Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Doctorado en Ingeniería del Lenguaje y del Conocimiento

Reconocimiento de Lengua de Señas con Base a Características Multimodales y Detección de Transiciones entre Señas

Tesis presentada para obtener el grado de
Doctor en Ingeniería del Lenguaje y del Conocimiento

presenta
Daniel Sánchez Ruiz

Director de Tesis
Dr. J. Arturo Olvera López

Co-Director de Tesis
Dr. Ivan Olmos Pineda

Enero, 2024

Agradecimientos

Agradezco al Consejo Nacional de Humanidades, Ciencias y Tecnología (CONAHCyT) por el apoyo otorgado a través de la beca no. 482941 durante los estudios de doctorado. Así como a la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla por brindarme la oportunidad de desarrollar este trabajo de investigación dentro de su programa doctoral.

También agradezco a mis asesores: Dr. José Arturo Olvera López y Dr. Ivan Olmos Pineda quienes con todos sus comentarios expertos, comprensión y apoyo ayudaron que me fuera posible culminar con este proyecto de investigación.

Finalmente, a mi comité revisor: Dra. Soraia Silva Prietch, Dr. Juan Manuel González Calleros y Dr. Manuel Isidro Martin Ortiz, quienes con todo el tiempo que invirtieron y sus comentarios pertinentes ayudaron a hacer más robusta la investigación y además participaron en mi formación como investigador.

Resumen

La capacidad de ser comprendido y transmitir sentimientos, peticiones o ideas a través de las palabras (habladas o escritas) es una de las más infravaloradas por la mayoría de los humanos que tienen el privilegio de hacerlo. La comunidad sorda que no tiene la posibilidad de comunicarse da través del habla enfrenta este desafío todos los días y, aunque las lenguas de señas existen como una forma de luchar contra este problema, no todos en la comunidad sorda saben usarlas; de hecho, la comunidad oyente sabe en menor proporción cómo interpretarlas. Es por ello que el área de reconocimiento de la lengua de señas cobra relevancia como un esfuerzo por solucionar este reto y ayudar a crear nuevos canales de comunicación.

Este trabajo se enfocó en el desarrollo de una metodología para el reconocimiento de lengua de señas a nivel de palabra, como aspectos principales se define un pequeño conjunto de características extraídas a mano, entre ellas, se exploran en profundidad las características no manuales y la utilidad de la identificación de transiciones entre señas. Además, se realizó la aumentación de datos y reducción de dimensionalidad para obtener un espacio de características reducido. Dos modelos de reconocimiento fueron definidos (memoria bidireccional a largo plazo y transformador) ocupando el conjunto de datos LIBRAS y WASL; los mejores resultados fueron de 96,65 % y 87,48 % de precisión, respectivamente.

Los resultados obtenidos en los experimentos que se diseñaron dejaron dilucidar hallazgos bastantes interesantes:

- Quedó claro que con un conjunto de descriptores reducido se pueden obtener resultados competitivos, el generar conjuntos de este tamaño ayudan a requerir menos recursos computacionales para el proceso del entrenamiento así como se obtiene una ganancia en el tiempo requerido para la misma etapa.
- Por otro lado, hay dos puntos claves que salieron a la luz que deben de considerarse, la independencia del señante tienen que considerarse fuertemente pues las variaciones que le pone cada señante a una misma seña pueden afectar cualquier modelo de reconocimiento. Algo que también se notó fue que los datos de tipo continuo siempre van a presentar un fuerte desbalanceo al considerar la información de las transiciones entre señas. Por lo mismo la aumentación de datos y el submuestro de instancias de la clase dominante fueron etapas claves.
- La información de las transiciones entre señas en datos de tipo continuo demostró ser una fuente de información relevante para el proceso de reconocimiento, dentro

del conocimiento de los autores después de hacer una revisión sistemática, esta es una forma de ocupar esos datos que ha sido muy poco usada.

Tabla de Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

De acuerdo con la Organización Mundial de la Salud [1], 430 millones de personas, aproximadamente el 5 % de la población mundial, reportan tener pérdida de audición en algún nivel. Las personas que han perdido la capacidad auditiva de forma considerable a severa son llamados sordos y han tenido muchos problemas de comunicación en el pasado, ya que la principal forma de comunicación que existe en la sociedad moderna es la que se dá de forma hablada. Por esta razón fueron creadas las lenguas de señas, para que la comunidad sorda pudiera comunicar sus ideas, sentimientos o necesidades [2]. Debido a esto se ha ido incrementando los desarrollos tecnológicos para remover los obstáculos que los sordos enfrentan en sus interacciones sociales, principalmente las que tienen con personas que desconocen la lengua de señas.

La lengua de señas es una lengua de expresión gestual, no verbal, que utiliza principalmente las manos como forma de comunicación. Desde el punto de vista del análisis de gestos, se puede considerar a la lengua de señas como gestos dinámicos realizados con las manos pero que involucran más factores. Cualquier lengua de señas consta de cinco componentes principales: movimiento, ubicación, configuración y orientación [3]. Todo esto con respecto a la región de las manos, a estas características se les conocen como manuales. Además, también existen rasgos no involucrados con la región de las manos, los cuales están relacionados en cambio con la postura del cuerpo, el movimiento de boca, la mirada o las expresiones faciales, a estas características se les conoce como no manuales [3].

Los sistemas de reconocimiento de lengua de señas que se han diseñado en recientes años buscan fungir el papel de un moderador entre la comunidad sorda y la comunidad hablante. Las lenguas de señas pueden ser divididas en tres categorías principales de acuerdo a estos sistemas: alfabetos y números, palabras y oraciones [4]. Los alfabetos y números son principalmente gestos estáticos, las palabras son generalmente gestos

dinámicos y las oraciones son una mezcla de gestos estáticos y dinámicos. La mayoría de los sistemas de reconocimiento de señas se han enfocado en las dos primeras categorías con un vocabulario limitado, en condiciones controladas y con dependencia de señantes [1] [5]. El que se adecúa más a situaciones realistas, tales como lo pueden ser conversaciones para comprar algo, para resolver dudas en un salón de clase, para tener conversaciones como al realizar algún trámite de gobierno, entre otras, es el reconocimiento continuo. No obstante, este tipo de reconocimiento es el más complejo pero en el que menos investigación existe [4].

