará en el trabajo relacionado.

Capítulo 3. Trabajo relacionado

En este capítulo, exploraremos el trabajo relacionado en el campo del reconocimiento de señas, exclusivamente para la LSM. Comprender el contexto y los avances previos en esta área es esencial para apreciar plenamente la contribución de nuestra investigación.

Organizaremos estos trabajos relacionados utilizando una taxonomía que clasificará los estudios en función de cuatro categorías clave: la modalidad de captura de datos, el tipo de datos, la modalidad de reconocimiento aplicada y el método empleado para evaluar los resultados (Figura 22).

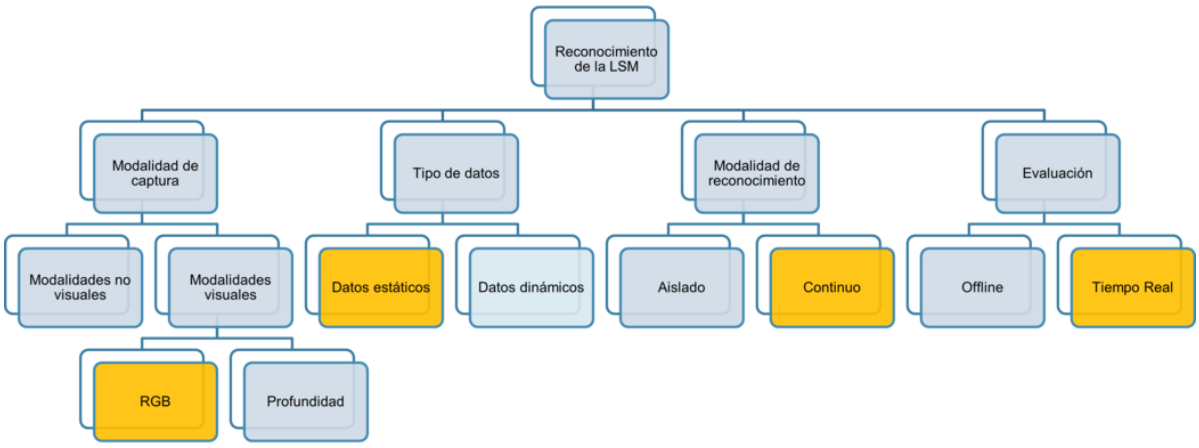


Figura 22. Taxonomía utilizada para la clasificación de trabajos relacionados. En amarillo se resaltan las categorías abordadas en este trabajo de tesis.

3.1. Modalidad de captura

Para organizar y clasificar los métodos utilizados en la captura de datos para el reconocimiento de señas en LSM, hemos segmentado las modalidades en dos categorías principales: visual y no visual. La modalidad visual se basa en el uso de cámaras para registrar las señas, permitiendo una captura precisa de los movimientos y las formas de la mano.

En contraste, la modalidad no visual involucra el empleo de prototipos de hardware específicamente diseñados para captar las formas y movimientos de la mano y otros aspectos relevantes de la comunicación en LSM. Dada la amplia diversidad de tecnologías y enfoques presentes en la modalidad no visual, hemos decidido no introducir subcategorías en esta área.

3.1.1. Modalidades no visuales

De entre la literatura revisada, se encontró un único estudio que utilizó una modalidad de captura datos no visual. El trabajo de Ocampo et al. (2020) presenta el diseño de un prototipo tecnológico para el reconocimiento de señas en LSM (Figura 23). El prototipo consiste en un guante equipado con sensores flexibles y un giroscopio. La información proveniente de estos sensores se transmite mediante la tecnología Bluetooth a un dispositivo Arduino, el cual, a su vez, está conectado a un teléfono inteligente.

El reconocimiento de señas se realiza mediante una aplicación móvil. A pesar de las ventajas que ofrecen las modalidades basadas en hardware, permitiendo un registro preciso de los movimientos de los dedos, suele destacarse su carácter intrusivo.

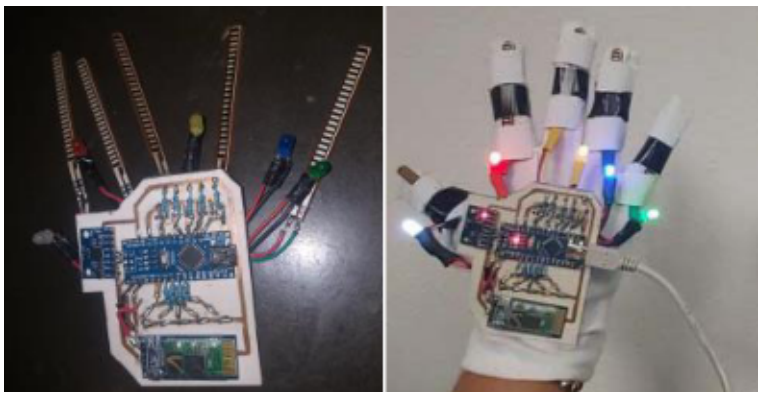


Figura 23. Prototipo utilizado en el estudio de (Ocampo et al., 2020), que comprende un guante equipado con sensores flexibles, un giroscopio, un transmisor Bluetooth y una placa de desarrollo Arduino.

3.1.2. Modalidades visuales

En esta categoría se encuentran los trabajos que utilizan cámaras para realizar la captura de datos, comúnmente cámaras RGB y RGB-D. El uso de cámaras permite la captura no invasiva de información, además de ser típicamente asequibles y portátiles, en gran parte gracias a la incorporación de cámaras en teléfonos inteligentes y dispositivos electrónicos de consumo.

En los últimos años, se ha observado un creciente interés por desarrollar sistemas de reconocimiento de imágenes, especialmente mediante el uso de técnicas de aprendizaje profundo. No obstante, el procesamiento de las imágenes se ve afectado por elementos como la iluminación, el desenfoque y la oclusión.

3.1.2.1. Uso de cámaras RGB

Las cámaras RGB, abreviatura de "Red Green Blue", son dispositivos de captura que registran imágenes a color en el espectro visible de la luz. Su precio accesible y amplia disponibilidad las convierten en una opción popular para diversas aplicaciones de captura de imágenes y reconocimiento de patrones.

El método propuesto por Cervantes et al. (2016) se enfoca en reconocer 249 señas dinámicas de la LSM. Estas señas representan las palabras comúnmente aprendidas por los estudiantes cuando comienzan a estudiar LSM. La captura se realizó con ayuda de 22 participantes en un ambiente controlado. Se extrajeron 743 características (momentos geométricos y características de color) de cada vídeo y utilizando un algoritmo genético se seleccionó el subconjunto de características más relevante. La clasificación se realizó utilizando Máquinas de Soporte Vectorial (SVMs), obteniendo un accuracy promedio del 97 %.

Sin embargo, es importante destacar que este enfoque se centra principalmente en el reconocimiento de los gestos de las manos y no aborda aspectos como las expresiones faciales, movimientos corporales o características no manuales que también forman parte de la comunicación en LSM.

El trabajo de Ramírez Sánchez et al. (2021) propone un sistema de reconocimiento, de 75 señas dinámicas de la LSM. Estas señas abarcan una variedad de categorías, incluyendo saludos, preguntas, términos médicos, verbos, emociones y lugares. Para lograr esto, se emplea la librería MediaPipe para la extracción de puntos de interés en las manos, el cuerpo y el rostro.

La extracción de características se lleva a cabo mediante tres Redes Neuronales Convolucionales (CNNs), una diseñada para cada región del cuerpo (manos, cuerpo y rostro). Para realizar la clasificación, se emplea un perceptrón multicapa. El sistema logra un accuracy promedio del 94.9 %.

Fregoso et al. (2021) explora la optimización de hiperparámetros en CNNs para la clasificación de señas estáticas de la Lengua de Señas Mexicana (LSM) y la Lengua de Señas Americana (ASL). La optimización de hiperparámetros se realiza con la ayuda del algoritmo Particle Swarn Optimization (PSO). Algunos de los hiperparámetros optimizados son el número de capas convolucionales en la red, el número y tamaño de filtros en cada capa y el tamaño del lote (*batch size*) usado en el entrenamiento. El mejor accuracy obtenido para el reconocimiento de señas estáticas en LSM fue de 99.45 %

Martínez-Guevara et al. (2019) presentan un enfoque de aprendizaje no supervisado que se centra en reconocer las unidades fonéticas de 50 señas dinámicas de la LSM. Esto se logra al considerar la configuración manual, la expresión facial, la posición y el movimiento de las manos. El proceso de clasificación

se realiza utilizando estas unidades fonéticas en combinación con las técnicas; algoritmo de Expectation Maximization (EM) y Naive Bayes (NB), logrando un nivel de precisión del 75.8 %. Aunque los resultados son modestos, la propuesta es interesante al descomponer las señas en primitivas.

3.1.2.2. Uso de cámaras RGB-D

Las cámaras RGB-D, abreviatura de "Red Green Blue - Depth", son dispositivos de captura que combinan las capacidades de las cámaras RGB tradicionales con la capacidad de medir la distancia o profundidad de los objetos en la escena. Esta característica de medición de profundidad ha generado un gran interés entre los investigadores, especialmente después de la introducción del sensor Microsoft Kinect, que facilitó enormemente la adquisición de datos RGB-D y la investigación en este campo.

En este contexto, García-Bautista et al. (2017) utilizaron el sensor *Microsoft Kinect* para adquirir las coordenadas 3D de 12 articulaciones del cuerpo en el reconocimiento de señas dinámicas en LSM. Estas señas pertenecen a seis categorías semánticas distintas, incluyendo saludos, familia, entre otras, y se considera el uso de una o ambas manos en su ejecución. La clasificación se llevó a cabo mediante el análisis de las trayectorias de ambas manos utilizando el algoritmo *Dynamic Time Warping (DTW)*. Aunque se obtuvieron resultados satisfactorios, con un promedio de precisión del 99.1 % en la identificación de 20 señas dinámicas, es importante destacar que este enfoque presenta limitaciones en términos de escalabilidad debido a la complejidad computacional inherente al algoritmo DTW.

También, haciendo uso del sensor *Microsoft Kinect*, Sosa-Jiménez et al. (2017) aplican una arquitectura bimodal para el reconocimiento de señas dinámicas. Estas señas abarcan diversas categorías, incluyendo partes del cuerpo, saludos, animales, deportes, entre otras. La metodología combina las coordenadas 3D de las manos y los hombros con los momentos geométricos obtenidos a partir de imágenes 2D para entrenar un *Modelo Oculto de Markov (HMM)*. El sistema logra reconocer con éxito 22 señas dinámicas, demostrando una sensibilidad promedio del 86 % y un F-score del 66 %.

Por otro lado, Mejía-Peréz et al. (2022) emplearon el sensor OAK-D y las librerías DepthAI y MediaPipe para obtener las coordenadas 3D de varios puntos de interés en las manos, el cuerpo y el rostro. En total, se capturaron 30 señas diferentes, de las cuales 4 son señas estáticas pertenecientes al alfabeto manual, mientras que el resto son señas dinámicas. Para la clasificación, se evaluaron tres arquitecturas de aprendizaje profundo: *Red Neuronal Recurrente (RNN)*, *Memoria a Corto Plazo Largo (LSTM)* y

Unidades Recurrentes Cerradas (GRU). El modelo más efectivo logró un accuracy del 97 %. Cabe destacar que este artículo es el único entre los que realizaron captura de datos que proporciona un enlace para descargar el conjunto de datos.

3.2. Tipo de datos

En esta sección, abordaremos la clasificación de trabajos relacionados con el reconocimiento de señas en LSM desde la perspectiva del tipo de datos, centrándonos en sí se trata de datos estáticos o dinámicos. Esta distinción influye en la forma en que se capturan y procesan las señas, así como en las técnicas de reconocimiento aplicadas.

El estudio realizado por Fregoso et al. (2021) se enfoca en el reconocimiento de señas utilizando datos estáticos, empleando tres bases de datos en total. Dos de estas bases de datos están relacionadas con la Lengua de Señas Americana (ASL), mientras que una se centra en la Lengua de Señas Mexicana (LSM). En ambos casos, las señas representan el alfabeto manual de las respectivas lenguas. Es importante destacar que el conjunto de datos de la LSM no está disponible públicamente, es decir, no existe un enlace de descarga disponible para acceder a él.

En la modalidad de señas dinámicas se pueden citar los trabajos de García-Bautista et al. (2017), Sosa-Jiménez et al. (2017), Martínez-Guevara et al. (2019), Cervantes et al. (2016), Mejía-Peréz et al. (2022) y Ramírez Sánchez et al. (2021).

3.3. Modalidad de reconocimiento

Cuando se trata de abordar el reconocimiento de señas en LSM, es importante resaltar las investigaciones realizadas por Sosa-Jiménez et al. (2017) y Ramírez Sánchez et al. (2021). Estos dos estudios se destacan por ser pioneros en el reconocimiento de señas en LSM a nivel de oración, lo que implica el procesamiento de secuencias de señas dinámicas. En este contexto, a nivel de oración, los datos se representan en forma de video, que incluye una o más señas en movimiento, creando así una frase completa en LSM. El trabajo de Sosa se enfocó en 11 oraciones, mientras que el de Ramírez amplió aún más este campo al analizar 22 oraciones.

3.4. Método de evaluación

Los trabajos de Sosa-Jiménez et al. (2017) y Ramírez Sánchez et al. (2021), a pesar de incorporar la frase "Real-time" en sus títulos, carecen de un método de evaluación que permita verificar la capacidad de sus sistemas para operar en tiempo real. En otras palabras, no proporcionan evidencia que respalde la eficacia de la velocidad de inferencia necesaria para llevar a cabo tareas de reconocimiento en tiempo real de manera efectiva.

El trabajo de García-Bautista et al. (2017) menciona haber realizado pruebas en tiempo real. Aunque se obtiene un accuracy promedio satisfactorio (98.57 %), se identifican ciertas problemáticas. En primer lugar, al igual que en los trabajos previamente mencionados, no se incluye una métrica que evalúe el tiempo de inferencia.

En segundo lugar, el empleo de Dynamic Time Warping (DTW) para la clasificación presenta limitaciones en cuanto a escalabilidad debido a la complejidad computacional inherente a este algoritmo. Además, es importante señalar que DTW típicamente requiere acceder a la secuencia temporal completa antes de poder llevar a cabo el alineamiento. En aplicaciones en tiempo real, a menudo se requiere tomar decisiones o acciones tan pronto como los nuevos datos estén disponibles, lo cual puede ser incompatible con la espera de la secuencia completa.

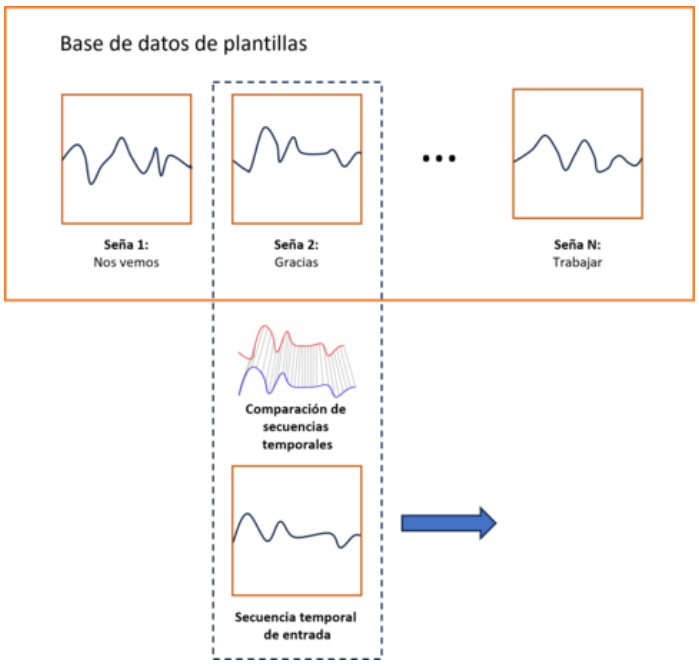


Figura 24. En este ejemplo, cada seña se representa como una secuencia temporal. Cada nueva secuencia de entrada se compara con cada una de las plantillas en la base de datos, lo que conlleva una tarea computacionalmente costosa.

3.5. Trabajo relacionado en otras lenguas señas

Esta sección busca complementar el presente documento dando un vistazo a los trabajos más relevantes de reconocimiento continuo de señas del alfabeto manual de señas en lenguas diferentes a la LSM.

El conjunto de datos ChicagoFSWild destaca como el primer conjunto de datos de deletreo utilizando el alfabeto manual de la Lengua de Señas Americana (ASL). Esta colección se compone de dos versiones, ChicagoFSWild y ChicagoFSWild+, ambas incluyendo breves clips de secuencias de deletreo con señas de la ASL extraídas de videos obtenidos de manera colaborativa en plataformas como YouTube y redes sociales de la comunidad sorda.

El reconocimiento continuo de señas del alfabeto manual en la Lengua de Señas Americana (ASL) se ve notablemente avanzado gracias a dos trabajos destacados. En el primero, Shi et al. (2019) presenta un modelo de extremo a extremo basado en un mecanismo de atención iterativo (Red Neuronal Recurrente basada en atención), prescindiendo de la detección o segmentación explícita de la mano. Este enfoque se centra dinámicamente en regiones de interés y no depende de módulos de detección de mano, segmentación ni estimación de postura. Logrando una precisión de letras del 61.2 %.

MiCT-RANet, desarrollado por Mahoudeau (2020), mejora el enfoque de Shi et al. (2019) al combinar un módulo recurrente de atención visual con una estructura principal MiCT-ResNet. Este sistema logra una precisión del 74.4 % en el reconocimiento de letras, superando a otros enfoques por 19.5 puntos. Además, alcanza una velocidad de 229 FPS utilizando una GPU TITAN RTX y una CPU Core i7 6700K.