La investigación en el reconocimiento continuo de lengua de señas se ha incrementado en los últimos tiempos, no obstante, los sistemas de reconocimiento están lejos de ser utilizables en cualquier situación [6]. Esto es debido a que aún existen diversos retos a resolver para mejorar los resultados actuales, uno de ellos es la extracción adecuada de características que provean información relevante para poder describir cualquier tipo de seña. Dentro de las características que se han analizado en investigaciones recientes además de las referentes a la zona de las manos, están las relacionadas a la forma de los labios, las expresiones faciales, la postura de la cabeza, entre otras, las cuales son conocidas como características no manuales [6]. Otro gran reto es identificar correctamente el inicio y fin de una seña, sin importar si se trata de un gesto estático o dinámico [7, 8]. Un enfoque que se está explorando para este reto es el de identificar los movimientos de epéntesis, los cuales son aquellos que no pertenecen a ninguna seña [9]. Estos movimientos están más relacionados con un posicionamiento de las manos cuando están en reposo, o al movimiento natural que ocurre entre la transición de una seña a otra.

Existen otros problemas a solucionar como la falta de conjuntos de datos que tengan escenarios representativos de distintas acciones y contextos, más cercanos a situaciones realistas; o la optimización de las metodologías y hardware para funcionar con un buen rendimiento en tiempo real [6, 10]. Este trabajo propone el reconocimiento continuo de lengua de señas basada en la extracción de características multimodales, es decir características manuales y no manuales, además de la detección de transiciones entre señas.

1.1. Objetivos

Se tiene el siguiente objetivo general para esta investigación, así como los subsecuentes objetivos particulares.

[1] persona que gesticula señas pertenecientes a una lengua de señas

1.1.1. Objetivo General

Diseñar una metodología basada en la extracción de características manuales y no manuales así como en la identificación de transiciones entre señas para realizar el reconocimiento continuo de lengua de señas.

1.1.2. Objetivos Particulares

- Identificar limitaciones de los enfoques existentes en la literatura mediante un cuadro comparativo para identificar áreas de oportunidad.
- Diseñar un método que haciendo uso de técnicas de reconocimiento de patrones sea capaz de identificar transiciones entre distintas señas.
- Definir un método que realice la extracción de información relevante con base a las regiones de interés definidas para generar vectores de características.
- Diseñar un método de reconocimiento de patrones que mediante el uso de técnicas de visión por computadora y fusión de características multimodales efectúe el reconocimiento de lengua de señas.
- Realizar el diseño de experimentos para la clasificación de lengua señas para medir el rendimiento de los métodos propuestos.
- Evaluar los resultados obtenidos ocupando métricas definidas en trabajo relacionado a través de tablas y gráficas representativas para realizar una comparación con el estado del arte.

1.2. Preguntas de Investigación

1. ¿Qué mejora de rendimiento se obtiene al hacer uso de características multimodales, así como de la correcta identificación de transiciones entre señas en el reconocimiento de lengua de señas?
2. ¿Cuáles de las características extraídas proveen información relevante para el propósito de distinguir entre señas que son similares?
3. ¿La identificación de transiciones entre señas puede ayudar a delimitar la duración de todas las señas presentes en una conversación compuesta de varias señas?
4. ¿Es posible tener un mejor rendimiento en la tarea del reconocimiento de lengua de señas tomando en cuenta características basadas en la mirada, forma de labios y posición de la cabeza de un señante?

1.3. Hipótesis

La extracción de características multimodales y la identificación de transiciones entre señas proveen información relevante para la tarea del reconocimiento de lengua de señas.

1.4. Limitaciones

En la presente investigación se cuentan con una serie de limitaciones de distinta índole, a continuación se listan:

- Los datos que se ocuparan forman parte de la base de datos de LIBRAS [11], los cuales se encuentran disponibles de forma pública, por lo cual no se hará la captura de datos nuevos, ni se hará procesamiento de datos en tiempo real.
- Al hacer uso de estos datos se tienen las limitaciones que vienen de forma inherente con ellos como: la distancia que hay del señante hacia la cámara, la calidad de los vídeos, el vocabulario presente en las conversaciones y las características físicas de los usuarios (tono de piel, número de dedos, complexión física).
- La investigación sólo comprende el reconocimiento de lengua de señas, además, sólo lo hace a nivel de palabra aunque se ocupan datos continuos. Por último no comprende el apartado de traducción o generación.

Capítulo 2

Marco Teórico

En este capítulo se presentan los conceptos fundamentales que son utilizados en el documento y para el desarrollo de la propuesta metodológica, tales como: la pérdida de la audición, las lenguas de señas, el procesamiento de lengua de señas y conceptos relacionados con la extracción de las características y la etapa del reconocimiento.

2.1. Pérdida de Audición

Más del 5 % de la población mundial (430 millones de personas) padece una pérdida de audición discapacitante y requiere rehabilitación (432 millones de adultos y 34 millones de niños). Se calcula que en 2050 esa cifra superará los 700 millones (una de cada diez personas) [1].

La pérdida de audición discapacitante se refiere a una merma superior a 35 decibelios (dB) en el oído de una persona. Casi el 80 % de las personas con este problema viven en países de ingresos bajos y medianos. La prevalencia de la pérdida de audición aumenta con la edad: entre los mayores de 60 años, más del 25 % padece una pérdida de audición discapacitante [1].

Se considera que alguien que tiene pérdida de audición cuando no es capaz de oír tan bien como una persona cuyo sentido del oído es normal, es decir, cuyo umbral de audición en ambos oídos es igual o mejor que 20 dB [1]. La pérdida de audición puede ser leve, moderada, grave o profunda. Puede afectar a uno o ambos oídos y provocar dificultades para oír una conversación o sonidos fuertes.

Las personas *duras de oído* son personas cuya pérdida de audición es entre leve y grave. Por lo general se comunican mediante la palabra y pueden utilizar como ayuda audífonos, implantes cocleares y otros dispositivos, así como los subtítulos [1].

2.1.1. Consecuencias de la pérdida de audición no tratada

Cuando no se trata, la pérdida de audición puede llegar a afectar algunos aspectos de la vida de una persona [12], como lo podrían ser:

- Comunicación y habla
- Cognición
- Educación y empleo: en los países en desarrollo, los niños con pérdida de audición y sordera rara vez son escolarizados. Asimismo, entre los adultos con pérdida de audición la tasa de desempleo es mucho más alta. Entre los que tienen un trabajo, el porcentaje de personas con pérdida de audición que ocupan puestos en las categorías más bajas es mayor que la media general de la fuerza de trabajo.
- Aislamiento social, soledad y estigma
- Consecuencias en la sociedad y la economía
- Años perdidos por discapacidad (APD) y años de vida ajustados en función de la discapacidad (AVAD)

La Organización Mundial de la Salud (OMS) calcula que los casos desatendidos de pérdida de audición representan un coste mundial anual de 980 000 millones de dólares. Dicha cifra incluye los costes del sector sanitario (excluyendo el coste de los dispositivos de ayuda a la audición), los costes del apoyo educativo, la pérdida de productividad y los costes sociales [1, 13]. Más del 57 % de esos costes se producen en países de ingresos bajos y medianos.

2.2. Lengua de Señas

Las personas de la comunidad *sorda* tienen una pérdida de audición en algún nivel, lo que significa que oyen con dificultades o nada. Una de las formas empleadas por los miembros de esta comunidad para comunicarse con personas hablantes o sordas es mediante alguna lengua de señas.

La lengua de señas es una lengua de expresión gestual, no verbal, que utiliza principalmente las manos como forma de comunicación [14–16]. Una particularidad interesante de la lengua de señas es que al igual que otras existentes, su alcance es regional; es decir, cada país e incluso regiones más pequeñas dentro del mismo tienen su propia lengua, lo

que significa que existen variaciones o modismos en la lengua base de un país en cada una de estas zonas.

Como se mencionó previamente, cualquier lengua de señas consta de cinco componentes principales [14, 15]: movimiento, ubicación, configuración, orientación de la palma de la mano y además rasgos no manuales. A pesar de todos estos componentes la mayoría de los sistemas computacionales sólo utilizan los primeros tres [4–6, 14, 15, 17].

Los movimientos generalmente son cortos y precisos, suelen ser lineales, con forma circular o vibrantes entre otros. La posición es otro elemento importante en la morfología de la seña. Generalmente, se considera la posición de las manos en relación con otra parte del cuerpo como suele ser la cabeza, ojos, pecho, hombros, abdomen, etc. [14, 18]. Por otro lado, el tercer componente, la configuración de las manos es un elemento también clave. Esto es la postura que los dedos de cada mano tienen al realizar el gesto. En una seña, cada mano podría tener configuraciones diferentes y cada configuración podría cambiar durante la ejecución.

Las señas suelen describirse con una configuración inicial y una configuración final (en cada mano) [19]. La Fig. 2.1 muestra de forma parcial algunas de las configuraciones que pueden tomar las manos en la lengua de señas argentina. Como toda lengua viva, esta información va cambiando con los años y en ocasiones los diccionarios y documentación existente debe ir actualizándose con nuevas señas, configuraciones, etc.

Si bien existe un sistema de configuraciones en cada lengua de señas, el cual es bastante amplio, también existe lo que se conoce como diccionario dactilológico [19]. En este tipo de diccionarios se especifica una seña particular para cada letra del alfabeto hablado, suelen ser unimanuales y generalmente se utilizan para traducir nombres propios y enseñar a las personas sordas el lenguaje verbal articulado de la región. Generalmente, las letras del abecedario suelen ser estáticas, con algunas excepciones que presentan pequeños movimientos.

Los signos pueden describirse a nivel de subunidad mediante fonemas. Estos codifican diferentes elementos de un signo. A diferencia del habla, no tienen por qué aparecer de forma secuencial, sino que pueden combinarse en paralelo para describir una seña [21].

FIGURA 2.1: Ejemplos de configuraciones tomadas por las manos en la lengua de señas argentina. Imagen tomada de [20].

2.2.1. Tipos de Descriptores

Las señas se transmiten a través de diferentes canales; el movimiento realizado por las manos, el lugar en el que se realiza la seña, las formas de las manos, la disposición relativa de las manos y, finalmente, la orientación tanto de las manos como de los dedos para explicar el plano en el que se sitúan las manos [21].

A continuación, se describe un pequeño subconjunto de los descriptores que componen las lenguas de señas. Si bien no se detalla toda la estructura que una lengua puede tener, se describen los principales descriptores que suponen un reto en el reconocimiento de lengua de señas de acuerdo con [21]:

- Adverbios que modifican verbos: como ejemplo, un señante no usara dos señas para decir *correr rápidamente*, sino que modificaran el signo de correr para denotar la aceleración.
- Rasgos no manuales: las expresiones faciales y la postura corporal son clave para determinar el significado de las frases; por ejemplo, la posición de las cejas puede determinar el tipo de pregunta. Algunas señas se distinguen sólo por la forma de los labios, ya que pueden compartir una seña manual común.
- Colocación: los pronombres como él, ella o eso no tienen su propio seña, sino que se describe al objeto directo y se le asigna una posición en el espacio de señas. Las

referencias futuras señalan la posición, y las relaciones pueden describirse señalando más de un objeto.

- Verbos direccionales: ocurren entre el señante y el/los destinatario/s, la dirección del movimiento indica la dirección del verbo. Buenos ejemplos de verbos direccionales son *dar* y *llamar*. La dirección del verbo transmite implícitamente qué sustantivos son el sujeto y el objeto.
- Señas posicionales: cuando un signo actúa sobre la parte del cuerpo de forma descriptiva, por ejemplo, moretón o tatuaje.
- Desplazamiento del cuerpo: representado por el giro de los hombros y la mirada, a menudo utilizado para indicar el cambio de rol al relatar un diálogo.
- Iconicidad; cuando una seña imita la cosa que representa, puede ser alterado para dar una representación apropiada. Por ejemplo, la seña para salir de la cama puede ser alterado entre saltar de la cama con energía a un recostado que se resiste a levantarse.

2.3. Modelado de Lengua de Señas

Una lengua de señas se expresa principalmente a través de las manos, pero las expresiones faciales y los movimientos de todo el cuerpo también desempeñan un papel importante, sobre todo para las funciones gramaticales [21]. La mayoría de las señas se realizan con una mano, pero también hay muchas que hacen uso de las dos manos. A menudo es importante distinguir entre las dos manos de un señante. Por tal motivo, es generalmente definida la mano fuerte para designar a la mano que realiza las señas con una mano y el componente principal de las señas de dos manos. La mano débil es la opuesta a la mano fuerte [10]. En el caso de las personas diestras, la mano fuerte es la mano derecha de la persona, y la mano débil es la mano izquierda de la persona; para las personas zurdas es de forma inversa.

La lengua de señas es altamente influenciable; es decir, muchas señas pueden ser modificadas según alguna función gramatical, como el número, la concordancia sujeto-verbo y la concordancia verbo-objeto. También pueden modificarse para indicar el aspecto (por ejemplo, rápido, lento), la repetición y la duración [22].

Un fonema se define como la unidad contrastante más pequeña de una lengua [23]; es decir, la unidad más pequeña que puede distinguir unos morfemas [1] de otros. En lengua

[1] unidades de significado

de señas, los equivalentes de los fonemas en las lenguas habladas son las diversas formas de las manos, ubicaciones, orientaciones y movimientos.

2.3.1. Modelo de Transiciones

S. Liddell y R. Johnson enfatizaron las secuencias de segmentos de fonemas. Estos modelos se denominan modelos de segmento. Describen dos clases principales de segmentos en su modelo Movement-Hold, que denominan movimientos (M) y retenciones (H). Los movimientos se definen como aquellos segmentos durante los cuales cambia algún aspecto de la configuración de la seña, como un cambio en la forma de la mano, un movimiento de la mano o un cambio en la orientación de la mano. Las retenciones se definen como aquellos segmentos durante los cuales todos los aspectos de la configuración de la seña permanecen inalterados; es decir, las manos permanecen inmóviles durante un breve período de tiempo [21, 24].