3.6. Visión General

La Tabla 1 ofrece una visión general de los trabajos más destacados en el campo del reconocimiento de la Lengua de Señas Mexicana mediante el uso de aprendizaje automático. En caso de que algún trabajo no proporcione datos específicos, se representará en la tabla con un guion "-".

Al revisar estos estudios, se puede apreciar que se han empleado diversas familias de modelos y técnicas para abordar el desafío del reconocimiento de señas en LSM. Estas técnicas incluyen:

- Dynamic Time Warping (DTW): DTW es una técnica basada en plantillas que se ha aplicado en

algunos trabajos. DTW es útil para comparar secuencias de tiempo y encontrar similitudes entre señas en LSM.

- Convolutional Neural Networks (CNN): Las redes neuronales convolucionales se utilizan para el procesamiento de imágenes y se han aplicado en el reconocimiento de señas estáticas en LSM. Estas redes son eficaces para extraer características visuales de las señas.
- Recurrent Neural Networks (RNN): Las redes neuronales recurrentes, como las LSTM y GRU, se emplean para modelar secuencias de señas dinámicas en LSM. Estas redes son especialmente adecuadas para capturar la estructura temporal de las señas en LSM.
- Support Vector Machines (SVM): Las SVM se utilizan en algunos trabajos como un enfoque de clasificación para el reconocimiento de señas en LSM. Estas son útiles para la separación de clases en conjuntos de datos.
- Expectation Maximization (EM): EM es una técnica de modelado estadístico que puede emplearse para estimar parámetros de modelos ocultos en el reconocimiento de señas en LSM.
- Naive Bayes (NB): El modelo de Naive Bayes se ha aplicado en algunos estudios para la clasificación de señas en LSM. Este enfoque se basa en la probabilidad condicional y es útil para la clasificación de datos.
- Hidden Markov Models (HMM): Los HMM son modelos probabilísticos ampliamente utilizados para modelar secuencias y se han aplicado en el reconocimiento de señas en LSM para capturar la estructura temporal y secuencial de las señas.

Luego de examinar en detalle estos trabajos, se ha notado que, solo uno de ellos (Mejía-Peréz et al., 2022) ofrece un enlace funcional para la descarga de su conjunto de datos. Este conjunto de datos se utiliza para el reconocimiento a nivel de palabra (datos dinámicos). A pesar de que se encontró otro trabajo con enlace de descarga, lamentablemente, este no estaba operativo. En cualquier caso, hasta donde llega nuestro conocimiento, no existe un conjunto de datos disponible para el reconocimiento de señas del alfabeto de la LSM (datos estáticos).

A pesar de la existencia de dos trabajos que abordan el desafío del reconocimiento continuo de señas (Sosa-Jiménez et al., 2017; Ramírez Sánchez et al., 2021) en LSM, ninguno de estos trabajos presenta una métrica que permita evaluar su viabilidad para aplicaciones en tiempo real.

Tabla 1. Tabla comparativa de trabajos para el reconocimiento automático de señas en LSM. Abreviaturas utilizadas: Accuracy (Acc) y F1-Score (F1).

Trabajo	Dispositivo	Datos	Modalidad	Evaluación	No. de clases	Clasificador	Resultados
Ocampo et al. (2020)	No visual	-	-	-	-	-	88.3 % (Acc)
Fregoso et al. (2021)	Cámara RGB	Estáticos	Aislada	Offline	21	CNN	98.91 % (Acc)
García-Bautista et al. (2017)	Kinect	Dinámicos	Aislada	Tiempo real	20	DTW	98.57 % (Acc)
Mejía-Peréz et al. (2022)	OAK-D	Dinámicos	Aislada	Offline	30	RNN, LSTM, GRU	96.44 % (Acc)
Cervantes et al. (2016)	Cámara RGB	Dinámicos	Aislada	Offline	249	SVM	97.04 % (Acc)
Martínez-Guevara et al. (2019)	Cámara RGB	Dinámicos	Aislada	Offline	50	EM, NB	75.80 % (Acc)
Sosa-Jiménez et al. (2017)	Kinect	Dinámicos	Continua	Tiempo real	22	HMM	69 % (F1)
Ramírez Sánchez et al. (2021)	Cámara RGB	Dinámicos	Continua	Tiempo real	74	CNN, HMM	94.1 % (Acc)

Por tanto, nuestro trabajo busca aportar al área en dos maneras, la primera de ella es generar un dataset públicamente accesible para el problema de reconocimiento de señas estáticas en LSM. Esto permitiría que los resultados de diferentes investigaciones sean comparables entre sí. En segundo lugar, nos esforzamos por desarrollar modelos que puedan implementarse en aplicaciones de tiempo real, lo que implica garantizar una inferencia rápida.

Capítulo 4. Reconocimiento de señas estáticas en modalidad aislada

En este capítulo, presentaremos en detalle la metodología que seguimos para abordar el reconocimiento de señas estáticas en la modalidad aislada (Figura 25). Nuestro enfoque se divide en cinco etapas principales que abarcan desde la captura de datos hasta la evaluación de los modelos.

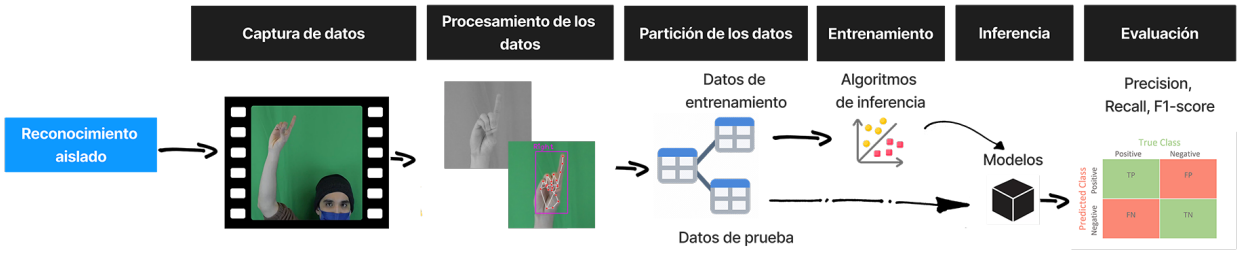


Figura 25. Metodología utilizada para el reconocimiento de señas estáticas en la modalidad aislada.

En la Sección 4.1 comenzaremos explicando cómo recopilamos los datos para entrenar y evaluar nuestros modelos. Se abordará tanto la configuración del escenario de captura como el protocolo de captura utilizado. En la Sección 4.2 describiremos cómo se prepararon los datos, esto incluye el procesamiento y transformación de los datos. Además, en esta etapa, generamos dos representaciones de los datos: una basada en keypoints de la mano y otra basada en imágenes.

En la Sección 4.3 discutiremos cómo dividimos nuestros datos en conjuntos de entrenamiento y prueba, basada en la asignación de un identificador único a cada participante. Para la Sección 4.4 presentaremos la arquitectura de los modelos, los ajustes de hiperparámetros y el proceso de entrenamiento.

Finalmente, en la Sección 4.5 proponemos una estrategia de evaluación, buscando probar la robustez del modelo ante diferentes ambientes. Este enfoque incluye la validación del sistema de reconocimiento con señas realizadas desde diferentes ángulos y orientaciones, asegurando así su capacidad para generalizar efectivamente en entornos variados.

4.1. Captura de datos en la modalidad aislada

En esta sección, se describirá en detalle el proceso de captura de datos utilizado para adquirir las señas estáticas en la modalidad aislada. Este proceso se llevó a cabo en un entorno cuidadosamente diseñado para garantizar la calidad y la consistencia de los datos capturados.

4.1.1. Configuración del escenario de captura

La calidad de los datos de entrada en un estudio de reconocimiento de señas estáticas es de suma importancia, y la configuración del escenario de captura desempeña un papel fundamental en este aspecto. Por ello se cuidaron tres aspectos: el fondo, la iluminación y la configuración de la cámara (Figura 26).

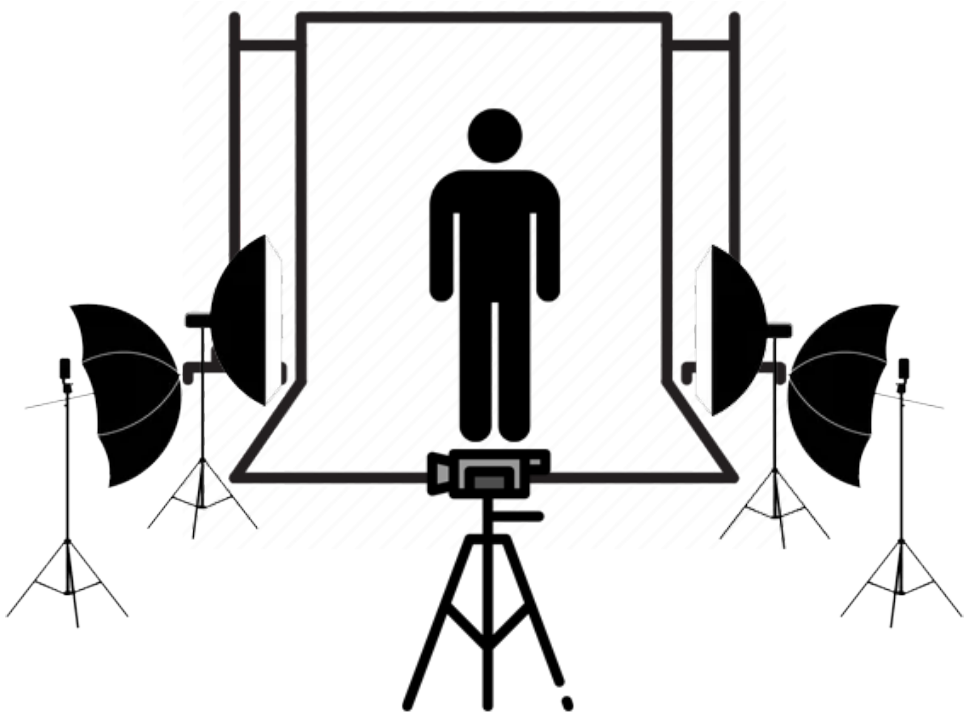


Figura 26. Configuración del escenario de captura.

Pantalla verde. La decisión de utilizar una pantalla verde como fondo se basó en varios factores clave. En primer lugar, para facilitar el trabajo del detector de manos de MediaPipe, permitiéndole identificar con precisión las manos y las señas al separarlas claramente del fondo. Además, la pantalla verde contribuyó a la generación de un conjunto de datos con un fondo uniforme y consistente, lo que simplificó el procesamiento y manipulación de los datos para su posterior análisis.

Iluminación uniforme. La uniformidad de la iluminación es esencial para evitar sombras no deseadas en las imágenes capturadas. Para lograr esto, se utilizaron dos "softbox" dirigidos hacia la pantalla verde, eliminando así las sombras generadas por el participante. Además, se emplearon dos sombrillas difusoras de luz para iluminar al participante de manera uniforme, garantizando que las señas fueran capturadas de manera clara.

Configuración de la cámara. La cámara utilizada se configuró para capturar video a una resolución de 1920x1080 pixeles, con una tasa de captura de 60 cuadros por segundo. Se optó por una velocidad de obturación de 1/649 para reducir el efecto de desenfoque por movimiento (motion blur), una característica crucial para obtener imágenes nítidas de las señas realizadas por los participantes.

Aunque esta configuración podría haber resultado en imágenes más oscuras debido a la menor cantidad de luz que entra por el sensor (por la velocidad del obturador), esto se compensó mediante el aumento de la sensibilidad a la luz de la cámara (valor ISO) y la buena iluminación del escenario de captura.

4.1.2. Proceso de captura de datos

Los datos utilizados en el reconocimiento aislado se capturaron con la ayuda de 20 participantes con diferentes niveles de familiaridad con la LSM, que variaban desde la completa inexperiencia hasta un nivel intermedio de competencia en el lenguaje. Cada participante grabó 21 señas distintas del alfabeto manual de la Lengua de Señas Mexicana, con la excepción de las señas dinámicas J, K, Ñ, Q, X y Z. Cada una de estas señas se grabó en videos que incluyen variaciones en distintos ángulos y orientaciones de la mano.

A pesar de que se capturaron videos durante esta fase, posteriormente se procesaron para extraer cuadros individuales, creando así un dataset de imágenes. Sin embargo, los detalles específicos de este proceso se abordarán con mayor profundidad en la Sección 4.2.

Grados de libertad. Antes de adentrarnos en las diversas variaciones de los datos, es esencial familiarizarnos con los *seis grados de libertad* (Figura 27) en los que una mano se puede mover, también conocido como espacio de configuraciones. Estos abarcan los movimientos de traslación en las direcciones de adelante/atrás, arriba/abajo y izquierda/derecha, así como las rotaciones alrededor de los ejes X, Y y Z, comúnmente conocidas como cabeceo (pitch), alabeo (roll) y guiñada (yaw). Para tener una referencia común en las próximas descripciones, consideraremos el movimiento de alabeo visto de frente.

En las sesiones de captura, cada participante fue instruido en cómo realizar cada una de las 21 señas (estáticas) distintas del alfabeto manual de la LSM. Se les proporcionó una descripción y una demostración visual de cómo realizar cada seña. Una vez que los participantes estuvieron familiarizados con las señas, se procedió a capturar los videos.

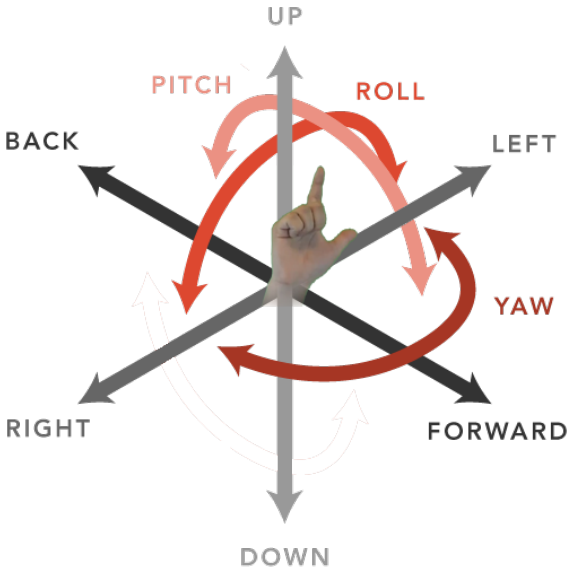


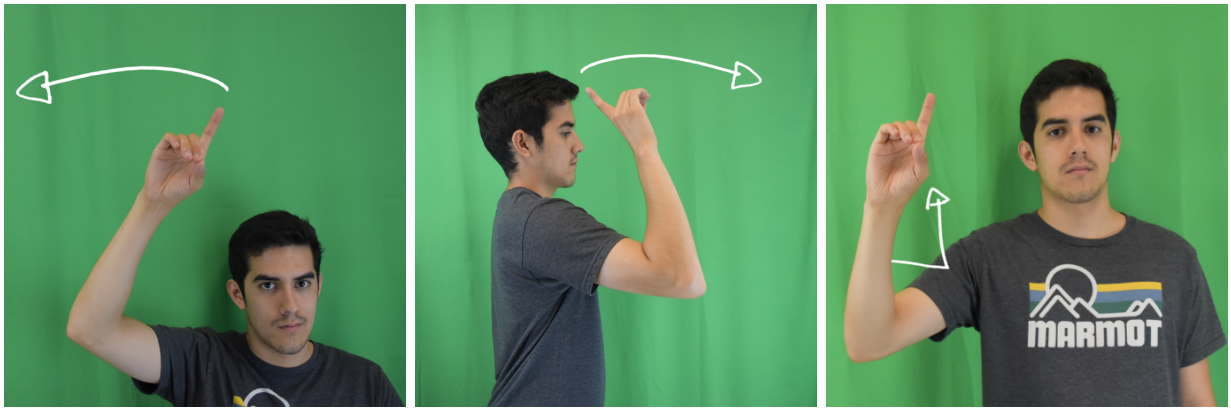
Figura 27. Los seis grados de libertad: adelante/atrás (forward/back), arriba/abajo (up/down), izquierda/derecha (left/right), cabeceo (pitch), guiñada (yaw), alabeo (roll).

Para lograr una representación diversa de las señas, se diseñaron tres grupos de variaciones que permitieran capturar la misma seña desde diferentes ángulos y orientaciones. Estos grupos, que denominamos grupos A, B y C, consistieron en movimientos específicos que se realizaron para cada seña. A continuación, se detallará en profundidad la estructura y el propósito de cada uno de estos grupos de variaciones.

Grupo A: En este nivel, se grabó un video de tres segundos por cada seña realizada por los participantes (Figura 28). Cada participante realizó movimientos suaves (un máximo aproximado de 30º por cada grado de libertad) en todos los grados de libertad, con el propósito de generar imágenes con poca variabilidad.

Figura 28. Participante realizando movimientos suaves en todos los grados de libertad.

Grupo B: En este nivel, se grabaron tres videos de un segundo de duración para cada seña realizada por los participantes. Cada participante realizó las tres rotaciones posibles: cabeceo, alabeo y guiñada. Este grupo se diseñó para capturar la mano desde diferentes ángulos.



a) Movimiento de alabeo. b) Movimiento de cabeceo. c) Movimiento de guiño.

Figura 29. Participante realizando movimientos en las tres rotaciones posibles.

En la primera variación, se solicitó a los participantes que realizaran un movimiento de arco de lado a lado (alabeo) (Figura 29a). En la segunda variación, se les pidió que realizaran un movimiento de arco de atrás hacia adelante (cabeceo) (Figura 29b). Finalmente, en la tercera variación, se les solicitó que giraran su antebrazo (guiñada) (Figura 29c).