Las señas se componen de secuencias de movimientos y retenciones. Algunas secuencias comunes son HMH (una retención seguida de un movimiento seguido de otra retención), MH (un movimiento seguido de una retención), y MMMH (tres movimientos seguidos de una retención). Los segmentos de movimiento tienen características que describen el tipo de movimiento (recto, redondo, en ángulo agudo, etc.), así como el plano y la intensidad del movimiento. Además, a cada segmento le corresponde un conjunto de características articulatorias que describen la configuración de la mano, la orientación, la ubicación y los movimientos locales [21, 24].

Liddell y Johnson también hablan de las transiciones entre dos signos cualesquiera, las consideran como un proceso fonológico en [24]. Un proceso fonológico cambia la apariencia de un enunciado a través de reglas bien definidas en la fonología, pero no cambia el significado del enunciado; a este proceso concreto lo llaman movimiento de epéntesis. Consiste en la inserción de movimientos adicionales entre dos signos adyacentes, y está causado por las características físicas de las lenguas de señas.

Los movimientos de epéntesis plantean un problema en el reconocimiento de lengua de señas, porque los movimientos son demasiado largos y visibles para ser ignorados, y por tanto tienen un efecto significativo en la precisión del reconocimiento.

2.3.2. Forma de Mano

La forma de la mano es una de las partes más importantes del reconocimiento de lengua de señas, no sólo para distinguir unos señas de otros a través del léxico, sino también en el papel de los clasificadores. Se trata de señas que representan una clase de objetos de forma colectiva al tiempo que trazan un recorrido de dicho objeto por el espacio [3].

La forma de la mano en los clasificadores es tan importante porque constituye el elemento distintivo de la seña, y en muchas situaciones es el único aspecto invariable de la configuración de la seña. Todos los demás aspectos como la orientación, la posición y el movimiento son de forma libre y cambian según lo que exprese el señante [3].

Por lo tanto, el reconocimiento de la forma de la mano es de vital importancia en un sistema de reconocimiento completo; no sólo para hacer más robusto el reconocimiento de las señas léxicas de una lengua de señas, sino también para dar el primer paso hacia la construcción de un puente hacia los sistemas de reconocimiento de gestos [15, 17].

2.4. Reconocimiento de Lengua de Señas

Para la etapa del reconocimiento de lengua de señas en esta investigación es clave definir regiones de interés, la forma en la qué se hará la extracción de características así como los métodos para la etapa de clasificación. A continuación se detalla la teoría de las técnicas ocupadas para todas estas tareas.

2.4.1. Métodos de Detección de Regiones de Interés

Una de las primeras etapa a considerar en cualquier metodología de sistemas de reconocimiento continuo de lengua de señas, es el de la detección y segmentación de regiones de interés. Generalmente, a través de cuadros envolventes se suelen identificar estas regiones, una vez que se obtienen estos, la segmentación es directa. A continuación se describen los métodos empleados en la investigación.

2.4.1.1. YOLOv5

YOLO (*You Only Look Once*) [25] es una red neuronal extremadamente rápida que detecta múltiples clases de objetos a la vez a la asombrosa velocidad de 45 fotogramas

por segundo (YOLO básico) a 155 fotogramas por segundo (YOLO rápido). A modo de comparación, la mayoría de las cámaras de los teléfonos móviles capturan vídeos a unos 30 fotogramas por segundo, mientras que cámaras de mayor velocidad lo hacen a unos 250 fotogramas por segundo.

Comparado con el tiempo que tarda el cerebro humano en detectar imágenes en unos 13 milisegundos, YOLO reconoce las imágenes al instante de forma similar a como lo hacen los humanos [26]. Así, proporciona a las máquinas una capacidad de detección de objetos instantánea.

El mecanismo de detección de YOLO se basa en una única red neuronal convolucional (CNN) que predice simultáneamente múltiples cuadros delimitadores de objetos y la probabilidad de detección de una determinada clase de objeto en cada cuadro delimitador. Las imágenes de la Fig. 2.2 ilustran esta metodología.

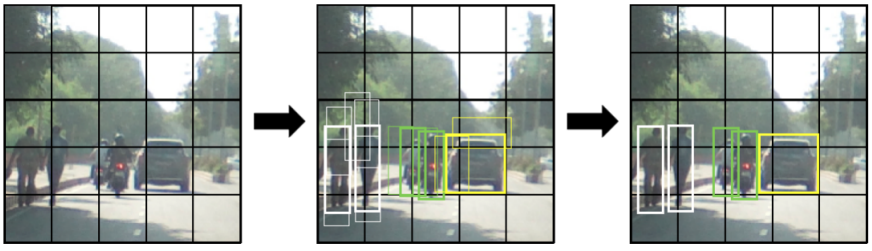


FIGURA 2.2: Metodología seguida en YOLO, imagen tomada de [25].

De acuerdo a la sintesis de Kar [26], las fotos anteriores muestran los tres pasos principales, desde el desarrollo de los cuadros delimitadores hasta el uso de la supresión no máxima y los cuadros delimitadores finales. Los pasos descritos por Redmon et al. [25] en su trabajo son los siguientes:

1. La CNN de YOLO utiliza características de toda la imagen para predecir cada cuadro delimitador. Así, la predicción es global, en lugar de local.

2. La imagen completa se divide en celdas de cuadrícula S x S y cada celda de cuadrícula predice B cuadros delimitadores y la probabilidad (P) de que el cuadro delimitador contenga un objeto. Así, hay un total de S x S x B cuadros delimitadores, con las correspondientes probabilidades para cada cuadro delimitador.

3. Cada caja delimitadora contiene cinco predicciones (x, y, w, h y c), donde se aplica lo siguiente:

- o(x, y) es la coordenada del centro de la caja delimitadora, relativa a la coordenada de la celda de la cuadrícula.
- o(w, h) es la anchura y la altura del cuadro delimitador, en relación con la dimensión de la imagen.
- o(c) es la predicción de confianza, que representa el IOU entre la caja predicha y la caja de verdad.

4. La probabilidad de que una celda de la cuadrícula contenga un objeto se define como la probabilidad de la clase multiplicada por el valor intersección sobre la unión (IoU). Esto significa que si una celda de la cuadrícula sólo contiene parcialmente un objeto, su probabilidad será baja y el valor IoU seguirá siendo bajo. Esto tendrá dos efectos en el cuadro delimitador de esa celda de la cuadrícula:

   - La forma del cuadro delimitador será menor que el tamaño del cuadro delimitador de una celda de la cuadrícula que incluya completamente el objeto, porque la celda de la cuadrícula sólo ve una parte del objeto e infiere su forma a partir de ella. Si la celda de la cuadrícula contiene una parte muy pequeña de un objeto, es posible que no reconozca el objeto en absoluto.
   - El nivel de confianza de la clase del cuadro delimitador será bajo porque el valor IoU resultante de la imagen parcial no se ajustará a la predicción de la verdad del terreno.

5. En general, cada celda de la cuadrícula sólo puede contener una clase, pero utilizando un principio de caja de anclaje se pueden asignar varias clases a una celda de la cuadrícula. Una caja de anclaje es una forma predefinida que representa la forma de las clases detectadas. Por ejemplo, si detectamos tres clases -coche, motocicleta y persona-, probablemente probablemente podamos arreglárnoslas con dos formas de caja de anclaje: una que represente la una que represente la motocicleta y el ser humano, y otra que represente el coche. Esto puede confirmarse observando la imagen más a la derecha de las imágenes anteriores. Podemos determinar la forma de la caja de anclaje para formar los datos CSV de entrenamiento analizando la forma de cada clase utilizando algoritmos como la agrupación de k-medias.

2.4.2. Métodos para la Extracción Características

Una vez que se tienen identificadas y segmentadas las regiones de interés, el siguiente paso es hacer la extracción de características relevantes. En dicho apartado existen varios enfoques para llevar a cabo la tarea, desde los que lo hacen con base en la región de las manos, de la cara o de la estimación de puntos claves referentes a la postura del cuerpo.

A continuación, se listan los métodos empleados en esta tarea para la investigación.

2.4.2.1. MediaPipe

MediaPipe [27] es un marco para crear canales de aprendizaje automático para procesar datos de series temporales como video, audio, etc. Este marco multiplataforma funciona en escritorio/servidor, Android, iOS y dispositivos integrados como Raspberry Pi y Jetson Nano.

En el trabajo del framework MediaPipe [27] se especifica que el mismo consta de tres elementos principales:

- Un marco para la inferencia a partir de datos sensoriales (audio o vídeo)
- Un conjunto de herramientas para la evaluación del desempeño.
- Componentes reutilizables para inferencia y procesamiento (calculadoras)

MediaPipe permite a un desarrollador crear un prototipo de canalización de forma incremental. Una canalización de visión se define como un gráfico dirigido de componentes, donde cada componente es un nodo (Çalculadora"). En el gráfico, las calculadoras están conectadas mediante "flujos" de datos. Cada flujo representa una serie temporal de "paquetes"de datos [27].

Juntos, las calculadoras y los flujos definen un gráfico de flujo de datos. Los paquetes que fluyen a través del gráfico se clasifican según sus marcas de tiempo dentro de la serie temporal. Cada flujo de entrada mantiene su propia cola para permitir que el nodo receptor consuma los paquetes a su propio ritmo. La canalización se puede refinar de forma incremental insertando o reemplazando calculadoras en cualquier parte del gráfico [27].

MediaPipe brinda acceso a una amplia variedad de potentes modelos de aprendizaje automático creados teniendo en cuenta las limitaciones de hardware de los dispositivos móviles. Los modelos que se incluyen incluyen los siguientes:

1. Modelos anatómicos
2. Seguimiento manual
3. Seguimiento de posturas

4. Seguimiento de malla facial
5. Seguimiento holístico (más de 3 combinados)
6. Modelos de segmentación
7. Segmentación de selfies
8. Segmentación del cabello
9. Detección/seguimiento de objetos 2D
10. Detección de objetos 3D y estimación de pose

Algunas de las soluciones que recién se acaban de mencionar se pueden visualizar en la Fig. 2.3

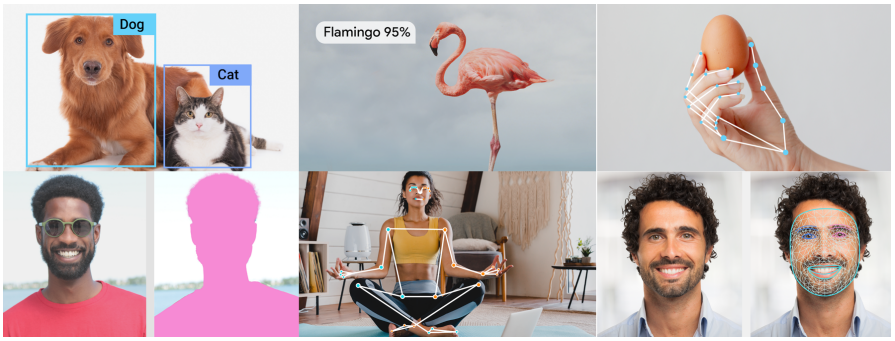


FIGURA 2.3: Soluciones presente en el framework MediaPipe, imagen tomada de [27].

2.4.2.2. Redes Dinámicas Bayesianas

Las redes bayesianas son un tipo de modelo gráfico probabilístico que utiliza la inferencia bayesiana para los cálculos de probabilidad. Estas redes buscan modelar la dependencia condicional y por tanto la causalidad, representando la dependencia condicional mediante aristas en un grafo acíclico dirigido (DAG). Un DAG es un grafo con enlaces dirigidos y que no contiene ciclos dirigidos. [28].

En las redes bayesianas, cada nodo representa una variable, como la altura, la edad o el sexo de alguien. Una variable puede ser discreta, como Género = {Hombre, Mujer} o puede ser continua, como la edad de alguien.

Los enlaces se añaden entre nodos para indicar que un nodo influye directamente en el otro. Cuando no existe un enlace entre dos nodos esto no significa que sean completamente independientes, ya que pueden estar conectados a través de otros nodos. Sin embargo, pueden ser dependientes o independientes en función de las pruebas que se establezcan en otros nodos [29].

Las redes dinámicas bayesianas son una extensión de las redes bayesianas, las cuales son capaces de soportar diversos estados de un mismo problema, es decir consideran la temporalidad [28].

Las redes bayesianas son uno de los formalismos más completos y consistentes para la adquisición y representación de conocimiento y para el razonamiento a partir de datos incompletos y/o inciertos. Estas redes son un tipo de modelo gráfico probabilístico que utiliza la inferencia bayesiana para los cálculos de probabilidad. Estas redes buscan modelar la dependencia condicional y por tanto la causalidad, representando la dependencia condicional mediante aristas en un grafo acíclico dirigido (DAG). Un DAG es un grafo con enlaces dirigidos y que no contiene ciclos dirigidos [28].

En las redes bayesianas, cada nodo representa una variable, como la altura, la edad o el sexo de alguien. Una variable puede ser discreta, como Género = Hombre, Mujer o puede ser continua, como la edad de alguien. Los enlaces se añaden entre nodos para indicar que un nodo influye directamente en el otro. Cuando no existe un enlace entre dos nodos esto no significa que sean completamente independientes, ya que pueden estar conectados a través de otros nodos. Sin embargo, pueden ser dependientes o independientes en función de las pruebas que se establezcan en otros nodos [29].