Grupo C: En este nivel, se grabó un video de tres segundos por cada seña realizada por los participantes (Figura 30). Cada participante realizó movimientos pronunciados en todos los grados de libertad[1], con el objetivo de generar imágenes altamente variables y poner a prueba la robustez de los modelos de clasificación.

En nuestra opinión, algunos datasets de alfabetos en otras lenguas de señas, como la Lengua de Señas Americana, parecen estar limitados en cuanto a diversidad, ya que se componen principalmente de imágenes con poca variabilidad, lo que nosotros caracterizaríamos como el "grupo A".

Por esta razón, hemos incorporado las variaciones presentes en los "grupos B y C" con el objetivo de entrenar y evaluar modelos en condiciones que no solo reflejen escenarios ideales, sino también situaciones más desafiantes.

[1]Aunque no se restringieron los grados en que se podía girar la mano, anatómicamente la muñeca tiene restricciones de movimiento.

Figura 30. Participante realizando movimientos pronunciados en todos los grados de libertad.

4.2. Procesamiento de los datos

En esta Sección, se abordará en detalle el proceso que se siguió para convertir las grabaciones de video de las señas en datos aptos para el entrenamiento y evaluación de los modelos. Esto incluye la extracción de imágenes individuales de los videos y la adaptación de los datos a las necesidades específicas de los distintos enfoques de aprendizaje automático (Figura 31).

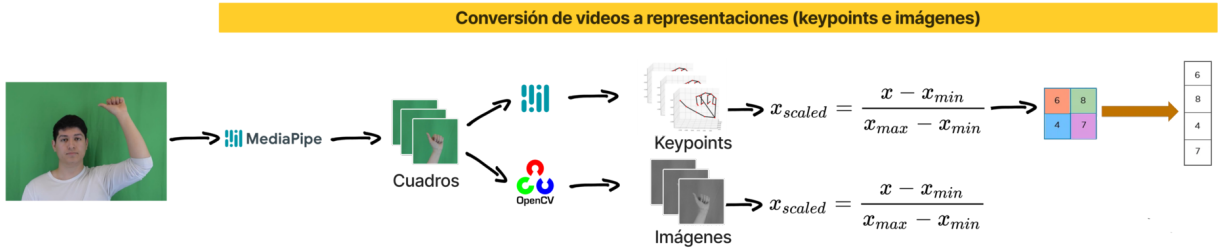


Figura 31. Vista general del procesamiento de los datos.

Segmentación. El primer paso en el procesamiento de los datos consistió en segmentar la mano a partir de los cuadros del video. Para lograrlo, se extrajo cada cuadro del video y se utilizó la librería MediaPipe para detectar la mano derecha del participante (todos los participantes eran diestros y usaron la mano derecha para realizar las señas).

Una vez que se localiza la mano, se determina su centro y se realiza un recorte. Esto resulta en imágenes

de 360x360 pixeles, las cuales denominaremos "Recortes de la Mano". Estas imágenes sirvieron como base para la generación de las dos representaciones de datos utilizadas en este trabajo: una basada en keypoints de la mano y otra basada en imágenes. A continuación, se describirá en detalle el proceso de generación de ambas representaciones.

4.2.1. Generación de representación basada en keypoints

Extracción de keypoints. Utilizando los "Recortes de la Mano", se extrajeron los keypoints de la mano con la ayuda, nuevamente, de la librería MediaPipe. Es importante destacar que los keypoints proporcionados por MediaPipe están normalizados en relación con el ancho y el alto de la imagen original. Para garantizar que nuestro sistema sea invariante a la escala, aplicamos una técnica de normalización conocida como *Min-Max Normalization*.

Normalización de keypoints. Los keypoints se normalizaron utilizando la técnica conocida como Min-Max Normalization. Esta técnica es un método de procesamiento de datos que escala los valores de las características dentro de un rango específico, en este caso, entre 0 y 1. La fórmula para la normalización Min-Max es la siguiente:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

donde X representa el valor original de la característica, X_{min} es el valor mínimo de esa característica en todo el conjunto de datos y X_{max} es el valor máximo de la misma característica en el conjunto de datos.

Esta normalización se aplicó por separado a las componentes X, Y y Z de los keypoints, asegurando así que cada componente tuviera sus valores normalizados dentro del rango [0, 1]. Este enfoque permite que el proceso de reconocimiento sea invariante a la escala y se centre únicamente en la forma de la mano, independientemente de la distancia entre la mano y la cámara.

Aplanamiento (flattening). Después de que los keypoints fueron normalizados, se sometieron a un proceso de aplanamiento conocido como *flattening*. En este paso, los datos se transformaron de su

estructura original en un vector unidimensional. En el caso de los keypoints de la mano, esto resultó en un vector de tamaño 63, ya que se tenían 21 keypoints detectados, y cada uno constaba de 3 componentes (X, Y, Z). Este vector resultante está listo para ser utilizado por los métodos de aprendizaje automático, como el Análisis Procrusteano, las Máquinas de Soporte Vectorial y las Redes Neuronales Profundas.

4.2.2. Generación de representación basada en imágenes

Escalamiento y conversión a escala de grises. Los "Recortes de la Mano" se procesaron mediante un redimensionamiento a una resolución de 128x128 pixeles y se transformaron en imágenes a escala de grises. Estos datos están ahora listos para ser utilizados en la Red Neuronal Convolucional (CNN) y la Memoria Asociativa Entrópica (AEM). Nos referiremos a estas imágenes como 'Imágenes a escala de grises".

Normalización de imágenes. Las "Imágenes a escala de grises" también pasaron por un proceso de normalización. Teniendo en cuenta que los valores de pixeles se encuentran dentro del rango [0, 255], se aplicó una normalización dividiendo cada valor de pixel entre 255, siendo esta técnica un caso particular de la normalización Min-Max, para $X_{min} = 0$ y $X_{max} = 255$. Cuando los valores están en una escala más pequeña, como [0, 1], el proceso de entrenamiento de las redes CNNs se vuelve más eficiente y efectivo.

4.3. Partición de los datos

Los datos utilizados en este estudio se obtuvieron con la colaboración de 20 participantes. De estos, se seleccionaron de manera aleatoria 18 participantes cuyos datos fueron utilizados en el entrenamiento de los modelos, mientras que los datos de 2 participantes se reservaron para la evaluación.

Esta estrategia de división se implementó con el propósito de evaluar la capacidad de generalización de los modelos en relación con una amplia gama de características físicas y estilos de señas, contribuyendo así a mejorar su capacidad de reconocimiento en situaciones del mundo real. De esta forma, los modelos se evaluarán con datos que los modelos no han visto o no conocen.

En la Figura 32 se presentan imágenes de los 20 participantes realizando la seña correspondiente a la letra A. En el lado izquierdo de la figura, se muestran imágenes de los 18 participantes que forman parte del conjunto de entrenamiento, mientras que en el lado derecho se presentan las imágenes de los dos participantes reservados exclusivamente para el conjunto de pruebas.

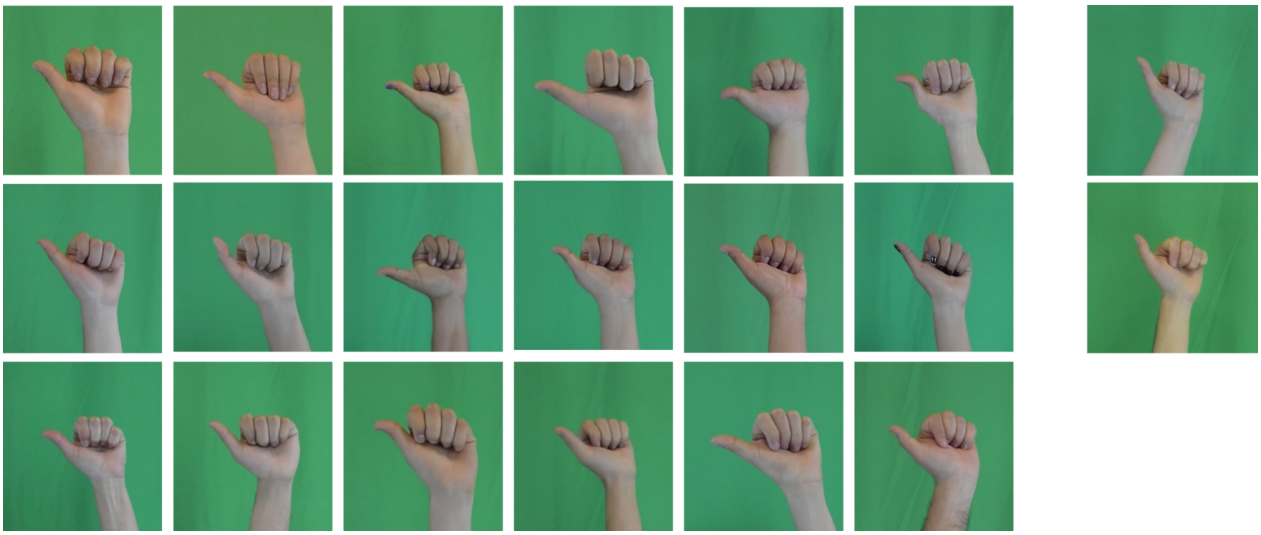


Figura 32. Ejemplo de imágenes para la letra A en el conjunto de datos. En el lado izquierdo, las imágenes son del conjunto de entrenamiento (18 participantes); en el lado derecho, del conjunto de prueba (2 participantes).

4.4. Entrenamiento y ajuste de parámetros de los modelos

En esta sección, nos adentraremos en la etapa de entrenamiento y ajuste de parámetros de nuestros modelos para el reconocimiento de señas estáticas. Comenzaremos con el Análisis Procrusteano, el cual no requiere un proceso de entrenamiento y servirá como punto de referencia o *baseline* para el resto de modelos. Después, abordaremos el entrenamiento de los dos modelos restantes que utilizan los keypoints de las manos como representación, concretamente SVM y DNN. Finalmente, exploraremos los modelos que utilizan imágenes en escala de grises como representación.

4.4.1. Análisis Procrusteano + keypoints

En el Análisis Procrusteano es necesario contar con una plantilla representativa por cada clase. Para seleccionar dicho ejemplo representativo, se opta por el uso del medoide. El medoide es el punto de datos

que minimiza la distancia promedio a todos los demás puntos de datos de la misma clase (Figura 33). En términos simples, es el ejemplo que se encuentra más cerca del centro de su propia clase en el espacio de características.

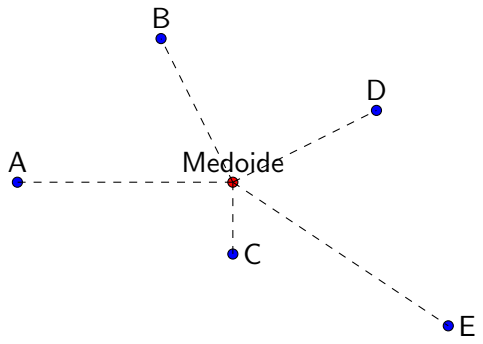


Figura 33. Ilustración del medoide en un conjunto de datos.

Se han elegido 21 plantillas representativas, una por cada clase, mediante la selección del medoide de cada clase. Durante la etapa de inferencia, cada nuevo ejemplo se compara con estas plantillas mediante el Análisis Procrusteano, lo que produce un valor de disimilitud en cada comparación. Para realizar la predicción final, se identifica la clase asociada al valor de disimilitud más bajo, determinando así la clase asignada al ejemplo.

4.4.2. Máquinas de Soporte Vectorial + keypoints

Para entrenar las Máquinas de Soporte Vectorial con keypoints de la mano como datos de entrada, se realizó un proceso de experimentación donde se probaron varios kernels y posteriormente diferentes grados (para el kernel polinomial). Este enfoque de prueba y ajuste gradual permitió identificar una configuración adecuada para el problema en cuestión. Después de explorar diversas opciones, se determinó que el kernel polinomial de grado 3 ofrecía un buen rendimiento en términos de rendimiento.

4.4.3. Red Neuronal Profunda + keypoints

Arquitectura general de la DNN. La Figura 34 presenta la arquitectura general de la DNN utilizada en este trabajo. Se trata de una red completamente conectada, también conocida como *fully connected*

neural network, en la que todas las unidades en una capa están conectadas a todas las unidades en la capa siguiente. A continuación, se detallarán las características de cada una de las capas que componen esta red.

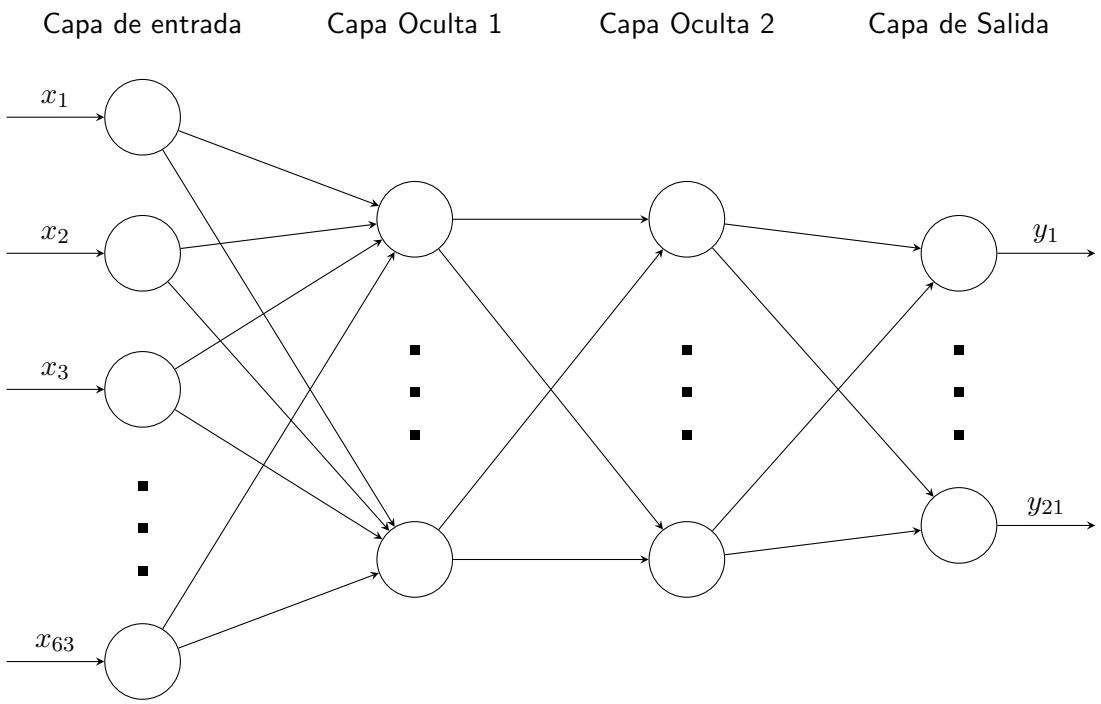


Figura 34. Diagrama de una red neuronal con dos capas ocultas.

Capa de entrada y primera capa oculta. La red comienza con una capa de entrada que tiene un tamaño de 63, lo que corresponde al número de características de los keypoints de la mano. A continuación se encuentra la primera capa oculta, que consta de 12 unidades y utiliza una función de activación de ReLU.

Segunda capa oculta. La red continúa con una segunda capa oculta que contiene 8 unidades, también activadas por ReLU. Esta capa busca extraer características más abstractas y de mayor nivel a partir de las representaciones previamente aprendidas.

Capa de salida. Finalmente, la capa de salida está compuesta por 21 unidades, igualando el número de etiquetas de clase en el conjunto de datos. Utiliza la función de activación Softmax para producir probabilidades de pertenencia a cada clase. La utilización de Softmax contribuye a obtener una distribución de probabilidad normalizada, asegurando que la suma de las probabilidades sea igual a uno, lo que es esencial para la correcta asignación de clases en el problema de clasificación.

Parámetros de entrenamiento. Durante el entrenamiento, se implementó un ajuste en la tasa de aprendizaje (learning rate reduction) que reducía la tasa de aprendizaje cada tres épocas sin mejora en la métrica de pérdida (en los datos de validación). También se aplicó una parada temprana (early stopping) si la métrica de pérdida no mostraba mejoras durante siete épocas consecutivas. Además, se empleó un tamaño de lote (batch size) de 512 y se estableció un límite máximo de 200 épocas.

4.4.4. Red Neuronal Convolucional + imágenes

Arquitectura general de la CNN. La Figura 35 ilustra la arquitectura general de la CNN que ese utiliza en este trabajo. A continuación, se detallan las características clave de cada capa en la arquitectura de la CNN.

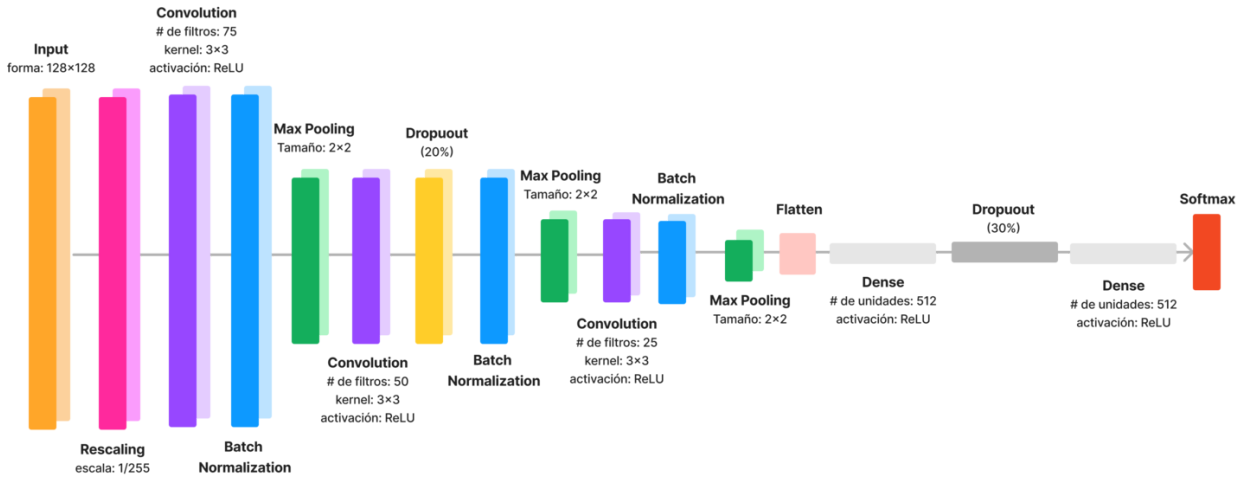


Figura 35. Arquitectura general de la red CNN.

Capa de entrada. La red neuronal comienza con imágenes en escala de grises de 128x128 pixeles. Estas imágenes se escalan para que los valores de pixeles estén en el rango [0, 1]. Escalar los valores de pixeles de las imágenes es una práctica común que tiene como beneficio mejorar la convergencia durante el entrenamiento de la CNN.

Capas convolucionales y Max Pooling. La CNN emplea tres capas convolucionales, cada una de las cuales utiliza una función de activación ReLU. Después de cada capa convolucional, se aplica una operación de Max Pooling para reducir la dimensionalidad y conservar las características más relevantes.

Regularización. La regularización conforma un conjunto de técnicas utilizadas para evitar el sobreajuste. En la CNN se utiliza una capa de Dropout después de la segunda capa convolucional y otra después de la primera capa densa. Además, se implementa la normalización por lotes (Batch Normalization) en cada capa convolucional para acelerar el entrenamiento y mejorar la estabilidad del modelo.

Capas densas (fully connected) y capa de aplanamiento (Flatten). Después de las capas convolucionales y de Max pooling, los datos se redimensionan mediante una capa Flatten, que convierte el espacio de características bidimensional en uno unidimensional. A continuación, estos datos pasan por dos capas densas, cada una compuesta por 512 unidades y activadas mediante la función ReLU. Estas capas están diseñadas para aprender características de alto nivel que son fundamentales para la clasificación de las señas.

Capa de salida. La capa de salida de la CNN tiene un número de unidades igual al número de etiquetas de clases en el conjunto de datos. Se utiliza una función de activación Softmax para producir una distribución de probabilidad sobre las clases, permitiendo la clasificación de las señas.

Parámetros de entrenamiento. En cuanto a los parámetros de entrenamiento, se utilizó un tamaño de lote (batch size) 128 y se estableció un límite de 300 épocas para evitar entrenamientos prolongados. Además, igual a la estrategia utilizada en la DNN, se implementó una reducción en la tasa de aprendizaje (learning rate reduction) que disminuía la tasa de aprendizaje cada tres épocas, sin observar mejoras en la métrica de pérdida, evaluada en los datos de validación. Asimismo, se aplicó una técnica de parada temprana (early stopping) en caso de que la métrica de pérdida no mostrara mejoras durante siete épocas consecutivas.

4.4.5. Memoria Asociativa Entrópica + imágenes

El sistema de reconocimiento de señas estáticas utilizando la Memoria Asociativa Entrópica consta de tres componentes principales: un codificador (encoder), un registro de memoria asociativa y una red totalmente conectada (fully connected) (Figura 36). A continuación, describiremos detalladamente cada uno de estos componentes para comprender mejor cómo funcionan en conjunto para el reconocimiento de señas estáticas.

El proceso inicia con la transformación de las imágenes de entrada (128x128 pixeles en escala de grises) en una representación amodal abstracta compatible con el registro de memoria asociativa mediante el uso de un autoencoder. Este autoencoder consta de dos partes fundamentales: el encoder, que reduce las imágenes a una representación codificada de 1024 valores (características), y el decoder, que reconstruye las imágenes originales a partir de esta representación codificada.

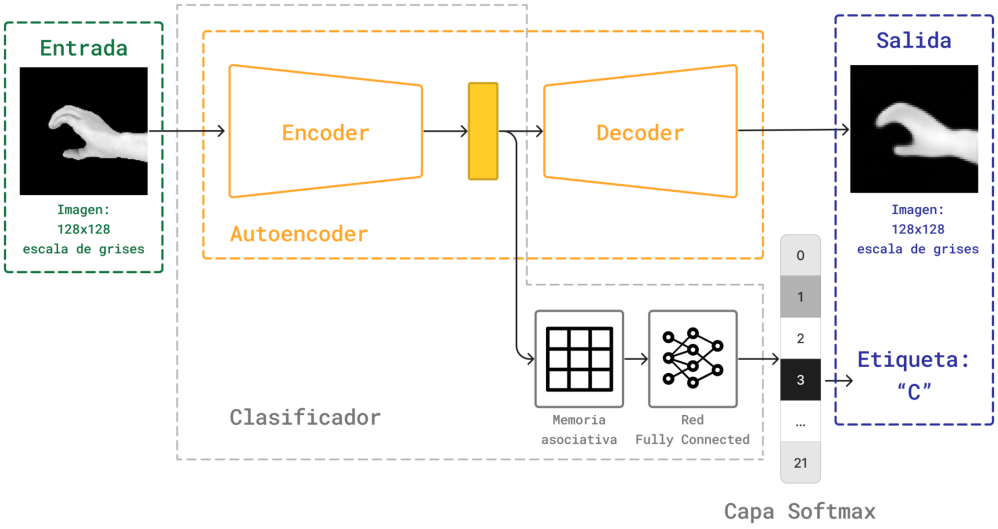


Figura 36. Arquitectura del sistema utilizado en el reconocimiento de señas estáticas utilizando AEM

Posteriormente, las características resultantes se someten a un proceso llamado cuantificación, que convierte las características normalizadas en valores discretos entre 0 y el tamaño de la memoria menos uno. Estas características cuantificadas se utilizan para el registro en la memoria asociativa.

En esta memoria asociativa, se almacenan los ejemplos de entrenamiento mediante la operación de registro λ. Luego, se emplea una red completamente conectada para el proceso de reconocimiento a través de la operación de reconocimiento η. Cabe destacar que la operación de recuperación β no se aborda en este estudio, ya que está relacionada con la generación de señas y va más allá del alcance de esta investigación.

Para definir el tamaño de la memoria asociativa, se evaluaron distintos tamaños de memoria, representados como $1024 \cdot 2^n$, donde 1024 es el número de características obtenidas del autoencoder. Los resultados, como se muestra en la Figura 37, indicaron que el valor más adecuado para "n" fue igual a dos, ya que ofreció un equilibrio favorable entre la precisión y el recall en las métricas de evaluación.

4.5. Estrategia de evaluación para el nivel estático

Recordando la Sección 4.1 que trata sobre la captura de datos para el reconocimiento en modalidad aislada, los datos se dividieron en tres grupos de variaciones. El grupo A (movimientos suaves en todos los grados de libertad), el grupo B (movimientos en tres rotaciones diferentes) y el grupo C (movimientos pronunciados en todos los grados de libertad).

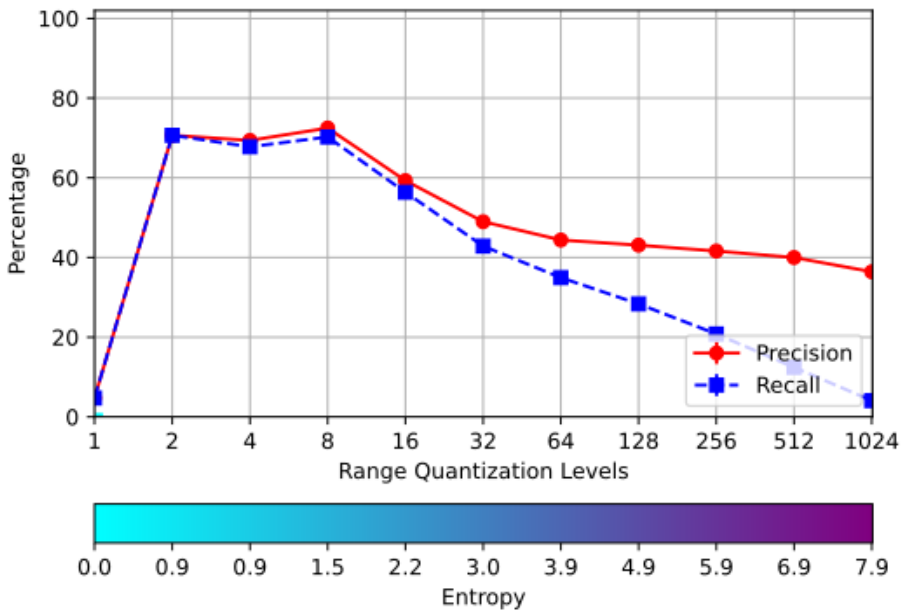


Figura 37. Rendimiento de la AEM para diferentes tamaños de registro de memoria.

Dado que contábamos con diferentes grupos de variaciones, optamos por una estrategia de evaluación que involucraba múltiples combinaciones (Figura 38). Inicialmente, entrenamos un modelo utilizando exclusivamente datos del grupo A y lo evaluamos tanto con datos de ese mismo grupo (A), como con datos de los grupos B y C.

Luego, procedimos a entrenar con datos del grupo B y probamos el modelo con todos los grupos disponibles (A, B, y C), repitiendo este proceso con el grupo C. Finalmente, realizamos una evaluación global, utilizando datos de todos los grupos tanto para el entrenamiento como para la evaluación. Esta estrategia nos permitió evaluar el desempeño del modelo en una variedad de escenarios y conjuntos de datos.

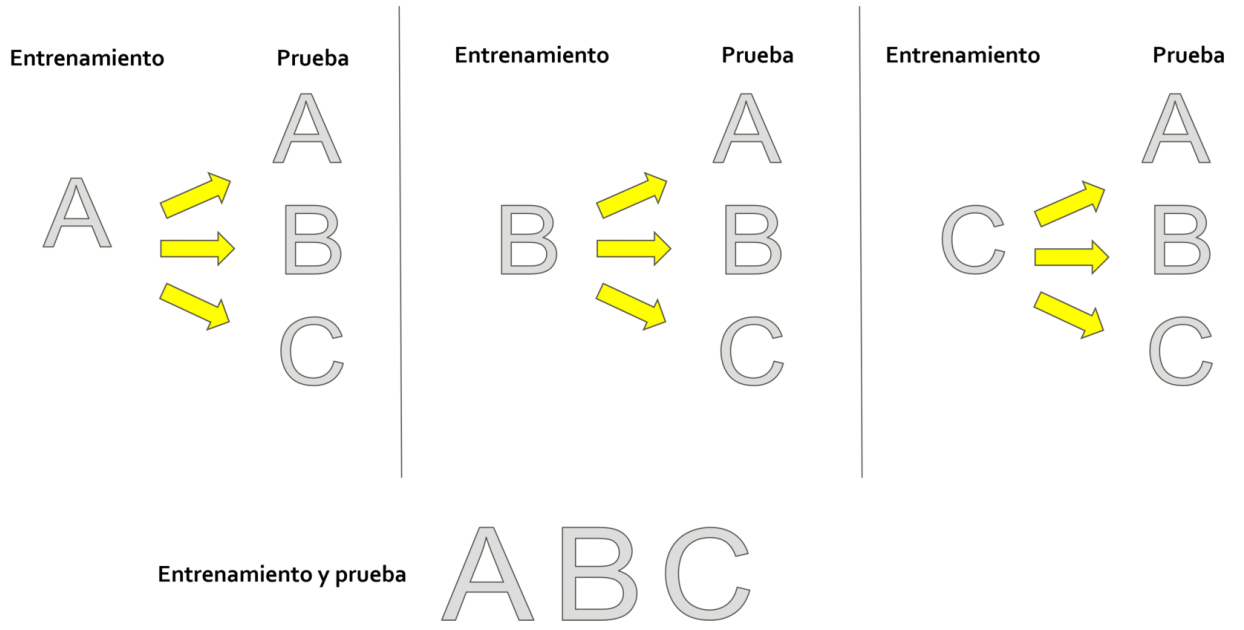


Figura 38. Enfoque de combinaciones utilizado para evaluar el rendimiento de los algoritmos de inferencia.

Como mencionamos previamente, al examinar conjuntos de datos en otras lenguas de señas, como la Lengua de Señas Americana, hemos observado que las señas se caracterizan principalmente por imágenes con una variabilidad limitada. Por consiguiente, consideramos que en muchos de los estudios existentes en la literatura, la evaluación se limita a la modalidad "A" (entrenamiento) - "A" (prueba).

Recapitulando, la recopilación de datos se llevó a cabo en un escenario de captura con óptimas condiciones: bien iluminado, fondo uniforme y una cámara de alta resolución, garantizando la obtención de datos de alta calidad.

Se capturaron 21 señas del alfabeto manual de la LSM desde múltiples ángulos y orientaciones para diversificar el dataset, con el objetivo de crear modelos efectivos tanto en escenarios ideales como desafiantes.

Estos datos se procesaron para generar dos representaciones distintas, una basada en imágenes y otra basada en keypoints. Además, se abordó el ajuste de hiperparámetros específico para cada modelo empleado en el estudio.

Se diseñó un enfoque para dividir los datos utilizando los identificadores únicos de cada participante. Además, se introdujo un método de evaluación que considera y saca provecho de las variaciones presentes en las señas capturadas.

Capítulo 5. Reconocimiento de señas estáticas en modalidad continua

5.1. Captura de datos para el reconocimiento continuo

Los datos utilizados en el reconocimiento continuo se capturaron con la ayuda de seis participantes, divididos en dos grupos. El primer grupo, denominado *semi-expertos* comprende tres participantes con un nivel intermedio de competencia en la LSM. El segundo grupo, denominado *inexpertos*, consta de tres participantes sin conocimiento previo de esta lengua.

Cada participante grabó un total de 20 videos deletreando un total de 20 palabras, el corpus de palabras busca incluir todas las letras del alfabeto manual de la LSM (excluyendo aquellas con un componente dinámico) al menos una vez. El corpus se conforma de las siguientes palabras:

1. A-G-U-A
2. B-O-T-E
3. C-A-M-A
4. D-E-D-O
5. E-L-E-F-A-N-T-E
6. F-O-C-A
7. G-A-T-O
8. H-A-C-H-A
9. I-S-L-A
10. L-U-P-A
11. M-O-N-O
12. N-U-B-E
13. O-R-O
14. P-A-T-O
15. R-A-N-A
16. T-A-C-O
17. U-V-A
18. V-A-S-O
19. W-I-F-I
20. Y-O-Y-O

Las grabaciones se realizaron de la siguiente manera: a cada participante se le indicó la palabra que debía deletrear. Para los participantes inexpertos, se proporcionaron instrucciones detalladas sobre cómo realizar cada una de las señas que componían cada palabra. En cambio, para los semi-expertos, no se proporcionaron instrucciones adicionales y realizaron las señas según su conocimiento previo. En ambos casos, se les dio tiempo para practicar cada palabra antes de iniciar con la grabación.

Después de familiarizarse con la palabra que debían señar, a cada participante se le pidió que comenzara en una posición de reposos con los brazos hacia abajo, señar la palabra indicada y luego regresara a la posición inicial. Los videos solo se volvieron a grabar en caso de que los participantes cometieran un error, es decir, si durante la grabación cambiaban letras o se equivocaban en las configuraciones manuales.

Ni a los participantes inexpertos ni a los semi-expertos se les indicó una velocidad específica para realizar las señas. Esto se hizo con el fin de evaluar si el sistema es capaz de reconocer señas realizadas por

individuos con diferentes niveles conocimiento en la LSM.

La configuración del escenario de captura se mantuvo consistente con la utilizada para la captura de datos en la modalidad aislada, misma que se describió en la Subsección 4.1.1.

5.2. Metodología utilizada en reconocimiento continuo

Para esta evaluación, se aprovechó el modelo de mejor rendimiento en la modalidad aislada. Este modelo se entrenó utilizando la totalidad de los datos, es decir, los datos de los 20 participantes, incorporando tanto los conjuntos de entrenamiento como los de prueba, además de todas las variaciones (A, B y C). Para que los datos fueran compatibles con este modelo, se aplicó un procesamiento igual al presentado en la Subsección 4.2.1.

Ventana deslizante y eliminación de las repeticiones. Para adaptar los modelos utilizados en el reconocimiento aislado y llevarlos al dominio continuo se implementó una estrategia de ventana deslizante. Durante la evaluación, se procesaron secuencias de cuadros de señas y se realizó una predicción para cada cuadro individualmente. Si se detecta que una misma predicción se repite durante un número mínimo de cuadros consecutivos (en este trabajo, se fijó un tamaño de ventana de 12 cuadros), se agrega esta predicción a una lista. Luego, se procede a eliminar las repeticiones consecutivas en la lista, de modo que se obtenga una secuencia de predicciones que representan una palabra deletreada (Figura 39).