Las redes dinámicas bayesianas (RDB) son una extensión de las redes bayesianas, las cuales son capaces de soportar diversos estados de un mismo problema, es decir consideran la temporalidad [28]. Las RDB son una clase de modelo general y flexible para representar procesos estocásticos complejos y se utilizan en varias áreas, como reconocimiento de voz, seguimiento e identificación de objetivos o genética [28]. En particular, estas redes han sido empleadas en problemas relacionados como: reconocimiento de gestos [30], reconocimiento de discurso [31] o en trabajos más recientes como el reconocimiento de lengua de señas americana [32].

2.4.3. Métodos de Clasificación

Para la última etapa se debe de considerar cuál será el mejor método de clasificación a utilizar. Como se está tratando de un problema que se compone de varios estados a lo largo del tiempo, algunos de los principales métodos estarán relacionados con modelos ocultos de markov o redes recurrentes, a continuación se exploran en detalle.

2.4.3.1. LTSM

Una red LSTM (*Long Short-Term Memory*) es un tipo particular de red neuronal recurrente (RNN). Las RNN contienen una capa (o célula) recurrente que es capaz de manejar datos secuenciales haciendo que su propia salida en un determinado paso de tiempo forme parte de la entrada del siguiente paso de tiempo, de modo que la información del pasado puede afectar a la predicción en el paso de tiempo actual. Se le llama red LSTM cuando se refiere a una red neuronal con una capa recurrente LSTM [33].

Cuando se introdujeron por primera vez las RNN, las capas recurrentes eran muy sencillas y consistían únicamente en un operador *tanh* que garantizaba que la información transmitida entre los pasos de tiempo se escalaba entre -1 y 1. Sin embargo, se demostró que se presenta el problema del gradiente desvaneciente y este tipo de redes no se adaptaban bien a todas las secuencias de datos [33].

Las células LSTM se presentaron por primera vez en 1997 en un artículo de Sepp Hochreiter y Jürgen Schmidhuber [34]. En el artículo, los autores describen cómo las LSTM no sufren el mismo problema de desvanecimiento de gradiente que experimentan las RNN de vainilla y pueden ser entrenadas en secuencias de cientos de pasos de tiempo. Desde entonces, la arquitectura LSTM ha sido adaptada y mejorada, y variaciones como las unidades recurrentes controladas (GRUs) son ahora ampliamente utilizadas y están disponibles como capas.

Una capa recurrente tiene la propiedad especial de poder procesar datos de entrada secuenciales $[x_1, ..., x_n]$. Consiste en una celda que actualiza su estado oculto, h_t, a medida que cada elemento de la secuencia x_t, un paso de tiempo a la vez. El estado oculto es un vector con una longitud igual al número de unidades de la célula; puede considerarse como la comprensión actual de la célula de la secuencia [33].

En el paso de tiempo t, la célula utiliza el valor anterior del estado oculto h_{t-1} junto con los datos del paso de tiempo actual x_t para para producir un vector de estado oculto actualizado ht . Este proceso recurrente continúa hasta el final de la secuencia. Una vez terminada la secuencia, la capa emite el estado oculto final de la célula, h_n que pasa a la siguiente capa de la red [33]. Este proceso proceso se muestra en la Fig. 2.4.

Ahora que se ha descrito cómo funciona una capa recurrente genérica, es necesario describir el interior de una célula LSTM individual. El trabajo de la célula LSTM es dar salida a un nuevo estado oculto, h_t dado su estado oculto anterior, h_{t-1}, y la incrustación

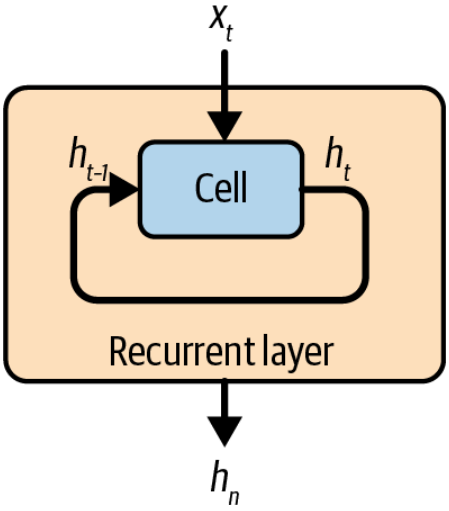


FIGURA 2.4: Capa recurrente de una red LSTM, imagen tomada de [33].

de la palabra actual, x_t [33]. En resumen, la longitud de h_t es igual a el número de unidades del LSTM.

Este es un parámetro que se establece cuando se define la capa y no tiene nada que ver con la longitud de la secuencia. Hay una celda en una capa LSTM que se define por el número de unidades que contiene, del mismo modo que la celda del prisionero de nuestra historia anterior contenía muchos prisioneros [33]. A menudo se dibuja una capa recurrente como una cadena de de celdas desenrolladas, ya que ayuda a visualizar cómo se actualiza el estado oculto en cada paso de tiempo.

Una célula LSTM mantiene un estado de célula C_t, que puede considerarse como las creencias internas de la célula sobre el estado actual de la secuencia. Es distinto del estado oculto, h_t, que es el que finalmente emite la célula después del último paso de tiempo. El estado de la célula tiene la misma longitud que el estado oculto (el número de unidades en la célula) [33]. La Fig. 2.5 muestra una célula y su proceso de actualización.

El estado oculto se actualiza en seis pasos [33]:

1. El estado oculto del paso de tiempo anterior, h_{t-1}, y la palabra incrustada actual, x_t, se concatenan y pasan por la compuerta del olvido. Esta es simplemente una capa densa con una matriz de pesos W_f, un sesgo b_f y una función de activación sigmoide. El vector resultante, f_t, tiene una longitud igual al número de unidades de la celda y contiene valores entre 0 y 1 que determinan cuánto del estado anterior de la celda, C_{t-1} debe conservarse.