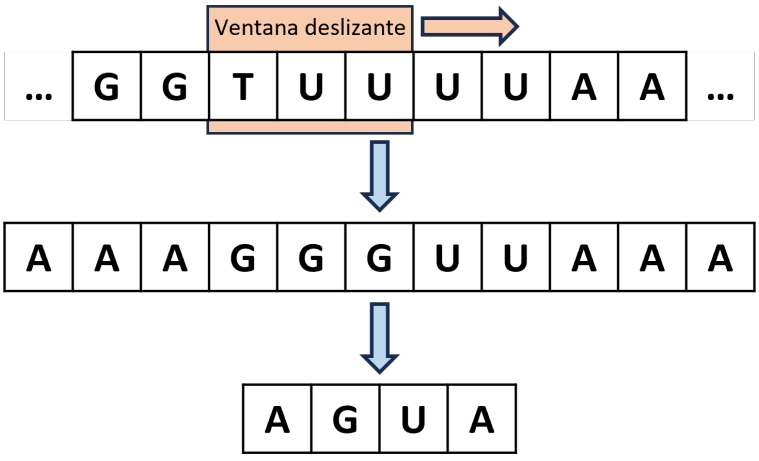


Figura 39. La figura ilustra la metodología aplicada al ejemplo A-G-U-A. En la primera fila, se presentan las predicciones para cada cuadro individual, junto con la aplicación de una ventana deslizante de tamaño 2. La segunda fila muestra el resultado de la ventana deslizante, con la lista resultante de predicciones. La tercera fila muestra el resultado final después de eliminar repeticiones consecutivas.

5.3. Estrategia de evaluación en el dominio continuo

En esta sección, presentamos dos evaluaciones fundamentales para medir el rendimiento en el dominio continuo. En primer lugar, evaluamos la precisión del reconocimiento, lo que nos permite comprender cuán efectivo es el modelo para identificar las señas. En segundo lugar, nos adentramos en la medición de la velocidad de reconocimiento, un aspecto crítico para aplicaciones en tiempo real.

5.3.1. Evaluación de la precisión del reconocimiento en el dominio continuo

Para evaluar la precisión del reconocimiento en el dominio continuo, se empleó una métrica de distancia de Levenshtein normalizada, propuesta en Tashima et al. (2018), que se calcula mediante la fórmula:

$$NLD(s_1, s_2) = \frac{LD(s_1, s_2)}{max\{\lambda(s_1), \lambda(s_2)\}}$$

Donde $LD(s_1, s_2)$ es la distancia de Levenshtein entre las cadenas s_1 y s_2, y la función λ calcula la longitud de cada cadena correspondiente.

5.3.2. Velocidad de reconocimiento

Para medir el rendimiento, se calculó el tiempo promedio de procesamiento por cuadro de video. Esta métrica proporciona una medida directa del tiempo que el sistema necesita para reconocer una seña en cada cuadro de video. Para ello se utilizó la totalidad de las imágenes recopiladas para la evaluación en la modalidad aislada, que ascienden a un total de 279,716.

Además, se determinó la velocidad de procesamiento por segundo, que es una métrica inversa al tiempo de procesamiento por cuadro. Esta métrica indica cuántos cuadros de video el sistema puede procesar en un solo segundo, lo que es esencial para aplicaciones que requieren un reconocimiento en tiempo real.

Para medir la capacidad de reconocimiento en tiempo real del sistema, se midieron tres aspectos clave:

- **Tiempo de inferencia de keypoints (MediaPipe):** Este factor representa el tiempo transcurrido

desde que MediaPipe recibe una imagen como entrada hasta que proporciona un conjunto de keypoints de la mano como salida. Un tiempo de inferencia rápido es esencial para una captura y procesamiento de información visual ágil.

- **Velocidad de predicción del modelo:** Se mide el tiempo que el modelo de reconocimiento requiere para generar una respuesta una vez que ha recibido los keypoints de la mano procesados por MediaPipe. Una velocidad de predicción eficiente es crítica para la identificación de señas en tiempo real.

- **Tiempo total (MediaPipe + modelo):** Se evalúa el tiempo completo que lleva desde la captura de una imagen hasta la generación de una respuesta por parte del modelo de reconocimiento. Un tiempo total bajo es fundamental para garantizar una experiencia en tiempo real.

En resumen, en el contexto de la evaluación continua, se grabaron videos de 20 palabras, siendo deletreadas por 3 participantes inexpertos y 3 semi-expertos. Para realizar esta evaluación en el dominio continuo, se aprovechó el modelo de mejor rendimiento previamente establecido en la modalidad aislada. Esto implicó la implementación de una ventana deslizante y la eliminación de repeticiones para adaptar los modelos de la modalidad aislada a la modalidad continua.

Además, se llevó a cabo una evaluación destinada a medir la velocidad de predicción de los modelos y determinar su capacidad para funcionar en escenarios de tiempo real. En el siguiente capítulo se presentarán tanto los resultados de la metodología presentada en el Capítulo 4 (modalidad aislada) como en este capítulo.

Capítulo 6. Resultados

Este capítulo presenta los resultados de nuestro estudio sobre el reconocimiento continuo de señas estáticas. Comenzamos por analizar la cantidad de imágenes recopiladas y la distribución de datos por clase. Luego, exploramos los resultados en las modalidades aislada y continua. Finalmente, destacamos los principales hallazgos y limitaciones del estudio. Estos resultados son fundamentales para comprender el desempeño de los modelos y orientar futuras investigaciones.

6.1. Características de los datos recabados para la modalidad aislada

Recordando lo mencionado en el Capítulo 4, se recopilaron 21 señas diferentes del alfabeto manual, realizadas de acuerdo a tres grupos de variaciones (A, B y C). Esto se logró con la ayuda de 20 participantes. Los datos de 18 participantes se seleccionaron para la partición de entrenamiento, mientras que 2 participantes conformaron la partición de prueba.

En total se recopilaron 279,716 imágenes, 252,097 (90.13 % de los datos) para la partición de entrenamiento y 27,619 (9.87 % de los datos) para la partición de prueba. Es decir, la partición entrenamiento/prueba de los datos tiene una proporción aproximada 90/10. La cantidad de datos recopilados se detalla en la Tabla 2.

Tabla 2. Descripción de la cantidad total de datos recopilados para el reconocimiento en la modalidad aislada

Grupo de variaciones	Entrenamiento	Prueba	Total
A	83,264	9,439	92,703
B	84,019	8,841	92,860
C	84,814	9,339	94,153
Total	252,097	27,619	279,716

El promedio de imágenes por clase es 13,319.81 $\pm$ 394.94, esto indica que hay una cantidad similar de imágenes en cada clase. En términos generales, en muchos problemas de clasificación, se considera que un conjunto de datos es "balanceado" si no hay una diferencia significativa en la cantidad de ejemplos entre las diferentes clases.

La presencia de datos balanceados contribuye a la disminución del sesgo y el sobreajuste de los modelos, promoviendo una mayor capacidad de generalización. Para mostrar una representación visual de esta

distribución, se incluye en la Figura 40 un gráfico de barras que ilustra la distribución de datos por clase.

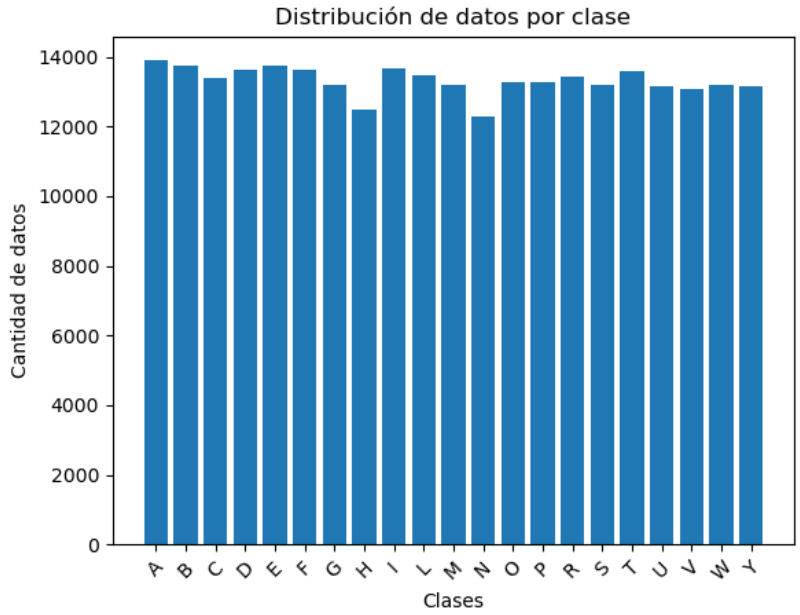


Figura 40. Gráfica de barras con la distribución de datos por clase.

6.2. Resultados en la modalidad aislada

Como recordatorio de lo discutido en la Sección 4.5, nuestra estrategia de evaluación se basó en los distintos grupos de variaciones de datos (A, B y C). Optamos por una estrategia de selección de datos de entrenamiento y prueba basada en combinaciones, que se ajustó según el grupo de variación correspondiente.

Es importante señalar que, en algunos casos, MediaPipe no pudo detectar la mano en la imagen, lo que resultó en la incapacidad de generar keypoints. Esta limitación afectó a tres de las técnicas que utilizan keypoints como datos de entrada, a saber: GPA, SVM y DNN. Dado que la pérdida de datos debida a la falta de detección de mano puede influir en la precisión del reconocimiento, hemos tenido en cuenta esta tasa de pérdida de datos (6.6 %) en nuestras métricas de evaluación.

Teniendo lo anterior en mente, los resultados de la evaluación del reconocimiento de señas estáticas en el dominio aislado se muestran en la Tabla 3. Como se puede observar, se realizaron 10 combinaciones en total. En cada combinación, se resalta el modelo con el mejor rendimiento en amarillo. Ahora, analizaremos en detalle estos resultados.

Tabla 3. Tabla de resultados de la evaluación del reconocimiento de señas estáticas en el dominio aislado. En amarillo se resalta el modelo con mejores resultados para cada uno de los experimentos.

Método	Entrenamiento	Prueba	Accuracy	Precision	Recall	F1-score
GPA	A	A	0.88	0.94	0.88	0.90
SVM	A	A	0.95	1.00	0.95	**0.97**
DNN	A	A	0.95	0.99	0.95	**0.97**
CNN	A	A	0.93	0.94	0.93	0.93
AEM	A	A	0.81	0.84	0.81	0.81
Promedio (σ)			0.90 (0.06)	0.94 (0.06)	0.90 (0.06)	0.91 (0.07)
GPA	A	B	0.77	0.86	0.77	0.80
SVM	A	B	0.87	0.94	0.87	**0.90**
DNN	A	B	0.81	0.88	0.81	0.84
CNN	A	B	0.74	0.77	0.74	0.74
AEM	A	B	0.59	0.67	0.59	0.60
Promedio (σ)			0.76 (0.10)	0.82 (0.10)	0.76 (0.10)	0.78 (0.11)
GPA	A	C	0.55	0.70	0.55	0.60
SVM	A	C	0.68	0.82	0.68	**0.73**
DNN	A	C	0.58	0.70	0.58	0.62
CNN	A	C	0.48	0.57	0.48	0.49
AEM	A	C	0.40	0.50	0.40	0.40
Promedio (σ)			0.54 (0.10)	0.66 (0.12)	0.54 (0.10)	0.57 (0.13)
GPA	B	A	0.86	0.93	0.86	0.88
SVM	B	A	0.95	1.00	0.95	**0.97**
DNN	B	A	0.92	0.97	0.92	0.94
CNN	B	A	0.92	0.94	0.92	0.92
AEM	B	A	0.87	0.89	0.87	0.87
Promedio (σ)			0.90 (0.04)	0.95 (0.04)	0.90 (0.04)	0.92 (0.04)
GPA	B	B	0.76	0.87	0.76	0.79
SVM	B	B	0.90	0.97	0.90	**0.93**
DNN	B	B	0.87	0.94	0.87	0.91
CNN	B	B	0.89	0.90	0.89	0.89
AEM	B	B	0.80	0.83	0.80	0.80
Promedio (σ)			0.84 (0.06)	0.90 (0.05)	0.84 (0.06)	0.87 (0.06)
GPA	B	C	0.55	0.69	0.55	0.60
SVM	B	C	0.64	0.78	0.64	**0.69**
DNN	B	C	0.60	0.72	0.60	0.64
CNN	B	C	0.48	0.57	0.48	0.49
AEM	B	C	0.47	0.52	0.47	0.46
Promedio (σ)			0.55 (0.07)	0.66 (0.11)	0.55 (0.07)	0.58 (0.10)
GPA	C	A	0.82	0.89	0.82	0.83
SVM	C	A	0.95	1.00	0.95	**0.97**
DNN	C	A	0.95	1.00	0.95	**0.97**
CNN	C	A	0.96	0.96	0.96	0.96
AEM	C	A	0.90	0.91	0.90	0.90
Promedio (σ)			0.92 (0.06)	0.95 (0.05)	0.92 (0.06)	**0.93 (0.06)**
GPA	C	B	0.71	0.82	0.71	0.74
SVM	C	B	0.88	0.95	0.88	**0.91**
DNN	C	B	0.84	0.91	0.84	0.87
CNN	C	B	0.75	0.81	0.75	0.75
AEM	C	B	0.83	0.85	0.83	0.83
Promedio (σ)			0.80 (0.07)	0.87 (0.06)	0.80 (0.07)	0.82 (0.07)
GPA	C	C	0.53	0.68	0.53	0.58
SVM	C	C	0.75	0.89	0.75	**0.81**
DNN	C	C	0.70	0.83	0.70	0.75
CNN	C	C	0.65	0.72	0.65	0.66
AEM	C	C	0.47	0.53	0.47	0.48
Promedio (σ)			0.62 (0.12)	0.73 (0.14)	0.62 (0.18)	0.66 (0.13)
GPA	ABC	ABC	0.72	0.82	0.72	0.76
SVM	ABC	ABC	0.87	0.95	0.87	**0.91**
DNN	ABC	ABC	0.82	0.91	0.82	0.86
CNN	ABC	ABC	0.91	0.93	0.91	**0.91**
AEM	ABC	ABC	0.82	0.85	0.82	0.82
Promedio (σ)			0.83 (0.07)	0.89 (0.05)	0.83 (0.07)	0.85 (0.06)

6.2.1. Resultados por combinación

En esta sección analizaremos el rendimiento promedio de todos los modelos en cada una de las combinaciones. Este análisis permitirá discernir las combinaciones más efectivas y ofrecerá recomendaciones fundamentales para futuras aplicaciones y entrenamientos.

Combinación C/A: resultados sobresalientes. En la Tabla 3 se puede apreciar que la combinación que arrojó los mejores resultados en general fue la combinación entrenamiento/prueba C/A (F1-score: $\mu = 0.93$, $\sigma = 0.06$). Este resultado era anticipado, ya que el grupo C contiene imágenes con una alta variabilidad, mientras que el grupo A se compone de datos con una variabilidad baja[1]. Es altamente probable que en el grupo de entrenamiento (C) se incluyan imágenes similares a las presentes en el grupo de prueba (A), lo que explica los buenos resultados obtenidos.

Combinación A/C y B/C: resultados menos favorables. En contraste con el experimento anterior, en este caso, la combinación de entrenamiento/prueba A/C reveló los resultados menos favorables (F1-score: $\mu = 0.57$, $\sigma = 0.13$), seguida de cerca por la combinación B/C (F1-score: $\mu = 0.58$, $\sigma = 0.10$).

El entrenamiento con datos de baja variabilidad tiende a dificultar el reconocimiento de datos con una gran diversidad. La falta de exposición a la variabilidad en los datos de entrenamiento puede resultar en un desempeño inferior al enfrentar datos más diversos en la fase de prueba.

Combinación ABC/ABC: resultados prometedores. Finalmente, en la combinación entrenamiento/prueba ABC/ABC se observan buenos resultados promedio (F1-score: $\mu = 0.85$, $\sigma = 0.06$). Esta observación adquiere relevancia, ya que sugiere que los modelos entrenados con una amplia gama de variaciones deberían ser capaces de desempeñarse eficazmente en una gran diversidad de escenarios.

6.2.2. Análisis de resultados por modelo

Es relevante destacar que las Máquinas de Vectores de Soporte (SVMs) sobresalen como el enfoque de aprendizaje automático que produjo los resultados más prometedores en todas las combinaciones

[1] "Prepararse para lo peor, esperando lo mejor"

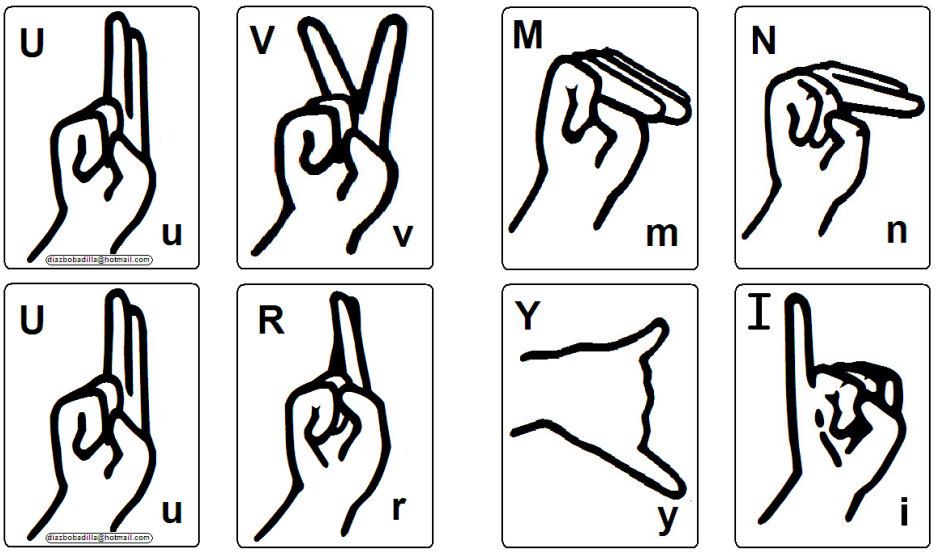
evaluadas. Únicamente se observaron situaciones de empate en cuanto al rendimiento en los casos C/A y ABC/ABC, donde las DNNs y las CNNs, respectivamente, lograron resultados comparables a los obtenidos por las SVMs.

Es notable señalar que tanto las SVMs como las DNNs, dos de las técnicas que arrojaron los mejores resultados, utilizan keypoints como datos de entrada. Sin embargo, esto conlleva una dependencia de la librería de extracción de keypoints, y en ocasiones, se produce una pequeña pérdida de datos cuando la librería no logra reconocer los keypoints. A pesar de ello, estas problemáticas se compensan con la alta calidad y la riqueza de la representación basada en keypoints.

En la Figura 41, se presenta la matriz de confusión correspondiente al modelo SVM, el cual fue entrenado y evaluado utilizando datos de los tres grupos de variaciones (A, B y C), en el contexto del reconocimiento de señas en la modalidad aislada. Esta representación gráfica permite un análisis detallado de las confusiones entre pares de letras generadas por el modelo SVM durante la evaluación. Las matrices de confusión para el resto de modelos se pueden encontrar en el Anexo B.

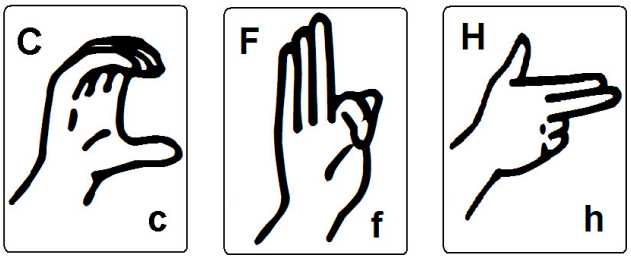
El modelo SVM, entrenado y evaluado utilizando datos de los tres grupos de variaciones, se destaca por las confusiones que genera, las cuales a menudo tienen una justificación visual. En la Figura 41, por ejemplo, se observa que la letra "V" tiende a confundirse con la letra "U" (Tabla 4), lo cual es comprensible al considerar la similitud visual entre ambas letras. Además, se identifican confusiones significativas en otros pares de letras, como "M/N", "U/R" y "Y/I".

Tabla 4. Pares de letras/señas que generan mayor confusión en el modelo SVM, el cual fue entrenado y evaluado utilizando datos de los tres grupos de variaciones.



Por otro lado, el mismo modelo SVM también muestra un rendimiento notable en la clasificación de algunas letras/señas específicas, destacándose por su alta precisión y puntaje F1. Entre las letras mejor clasificadas por este modelo se encuentran la letra "C", con un F1-score de 0.99, seguida de cerca por la letra "F" y "H", ambas con puntuaciones F1 de 0.96 (Tabla 5).

Tabla 5. Letras/señas con el mejor F1-Score en el modelo SVM, el cual fue entrenado y evaluado utilizando datos de los tres grupos de variaciones.



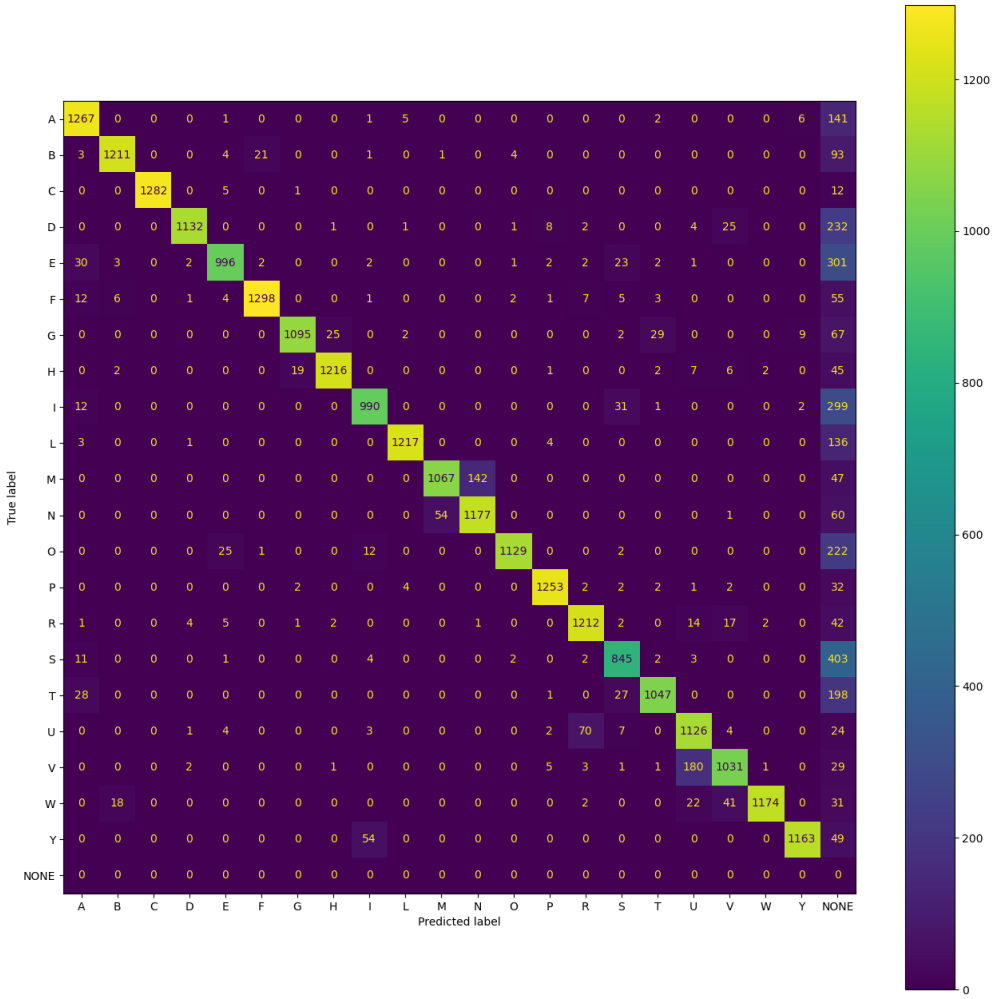


Figura 41. Matriz de confusión para el modelo basado en Máquinas de Soporte Vectorial (SVM), entrenado y evaluado con datos de los tres grupos de variaciones (A, B y C) para el reconocimiento en la modalidad aislada.

Es relevante mencionar que, para los modelos que utilizaron keypoints como datos de entrada, se incluyó una columna adicional etiquetada como "NONE". Esta columna señala aquellos casos en los que MediaPipe no pudo generar keypoints de la mano a partir de la imagen de entrada.

Dado que las SVMs mostraron ser el enfoque más efectivo para el reconocimiento de señas estáticas en el dominio aislado, la próxima etapa de nuestro análisis se enfocará en llevar a cabo una evaluación de este enfoque en dominio continuo.

6.3. Discusión de resultados en la modalidad aislada

Interpretación de los grupos de variaciones. Los distintos grupos de variaciones en nuestro estudio proporcionan valiosas perspectivas sobre la interpretación de los datos. El grupo A, caracterizado por señas con poca variabilidad, podría considerarse como gestos realizados por expertos en la LSM, ya que tienden a ser consistentes y centrados.

En contraste, el grupo B engloba señas capturadas en condiciones de laboratorio, algunas de las cuales pueden ser extremadamente desafiantes y poco representativas de situaciones del mundo real. Por último, el grupo C, que exhibe una alta variabilidad en las imágenes, sugiere la posibilidad de que los datos sean el resultado de personas con un conocimiento limitado en LSM, lo que se refleja en la diversidad y complejidad de los gestos.

Interpretando los resultados, podemos concluir que no es recomendable entrenar modelos solamente con datos de expertos, ya que la capacidad de generalización es reducida. Por otra parte, se recomienda incluir dentro de los datos de entrenamiento señas realizadas por inexpertos, especialmente cuando se contemple el uso de los modelos de reconocimiento de señas en escenarios donde los usuarios puedan ser inexpertos, como en sistemas de enseñanza.

6.4. Resultados en la modalidad continua

La Tabla 7 presenta los resultados obtenidos de la evaluación del modelo entrenado utilizando Máquinas de Soporte Vectorial (SVMs) en la modalidad continua. Este modelo fue entrenado utilizando la totalidad de los datos, los cuales incluyen ejemplos de los tres grupos de variaciones (A, B y C). La evaluación

se realizó a varios niveles: a nivel de palabra, a nivel de participante, a nivel de grupo de participantes (semi-expertos e inexpertos) y nivel general. A lo largo de esta sección, se analizarán en detalle los hallazgos más destacados de este análisis.

Resultados generales. Los resultados generales del reconocimiento de señas en la modalidad continua son prometedores, con una tasa de errores promedio de aproximadamente el 9 %. Esta cifra refleja un desempeño alentador en la capacidad del sistema para interpretar y reconocer de manera precisa las señas estáticas en la modalidad continua.

Diferencias leves en el rendimiento de los participantes. En primer lugar, es relevante resaltar las diferencias en el rendimiento de los participantes, con una ligera distinción entre los participantes 5 y 6, quienes obtuvieron los mejores y peores resultados, respectivamente. Ambos participantes pertenecen al grupo de participantes inexpertos.

Generalización del modelo a distintos niveles de conocimiento en LSM. A pesar de las diferencias a nivel de participante, a nivel general se observa que el promedio por grupo es prácticamente idéntico entre los dos grupos (semi-expertos e inexpertos). Este hallazgo sugiere que el sistema de reconocimiento es capaz de generalizar efectivamente para usuarios con distintos niveles de conocimiento en la LSM, lo que es un indicador positivo de la robustez del modelo.

Errores típicos en el reconocimiento de señas. Con el propósito de analizar y entender los errores cometidos por el sistema, examinamos ejemplos de palabras en las que se presentaron fallas en el reconocimiento. Esto nos permitió identificar errores que se producen con frecuencia.

Entre estos errores se encuentran situaciones en las que el sistema no logra reconocer algunas de las letras en la palabra deletreada, lo que se conoce como un "error por omisión". Por ejemplo, en el caso de la palabra "A-G-U-A" realizada por el participante 4, el sistema no pudo reconocer la letra "G", lo que resultó en una interpretación incorrecta de la palabra como "A-U-A".

También se identificaron situaciones en las que el sistema cometió errores de confusión de letras (error por sustitución), como en el caso de la palabra "L-U-P-A" realizada por el participante 6, que fue reconocida incorrectamente como "L-U-D-A". Estos errores suelen estar relacionados con el rendimiento del modelo utilizado para la clasificación.

Por último, se encontraron casos en los que se detectaron letras adicionales, es decir, movimientos de transición que fueron incorrectamente clasificados como letras (error por inserción). Estos movimientos de transición se observaron al comienzo y al final del video, así como en la mitad del mismo, como en el caso de la palabra "G-A-T-O" realizada por el participante 6, que fue erróneamente reconocida como "G-A-T-O-S".

Estos hallazgos proporcionan una visión detallada de los resultados de la evaluación del modelo SVM y ayudan a comprender las áreas en las que el sistema ha demostrado ser efectivo, así como aquellas en las que se requiere mejora.

6.4.1. Rendimiento del sistema en escenarios de tiempo real

Hardware utilizado. La computadora utilizada en este estudio está equipada con un procesador Intel Core i5-8400 de 6 núcleos a 2.80 GHz, una tarjeta gráfica Nvidia GTX 1660 y 16 GB de RAM. Estas características proporcionaron un rendimiento adecuado para el procesamiento de imágenes y análisis de datos.

Evaluación del modelo en escenarios de tiempo real. En la Tabla 6 se presentan los resultados de la evaluación en tiempo real del sistema de reconocimiento de señas. Los tiempos de inferencia y las velocidades de procesamiento se midieron para MediaPipe, el modelo SVM y la combinación de ambos, lo que proporciona una visión detallada del rendimiento en tiempo real del sistema.

Tabla 6. Tabla de resultados de la evaluación en tiempo real del sistema de reconocimiento de señas. Abreviaturas utilizadas: Media aritmética (Me.), desviación estándar (D.E.), mínimo (Min.), máximo (Max.)

	Tiempo de inferencia (ms)				Cuadros por segundo (FPS)			
Proceso	Me.	D.E.	Min.	Max.	Me.	D.E.	Min.	Max.
MediaPipe	20.47	4.05	17.52	570.81	49.34	4.47	1.75	57.06
Modelo	1.88	0.711	1.42	13.68	582.85	140.99	73.09	701.27
MediaPipe + Modelo	22.37	4.06	19.02	572.29	45.12	3.83	1.75	52.58

Medición del tiempo de inferencia de keypoints. Aunque se destaca el buen rendimiento global de MediaPipe con un tiempo promedio de inferencia por cuadro de aproximadamente 20.47 milisegundos, es importante señalar que existe una desviación estándar ligeramente elevada de 4.05 ms. A pesar de

este detalle, la velocidad promedio de inferencia alcanza los 49.34 Cuadros por segundo (FPS), indicando una eficiente captura de keypoints en términos generales.

Medición de la velocidad de predicción del modelo SVM. El modelo SVM, empleado para el reconocimiento de señas, tiene un tiempo de procesamiento promedio de 1.88 ms, con una desviación estándar de 0.711 ms. Aunque el tiempo promedio sugiere eficiencia, la desviación estándar elevada señala una variabilidad considerable en los tiempos de procesamiento.

Modelo SVM + MediaPipe (proceso combinado). Cuando se combinan MediaPipe y el modelo SVM, el tiempo de procesamiento promedio es de 22.37 ms, con una desviación estándar de 4.06 ms. La velocidad promedio en este proceso combinado es de 45.12 FPS, con una desviación estándar de 3.83. Este valor de proceso combinado nos da una idea del rendimiento del sistema en un escenario de tiempo real.

Las evaluaciones revelan que, si bien se puede lograr un rendimiento sólido al combinar MediaPipe y el modelo SVM, es esencial destacar que todas las mediciones muestran una desviación estándar relativamente alta. Este indicador sugiere que el tiempo de respuesta, tanto de MediaPipe como del modelo SVM, no es consistentemente.

6.5. Discusión de resultados en la modalidad continua

Desempeño del sistema con usuarios inexpertos y semi-expertos. Como se indicó previamente, los participantes que mostraron el rendimiento más destacado y menos destacado en el reconocimiento en modalidad continua pertenecen al grupo de inexpertos. Estas variaciones, aunque mínimas, pueden atribuirse a la diversidad en la ejecución de las señas, ya que algunos participantes pueden realizarlas con mayor deliberación y precisión, mientras que otros pueden enfrentar desafíos al intentar reproducir las señas de manera exacta. Sin embargo, debido a la escasa magnitud de estas diferencias, consideramos que el sistema es capaz de generalizar eficazmente tanto con usuarios inexpertos como con usuarios semi-expertos.

Uso del sistema de reconocimiento de señas en escenarios de tiempo real A juzgar por los promedios en el tiempo de procesamiento, el sistema de reconocimiento de señas utilizando MediaPipe y el

modelo SVM es lo suficientemente rápido para ser utilizado en aplicaciones en tiempo real, destacándose el modelo SVM por su excepcional velocidad de procesamiento.

Sin embargo, es relevante señalar que si bien MediaPipe demostró ser eficiente en este estudio, no todas las bibliotecas de reconocimiento de keypoints ofrecen la misma velocidad de procesamiento. En consecuencia, la metodología empleada aquí depende en gran medida de MediaPipe para garantizar una eficiencia óptima.

Además, es necesario tener en cuenta que el rendimiento de este sistema puede variar según las características de la computadora utilizada. La velocidad y capacidad de procesamiento del hardware en el que se implementa el sistema pueden influir en su desempeño en tiempo real, lo que subraya la importancia de contar con un equipo adecuado para aplicaciones de este tipo.

6.6. Hallazgos y limitaciones

6.6.1. Principales hallazgos

Características de los datos aislados. Se recopilaron datos de 21 señas diferentes del alfabeto manual, agrupadas en tres grupos de variaciones (A, B y C) y se utilizaron imágenes de 20 participantes. La partición de entrenamiento/prueba se dividió en una proporción aproximada 90/10, con un total de 279,716 imágenes recopiladas.

Balance de datos. Se logró un equilibrio en la cantidad de imágenes por clase, con un promedio de imágenes por clase de aproximadamente 13,319.81 $\pm$ 394.94. La presencia de datos balanceados es esencial para reducir el sesgo y el sobreajuste de los modelos, lo que favorece la generalización.

Resultados en la modalidad aislada (por combinación). Se evaluaron diferentes combinaciones de datos de entrenamiento y prueba, y se observó que la combinación C/A obtuvo los mejores resultados (F1-score: 0.93). Esto sugiere que el entrenamiento con datos variados y de expertos es beneficioso para el reconocimiento de señas.

Resultados en la modalidad aislada (por modelo). Las Máquinas de Soporte Vectorial (SVMs) destacaron como el enfoque de aprendizaje automático más efectivo en la mayoría de las combinaciones. También se observaron resultados prometedores con Redes Neuronales Profundas (DNNs). Destaca el hecho que ambas técnicas utilizan keypoints como datos de entrada.

Resultados en la modalidad continua. Se evaluaron los resultados del modelo SVM en la modalidad continua y se encontró que el modelo es capaz de generalizar efectivamente para usuarios con diferentes niveles de conocimiento en la LSM. Además, las métricas de rendimiento indican que el sistema podría funcionar en escenarios de tiempo real.

6.6.2. Principales limitaciones

Pérdida de datos debido a la falta de detección de mano. Se menciona que en algunos casos, el sistema MediaPipe no pudo detectar la mano en la imagen, lo que resultó en la incapacidad de generar keypoints. Esto afectó a las técnicas que utilizan keypoints como datos de entrada y puede influir en la precisión del reconocimiento.