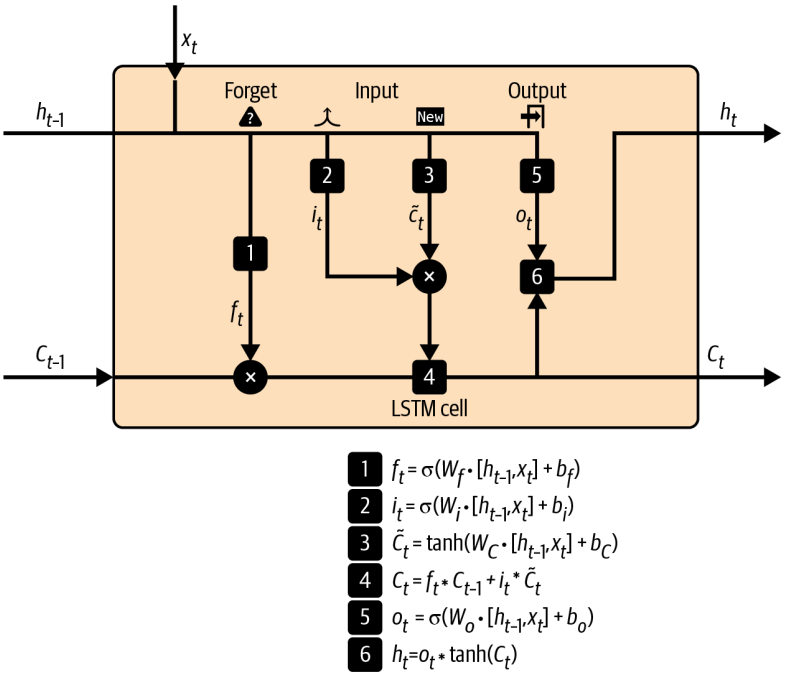


FIGURA 2.5: Célula de una red LSTM, imagen tomada de [33].

2. El vector concatenado también pasa por la compuerta de entrada que, al igual que la compuerta de olvido, es una capa densa con una matriz de pesos W_i, un sesgo b_i y una función de activación sigmoide. La salida de esta compuerta, i_t, tiene una longitud igual al número de unidades de la célula y contiene valores entre 0 y 1 que determinan cuánta información nueva se añadirá al estado anterior de la célula, C_{t-1}.

3. El vector concatenado después es procesado por una capa densa con la matriz de pesos W_C, el sesgo b_C y una función de activación *tanh* para generar un vector C_t que contiene la nueva información que la célula va a considerar conservar. También tiene una longitud igual al número de unidades de la célula y contiene valores entre -1 y 1.

4. f_t y C_{t-1} se multiplican por elementos (*element-wise*) y se añaden a la multiplicación por elementos de i_t y C_t. Esto representa el olvido de partes del estado anterior de la célula y la adición de nueva información relevante para producir el estado actualizado de la célula, C_t.

5. El vector original concatenado también es procesado por una compuerta de salida: una capa densa con la matriz de pesos W_o, el sesgo b_o y una activación sigmoide. El vector resultante, o_t, tiene una longitud igual al número de unidades de la célula y almacena valores entre 0 y 1 que determinan la cantidad de estado actualizado de la célula, C_t, que debe salir de ella.

6. o_t se multiplica elemento a elemento con el estado actualizado de la celda C_t después de aplicar una activación *tanh* para producir el nuevo estado oculto, h_t.

2.4.3.2. Transformadores

Un transformador es un tipo de arquitectura de red neuronal, inicialmente fue propuesta para problemas de procesamiento de lenguaje natural (NLP), pero eventualmente ha sido adaptado a cualquier tipo de problema con datos de secuencias. Fueron presentados en el 2017 [35] y han obtenido resultados de estado del arte en varios problemas donde redes LSTM o redes neuronales recurrentes (RNN) habían tenido los mejores resultados. La arquitectura de un transformador consiste de dos bloques, uno de codificador y otro de decodificador, los bloques se pueden visualizar en la Fig. 2.6 y serán descritos a continuación en su propuesta original para NLP.

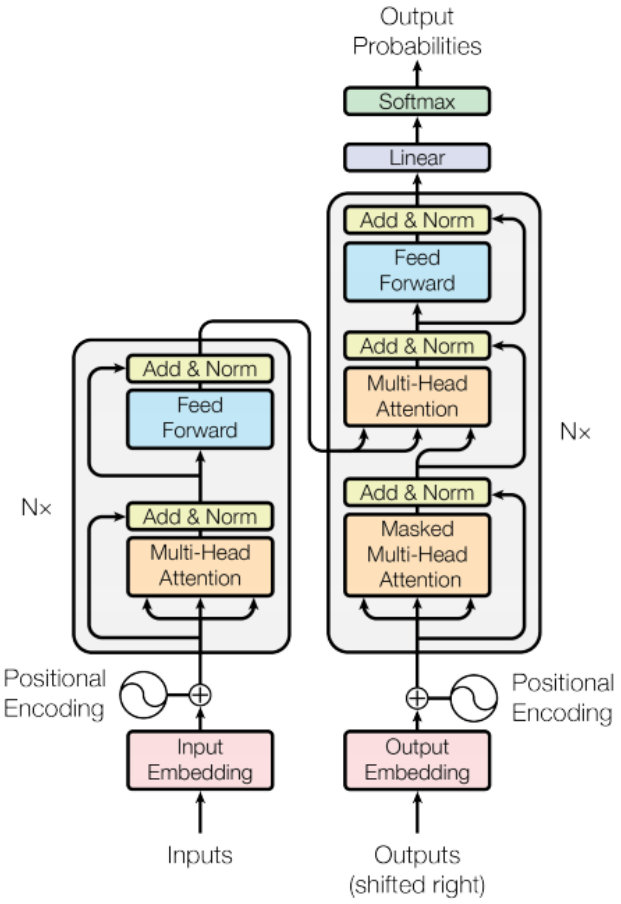


FIGURA 2.6: Arquitectura del modelo Transformador, imagen tomada de [35]

2.4.3.3. Input Embeddings

Los transformadores no aceptan texto sin formato como entrada [35]. Por lo tanto, la entrada de las palabras o datos tiene que ser como secuencias completas, es decir, no se pueden presentar todas las instancias de un dato una por una, en cambio, se tiene que generar una sola instancia que tenga incrustadas todas las características de cada una de las instancias para una palabra, seña, imagen, etc.

2.4.3.4. Positional Encoding

Los *embeddings* representan un token en un espacio d-dimensional donde los tokens con un significado similar están más cerca unos de otros. Sin embargo, los *embeddings* no codifican la posición relativa de los tokens en una oración [36]. Como su nombre lo indica, la codificación posicional codifica la posición de las palabras en la secuencia.

La codificación posicional (PE) se calcula mediante las siguientes fórmulas [36]:

$$PE_(pos, 2i) = sin(\frac{pos}{1000(\frac{2i}{d_{model}})}) \tag{2.1}$$

$$PE_(pos, 2i+1) = cos(\frac{pos}{1000(\frac{2i}{d_{model}})}) \tag{2.2}$$

$PE_(pos, 2i)$ nos dice que si el valor es igual a 1 entonces una palabra está en la primera mitad de la oración, si el valor es 0 entonces está en la segunda mitad; análogamente $PE_(pos, 2i+1)$ se puede hacer uso de la ecuación para identificar la posición relativa de una palabra [36].