Errores comunes en el reconocimiento de señas. Se identificaron errores comunes, como la omisión de algunas letras en las palabras deletreadas y la confusión entre letras similares. Estos errores son importantes para mejorar el sistema de reconocimiento.

Limitaciones de ciertas combinaciones de datos. Se observó que ciertas combinaciones de datos de entrenamiento/prueba, como A/C y B/C, resultaron en un rendimiento menos favorable. Esto sugiere que el entrenamiento con datos de baja variabilidad puede dificultar el reconocimiento de datos más diversos.

Ausencia de espacios y posiciones de reposo. Los resultados generales del reconocimiento de señas en la modalidad continua son prometedores, con una tasa de errores promedio de aproximadamente el 9 %. No obstante, es crucial señalar que la limitación actual del sistema se centra en que cada video en el conjunto de datos para el reconocimiento continuo presenta solo una palabra deletreada, sin incluir espacios ni posiciones de reposo.

Tabla 7. Tabla de resultados de la evaluación del reconocimiento de señas estáticas en el dominio continuo entrenando con datos del grupo de variaciones ABC.

	Grupo de semi-expertos						Grupo de inexpertos					
	Participante 1		Participante 2		Participante 3		Participante 4		Participante 5		Participante 6	
Ground truth	Predicción	NLD	Predicción	NLD	Predicción	NLD	Predicción	NLD	Predicción	NLD	Predicción	NLD
A-G-U-A	A-G-U-A	0.00	A-G-U-A	0.00	A-G-A	0.25	A-U-A	0.25	A-G-U-A	0.00	A-G-U-A	0.00
B-O-T-E	B-O-T-E-S	0.20	B-O-T-E	0.00	B-O-P-T-E	0.20	B-O-T-E	0.00	B-O-T-E-F	0.20	B-O-E	0.25
C-A-M-A	C-A-M-A	0.00	C-A-M-A	0.00	C-E-M-A	0.25	F-A-M-A	0.25	E-C-A-M-O-A	0.33	C-A-C-A	0.25
D-E-D-O	D-E-D-O	0.00	D-E-D-O	0.00	D-E-D-O	0.00	D-E-D-O	0.00	D-E-D-O	0.00	D-E-D-O	0.00
E-L-E-F-A-N-T-E	E-L-E-F-A-H-T-E	0.12	E-L-E-F-A-N-T-E	0.00	E-L-E-D-F-A-N-P-T-E	0.20	E-L-E-F-A-N-P-T-E	0.11	E-L-E-F-A-N-T-E	0.00	E-L-E-F-A-R-T-E	0.12
F-O-C-A	F-O-C-A	0.00	F-O-E-A	0.25	F-O-C-A	0.00	F-O-A	0.25	F-O-E-A	0.25	F-O-C-A	0.00
G-A-T-O	G-A-T-O-S	0.20	G-A-T-E-O	0.20	G-T-O	0.25	G-A-D-T-O	0.20	G-A-T-O	0.00	G-A-D-T-O	0.20
H-A-C-H-A	H-A-C-H-A	0.00	H-A-E-C-H-A	0.17	H-C-H-A	0.20	H-A-C-H-A	0.00	H-A-E-C-H-A	0.17	H-A-O-C-H-A	0.17
I-S-L-A	I-S-L-A	0.00	I-S-L-A	0.00	I-S-L-A	0.00	I-S-L-A	0.00	I-S-L-A	0.00	I-A-L-A	0.25
L-U-P-A	L-P-A-S	0.50	L-V-U-P-A	0.20	L-U-A-P-A	0.20	A-L-U-P-A	0.20	L-U-P-A	0.00	L-U-D-A	0.25
M-O-N-O	M-O-R-N-O	0.20	M-E-O	0.50	M-O-N-O	0.00	N-M-O-N-O	0.20	M-O-N-O	0.00	M-C-O-H-N-O	0.33
N-U-B-E	N-U-B-E	0.00	N-U-B-E	0.00	N-U-B-E	0.00	N-U-B-E	0.00	N-U-B-E	0.00	N-U-B-E	0.00
O-R-O	O-R-O	0.00	O-R-O	0.00	O-R-O	0.00	O-R-O	0.00	O-R-O	0.00	O-R-O	0.00
P-A-T-O	P-A-T-E-O-E	0.33	P-A-T-O	0.00	P-A-T-O	0.00	P-A-P-T-O-C	0.33	P-A-P-T-O	0.20	P-A-D-O	0.25
R-A-N-A	R-A-N-A	0.00	R-A-N-A	0.00	R-T-N-A	0.25	R-A-N-A	0.00	R-A-N-A	0.00	R-A-H-A	0.25
T-A-C-O	T-A-C-O	0.00	T-A-F-O	0.25	T-A-C-O	0.00	T-A-C-O	0.00	P-T-A-E-O	0.40	T-A-C-O	0.00
U-V-A	U-V-A	0.00	U-V-A	0.00	U-V-A	0.00	U-V-A	0.00	U-V-A	0.00	U-V-A	0.00
V-A-S-O	S-V-A-S-O	0.20	V-A-S-O	0.00	V-A-S-O	0.00	V-A-S-O-C	0.20	V-A-S-O	0.00	U-V-A-S-O	0.20
W-I-F-I	W-I-F-I-S	0.20	W-I-F-I	0.20	W-I-F-I	0.00	B-W-I-F-I	0.20	W-I-O-F-I	0.20	M-W-I-F-I-A	0.33
Y-O-Y-O	Y-O-Y-O	0.00	Y-O-Y-O	0.00	Y-O-Y-O	0.00	Y-O-Y-O	0.00	Y-O-Y-O	0.00	Y-O-Y-O	0.00
Promedio (σ)		0.08 (0.13)		0.08 (0.14)		0.09 (0.11)		0.10 (0.11)		**0.07 (0.12)**		0.12 (0.12)
Promedio por grupo (σ)			0.09 (0.13)						**0.09 (0.12)**			
Promedio general (σ)					0.09 (0.12)							

La ausencia de estos elementos podría afectar la capacidad del sistema para lidiar con situaciones más complejas, donde la presencia de espacios y posiciones de reposo podría aumentar el ruido y afectar el rendimiento del reconocimiento.

Limitaciones del sistema en escenarios de tiempo real. Aunque el sistema presenta un rendimiento para operar en tiempo real, es esencial señalar que esta eficacia no es consistente. Además, aún no se ha sometido a pruebas en equipos con otro tipo de hardware, lo que podría tener implicaciones significativas en su desempeño. Estas consideraciones subrayan la necesidad de evaluar la robustez del sistema en diversas configuraciones de hardware antes de su implementación generalizada.

Estos hallazgos y limitaciones proporcionan una visión detallada de los resultados del estudio y ofrecen una base sólida para futuras investigaciones y mejoras en el reconocimiento de señas estáticas y continuas.

Capítulo 7. Conclusiones

7.1. Conclusiones y discusión

A lo largo de este estudio, se ha observado que el entrenamiento de modelos con datos que abarcan tanto condiciones ideales como escenarios retadores permite que los sistemas de reconocimiento de señas funcionen de manera efectiva con usuarios expertos e inexpertos en la LSM. Este hallazgo se reflejó claramente en los resultados de la evaluación continua, donde el modelo SVM con keypoints como datos de entrada mostró un rendimiento consistente y prometedor en ambas poblaciones. Las capacidades de generalización de estos modelos son cruciales para su aplicabilidad en situaciones del mundo real.

Una de las observaciones clave radica en el rendimiento superior de los modelos que emplean keypoints como datos de entrada en comparación con aquellos que utilizan imágenes. Esta diferencia puede atribuirse a la naturaleza limpia y de alto nivel de los keypoints. En contraste, el entrenamiento de modelos para aprender a reconocer características de alto nivel a partir de imágenes de manos puede resultar un proceso más lento y desafiante. No obstante, es fundamental tener presente que la elección de keypoints como datos de entrada introduce una dependencia con la biblioteca utilizada para su extracción.

Un hallazgo relevante es que el modelo SVM, utilizando keypoints como datos de entrada, se destacó como el más eficaz (F1-Score promedio del 0.91). La aplicación de este modelo combinado con técnicas de ventaneo ha demostrado ser eficaz en el reconocimiento de señas en modalidad continua, logrando una tasa de errores promedio de aproximadamente el 9 %, lo que representa un desempeño alentador. Es importante resaltar que este modelo exhibe un rendimiento consistente tanto con usuarios que son inexpertos en la LSM como con aquellos que poseen conocimiento previo en esta lengua.

Los análisis demostraron que al utilizar MediaPipe para obtener los puntos clave junto (keypoints) con el modelo SVM, se logra un buen desempeño. En conjunto, todos los procedimientos evaluados son lo bastante ágiles, funcionando a una velocidad de 45 cuadros por segundo (45 Cuadros por segundo (FPS)), lo que permite su uso en aplicaciones en tiempo real.

En cuanto a las aplicaciones prácticas de este trabajo, se vislumbran oportunidades significativas, especialmente considerando la eficiencia en escenarios de tiempo real. Un ejemplo concreto de aplicación podría ser en el ámbito educativo, donde este sistema de reconocimiento de señas podría ser utilizado

para enseñar el alfabeto manual a personas interesadas en aprender la LSM. Estas aplicaciones podrían tener un impacto positivo en la promoción de la comunicación inclusiva en la sociedad.

7.2. Limitaciones

Es fundamental reconocer ciertas limitaciones de este trabajo. Por ejemplo, una limitación importante es que el sistema de reconocimiento de señas estáticas no se ha probado con datos de expertos en LSM. Aunque entrenamos el modelo con datos que incluyen diferentes niveles de habilidad en la lengua de señas mexicana, no hemos validado específicamente con palabras deletreadas por expertos, lo que deja una brecha en la evaluación.

Aunque el sistema presenta un rendimiento promedio satisfactorio y demuestra su capacidad para operar en tiempo real, es esencial señalar que, si bien es rápido, este rendimiento no es consistente. Además, el sistema de reconocimiento de señas no se ha probado en diferentes dispositivos, y su rendimiento podría verse influenciado por las características del hardware en el que se implementa. En particular, no se ha diseñado ni implementado específicamente para dispositivos móviles inteligentes.

A pesar de los resultados alentadores en el reconocimiento continuo de señas, es crucial señalar la limitación actual: cada video en el conjunto de datos para el reconocimiento continuo presenta solo una palabra deletreada, sin incluir espacios ni posiciones de reposo. Esta carencia puede afectar la capacidad del sistema para abordar situaciones más complejas, ya que la presencia de espacios y posiciones de reposo podría introducir ruido y alterar la precisión del reconocimiento.

Además, otra de las limitaciones del sistema es que el conjunto de datos recopila imágenes de manos sobre un fondo verde de manera uniforme. A pesar de que este fondo verde se elige por su facilidad de extracción digital y su capacidad para aislar las manos, sería valioso considerar la recopilación de datos con diferentes fondos, especialmente aquellos que representen entornos más realistas y variados.

Por último, cabe destacar que los problemas más frecuentes en el reconocimiento continuo de señas incluyen las confusiones entre letras, en gran medida debido al rendimiento de los clasificadores, así como la detección errónea de señas debido a los movimientos de transición entre señas. Estos desafíos representan áreas clave para futuras mejoras y optimizaciones en los sistemas de reconocimiento de señas.

7.3. Aportaciones

En este estudio, se han logrado contribuciones significativas, destacando la creación de un conjunto de datos que abarca señas estáticas del alfabeto manual de la LSM. Este conjunto de datos se caracteriza por su diversidad y capacidad para poner a prueba la robustez de los modelos de reconocimiento, lo que representa un recurso valioso para la investigación en este campo. El dataset utilizado en esta investigación está disponible para su descarga en el siguiente enlace: `https://zenodo.org/doi/10.5281/zenodo.10067508`.

Otro aporte significativo de este estudio radica en la creación de un modelo de reconocimiento de señas estáticas en LSM utilizando un conjunto de datos diverso y robusto.

7.4. Trabajo futuro

En futuras investigaciones, se propone la creación de un conjunto de datos que incluya personas expertas en la LSM, es decir, individuos cuya lengua nativa sea la LSM, deletreando palabras en esta lengua. La intención es desarrollar un conjunto de datos en un escenario "in the wild", lo que significa que las grabaciones se realizarán en entornos cotidianos y no en un entorno controlado. Este enfoque imitará los conjuntos de datos utilizados en el reconocimiento continuo de señas estáticas en otras lenguas de señas, como la American Sign Language (ASL), permitiendo así una evaluación más realista y robusta del sistema.

Además, se propone explorar el reconocimiento continuo de señas dinámicas. Este enfoque en señas dinámicas permitirá una comunicación más rica y fluida en situaciones de la vida real, donde los usuarios de la LSM a menudo utilizan gestos en movimiento para expresar ideas y conceptos complejos. El desarrollo de un sistema de reconocimiento de señas dinámicas proporcionaría a las personas sordas e hipoacúsicas una herramienta más poderosa y versátil para interactuar con el mundo que les rodea. Asimismo, consideramos que es esencial explorar enfoques orientados a la escalabilidad y aplicaciones en tiempo real en futuros trabajos.

Literatura citada

Borg, M. & Camilleri, K. P. (2020). Phonologically-meaningful subunits for deep learning-based sign language recognition. *European Conference on Computer Vision*, 199–217. https://doi.org/10.1007/978-3-030-66096-3_15.

Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10023–10033. https://doi.org/10.1109/CVPR42600.2020.01004.

Cervantes, J., García-Lamont, F., Rodríguez-Mazahua, L., Rendon, A. Y., & Chau, A. L. (2016). Recognition of mexican sign language from frames in video sequences. *International Conference on Intelligent Computing*, 9772, 353–362. https://doi.org/10.1007/978-3-319-42294-7_31.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297. https://doi.org/10.1007/BF00994018.

Cruz-Aldrete, M. (2008). Gramática de la lengua de señas mexicana. [Tesis doctoral, Centro de Estudios Lingüísticos y Literarios]. Colecciones digitales de El Colegio de México https://repositorio.colmex.mx/concern/theses/kk91fk72t.

del Medico, A. P., Cabodevila, V. G., Vitelleschi, M. S., & Pratta, G. R. (2020). Characterization of tomato generations according to a three-way data analysis. *Bragantia*, 79, 8–18. https://doi.org/10.1590/1678-4499.20190047.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), 141–142. https://doi.org/10.1109/MSP.2012.2211477.

Escobedo Delgado, C. E. (2017). Diccionario de lengua de señas mexicana de la ciudad de méxico. https://pdh.cdmx.gob.mx/storage/app/media/banner/Dic_LSM%202.pdf.

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.

García-Bautista, G., Trujillo-Romero, F., & Caballero-Morales, S. O. (2017). Mexican sign language recognition using kinect and data time warping algorithm. *2017 International conference on electronics, communications and computers (CONIELECOMP)*, 1–5. https://doi.org/10.1109/CONIELECOMP.2017.7891832.

Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, 40, 33–51. https://doi.org/10.1007/BF02291478.

Jurafsky, D. & Martin, J. (2008). *Speech and Language Processing*, (2a ed.). Pearson.

Koller, O., Forster, J., & Ney, H. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141, 108–125. https://doi.org/https://doi.org/10.1016/j.cviu.2015.09.013.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8), 707–710. https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf.

Liddell, S. K. & Johnson, R. E. (1989). American sign language: The phonological base. *Sign language studies*, 64(1), 195–277. https://doi.org/10.1353/sls.1989.0027.

Mahoudeau, F. (2020). Mict-ranet for real-time asl fingerspelling video recognition. https://github.com/fmahoudeau/MiCT-RANet-ASL-FingerSpelling.

Martínez-Guevara, N., Rojano-Cáceres, J.-R., & Curiel, A. (2019). Detection of phonetic units of the mexican sign language. *2019 International Conference on Inclusive Technologies and Education (CONTIE)*, 168–1685. https://doi.org/10.1109/CONTIE49246.2019.00040.

Mejía-Peréz, K., Córdova-Esparza, D.-M., Terven, J., Herrera-Navarro, A.-M., García-Ramírez, 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.

Ocampo, J. C. C., León, M. A. C., Bringas, J. A. S., Encinas, I. D., & Muñoz, J. G. S. (2020). Design of a glove like support for the learning of the mexican sign language. *2020 3rd International Conference of Inclusive Technology and Education (CONTIE)*, 167–172. https://doi.org/10.1109/CONTIE51334.2020.00038.

Pineda, L. A. & Morales, R. (2023). Imagery in the entropic associative memory. *Scientific Reports*, 13(1), 9553. https://doi.org/10.1038/s41598-023-36761-6.

Ramírez Sánchez, J. E., Rodríguez, A. A., & Mendoza, M. G. (2021). Real-time mexican sign language interpretation using cnn and hmm. *Mexican International Conference on Artificial Intelligence*, 55–68. https://doi.org/10.1007/978-3-030-89817-5_4.

Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164, 113794. https://doi.org/https://doi.org/10.1016/j.eswa.2020.113794.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386. https://doi.org/10.1037/h0042519.

Serafín de Fleischmann, M. E. (2014). *Lenguaje Manual: Aprendizaje del Español Signado*. Editorial Trillas.

Serafín & Pérez (2011). *Manos con voz: diccionario de lengua de señas mexicana*. Consejo Nacional para Prevenir la Discriminación. https://www.conapred.org.mx/documentos_cedoc/DiccioSenas_ManosVoz_ACCSS.pdf.

Shi, B., del Rio, A. M., Keane, J., Brentari, D., Shakhnarovich, G., & Livescu, K. (2019). Fingerspelling recognition in the wild with iterative visual attention. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5400–5409. https://doi.org/10.48550/arXiv.1908.10546.

Sosa-Jiménez, C. O., Ríos-Figueroa, H. V., Rechy-Ramírez, E. J., Marin-Hernandez, A., & González-Cosío, A. L. S. (2017). Real-time mexican sign language recognition. *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, 1–6. https://doi.org/10.1109/ROPEC.2017.8261606.

Tashima, K., Aman, H., Amasaki, S., Yokogawa, T., & Kawahara, M. (2018). Fault-prone java method analysis focusing on pair of local variables with confusing names. *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 154–158. https://doi.org/10.1109/SEAA.2018.00033.

Zhou, H., Zhou, W., Zhou, Y., & Li, H. (2020). Spatial-temporal multi-cue network for continuous sign language recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, 13009–13016. https://doi.org/10.1609/aaai.v34i07.7001.

Anexos

Apéndice A. Acrónimos

AEM Memoria Asociativa Entrópica

CNN Red Neuronal Convolucional

DNN Red Neuronal Profunda

FPS Cuadros por segundo

GPA Análisis Procrusteano Generalizado

INEGI Instituto Nacional de Estadística y Geografía

LSM Lengua de Señas Mexicana

OMS Organización Mundial de la Salud

ReLU Rectified Linear Unit

SVM Máquinas de Soporte Vectorial

Apéndice B. Matrices de confusión de la evaluación de reconocimiento de señas estáticas para el dominio aislado

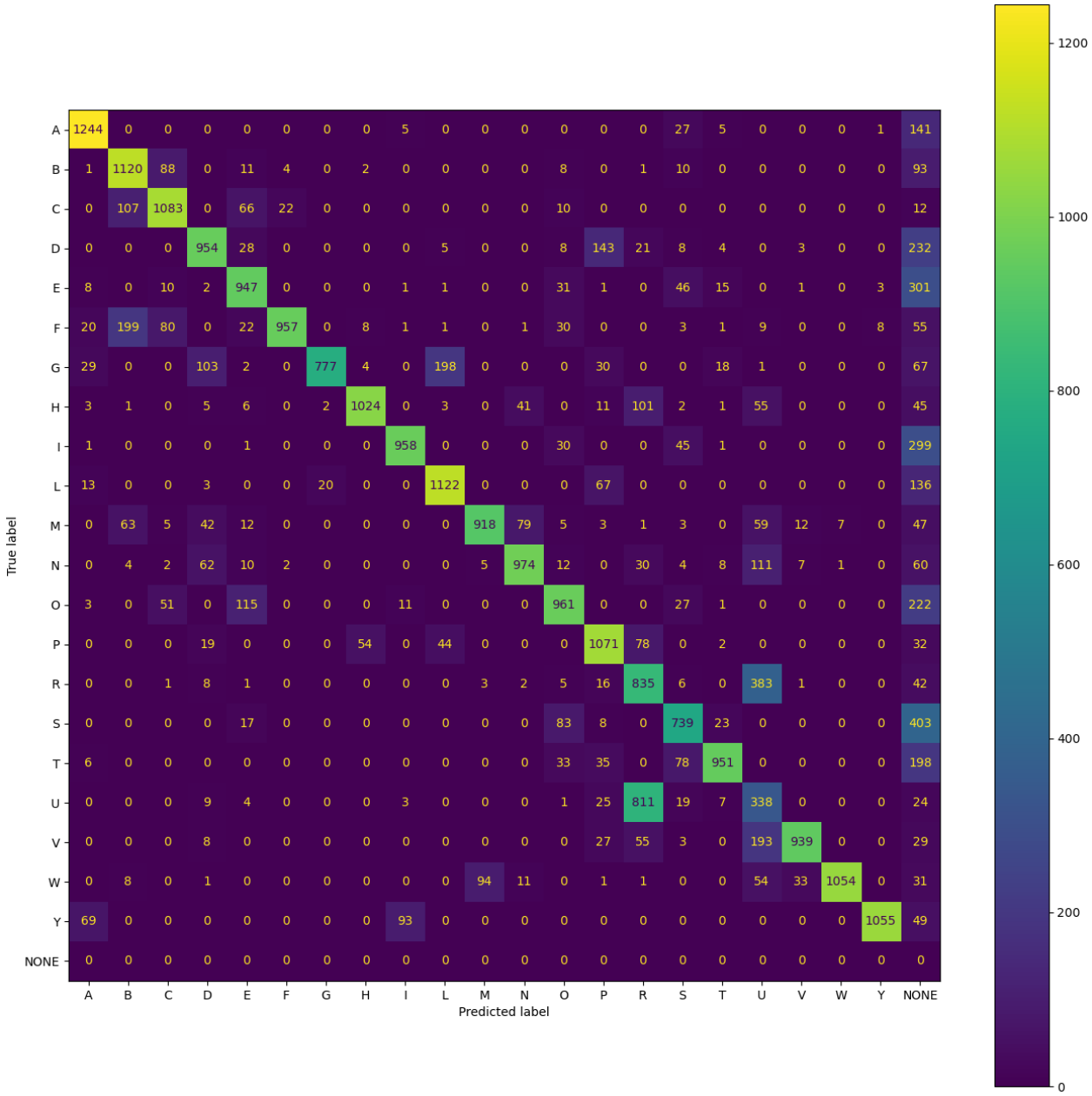


Figura 42. Matriz de confusión para el modelo basado en Análisis Procrusteano Generalizado (GPA), entrenado y evaluado con datos de los tres grupos de variaciones (A, B y C) para el reconocimiento en la modalidad aislada.

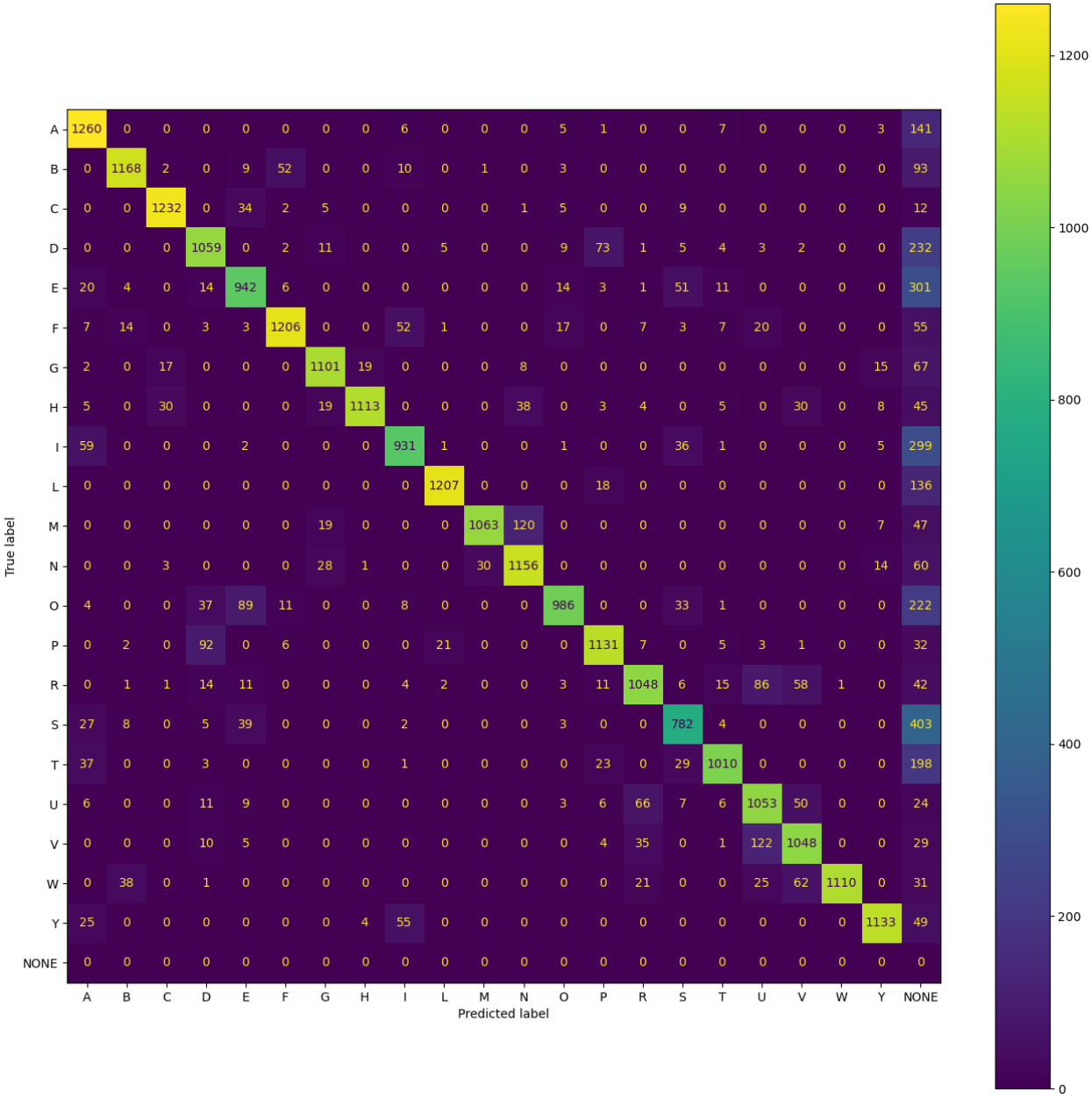


Figura 43. Matriz de confusión para el modelo basado en Red Neuronal Profunda (DNN), entrenado y evaluado con datos de los tres grupos de variaciones (A, B y C) para el reconocimiento en la modalidad aislada.

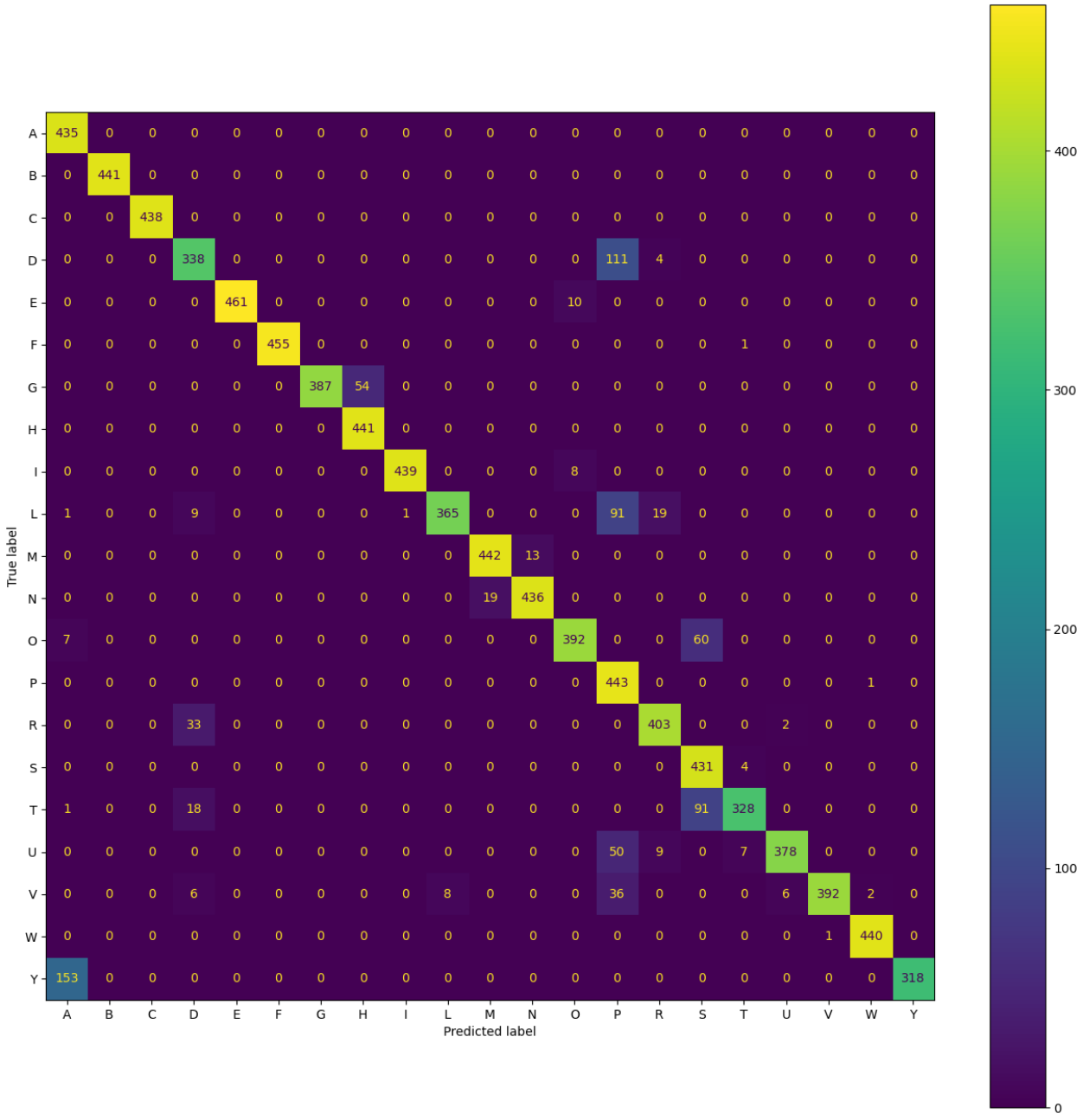


Figura 44. Matriz de confusión para el modelo basado en Red Neuronal Convolucional (CNN), entrenado y evaluado con datos de los tres grupos de variaciones (A, B y C) para el reconocimiento en la modalidad aislada.

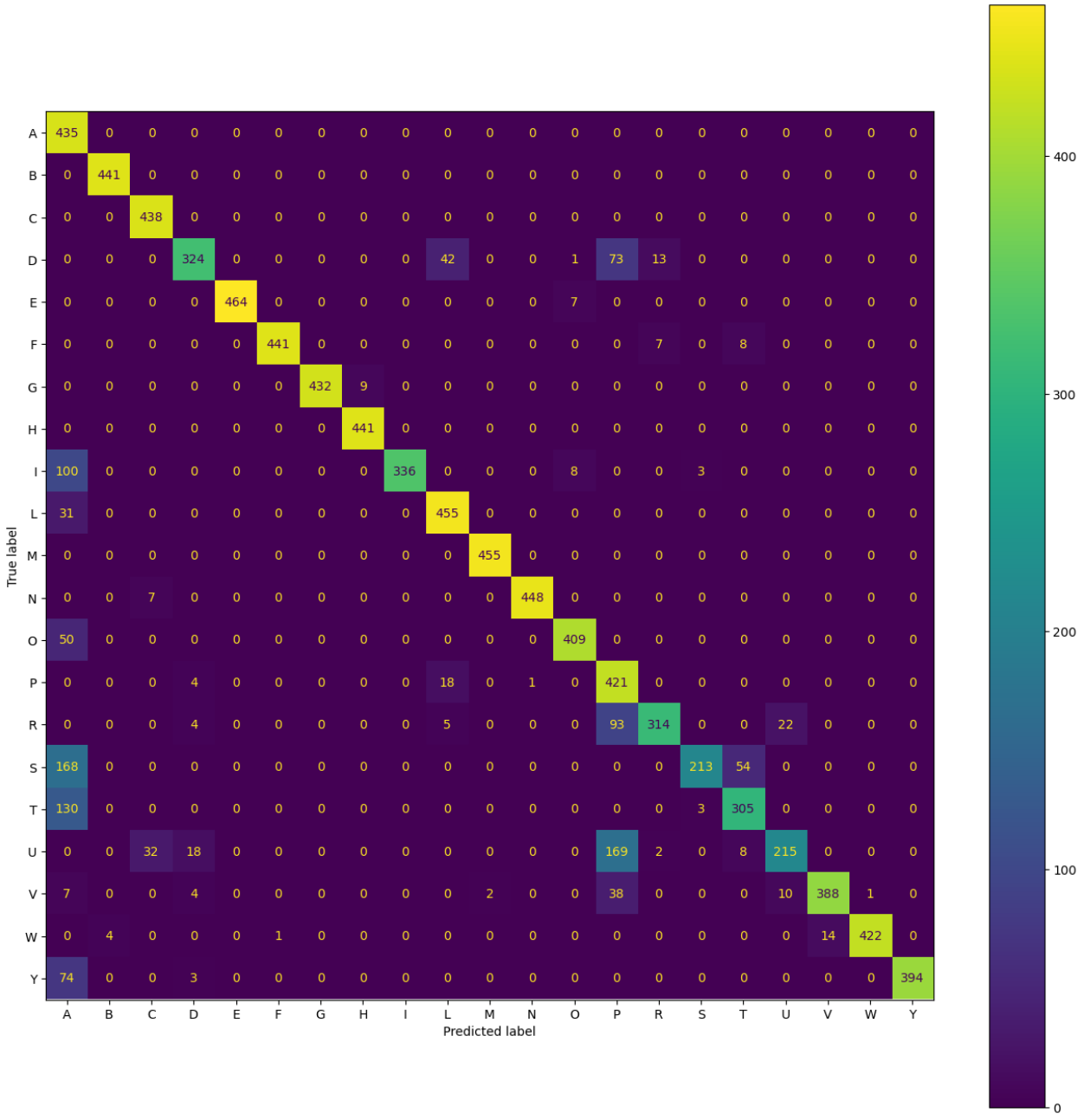


Figura 45. Matriz de confusión para el modelo basado en Memoria Asociativa Entrópica (AEM), entrenado y evaluado con datos de los tres grupos de variaciones (A, B y C) para el reconocimiento en la modalidad aislada.