La codificación posicional funciona porque la posición absoluta es menos importante que la posición relativa [36]. Por ejemplo, si tenemos la oración “Todas las personas tienen un aspecto bueno“, no se necesita saber que la palabra “bueno“ está en el índice 6 y la palabra “aspecto“ está en el índice 5. Basta recordar que la palabra “bueno“ tiende a seguir a la palabra “aspecto“.

2.4.3.5. Encoder Input

Después de agregar la codificación posicional al vector de *embeddings*, los tokens estarán más cerca entre sí en función de la similitud de su significado y su posición en la oración [35].

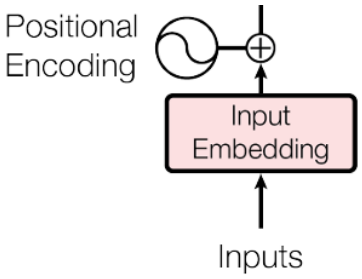


FIGURA 2.7: Entrada del modulo codificador, imagen tomada de [35].

2.4.3.6. Encoder

El trabajo del codificador es mapear todas las secuencias de entrada en una representación continua abstracta que contiene la información aprendida [36], es decir, cómo las palabras se relacionan entre sí, esto se puede visualizar en la Fig. 2.8.

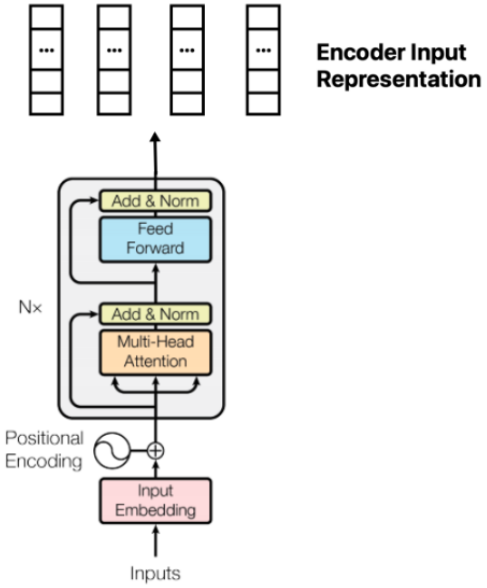


FIGURA 2.8: Modulo codificador del Transformador, imagen tomada de [35].

2.4.3.7. Scale Dot-Product Attention

Después de alimentar los vectores de consulta Q, clave K y valor V a través de una capa lineal, se calcula el producto escalar de los vectores de consulta y clave. Los valores en la matriz resultante determinan cuánta atención se debe prestar a las otras palabras en la secuencia dada la palabra actual [35]. Esta parte de la arquitectura se visualizar en el diagrama de bloques de la Fig. 2.9

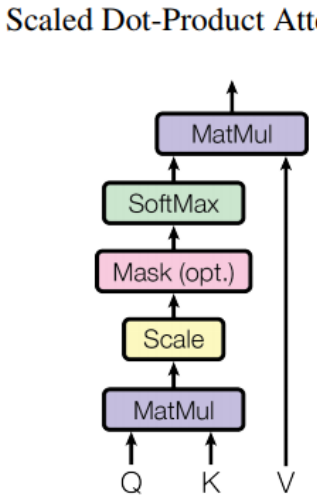


FIGURA 2.9: Modulo de atención de producto punto escalado, imagen tomada de [35].

Es decir, cada palabra (fila) tendrá una puntuación de atención para cada otra palabra (columna) en la secuencia; en la Fig. 2.10 hay un ejemplo de una matriz de puntuaciones [37].

	On	the	river	bank
On	89	27	44	15
the	33	96	64	61
river	66	10	91	54
bank	42	67	55	88

FIGURA 2.10: Matriz de puntuaciones con base en la atención que se deben de prestar las clases, imagen tomada de [37].

El producto escalar d_k se escala por un factor de raíz cuadrada de la profundidad. Esto debido a que para valores grandes de profundidad, el producto escalar crece rápidamente en magnitud haciendo que la función *softmax tenga problemas en el aprendizaje* [35].

$$Attention(V, K, Q) = softmax_k(\frac{QK^T}{\sqrt{d_k}})V \tag{2.3}$$

Una vez escalados los valores, se aplica una función $softmax$, para obtener valores en el rango [0-1], la ecuación 2.3 resume todo este proceso [35]. La Fig. 2.11 muestra un ejemplo de una matriz resultante de todo lo recién descrito.

Finalmente, se lleva a cabo el producto entre la matriz resultante y el vector de valor [35]; operación ilustrada en la Fig. 2.12.

	On	the	river	bank
On	0.7	0.1	0.1	0.1
the	0.2	0.9	0.5	0.4
river	0.4	0.1	0.8	0.3
bank	0.3	0.4	0.3	0.8

FIGURA 2.11: Matriz de valores escalados después de pasar por la función $softmax$, imagen tomada de [37].

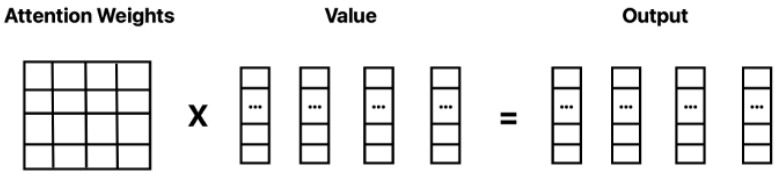


FIGURA 2.12: Salida del modulo de atención, imagen tomada de [37].

2.4.3.8. Multi-Headed Attention

Q, K y V se dividen en múltiples cabezas porque permite que el modelo atienda conjuntamente la información de diferentes subespacios de representación en diferentes posiciones [35]. Por ejemplo, dada la palabra *the* en la oración *On the river bank*, el primer encabezado prestará más atención a la palabra *bank*, mientras que el segundo encabezado prestará más atención a la palabra *river*.

FIGURA 2.13: Ejemplo de una oración y la atención que prestan según la cabeza iterada, imagen tomada de [37].

La salida de atención para cada cabeza se concatena y pasa a través de una capa densa (Fig. 2.14) [35].

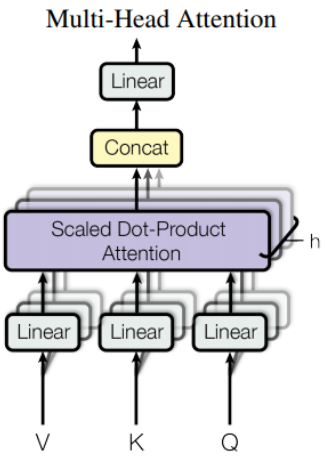


FIGURA 2.14: Modulo de atención de cabezas múltiples, imagen tomada de [35].

2.4.3.9. Residual Connections, Layer Normalization, and Feed Forward Network

El *embedding* de entrada posicional original se agrega al vector de salida de atención de múltiples cabezas. Esto se conoce como conexión residual. Cada capa oculta tiene una conexión residual a su alrededor seguida de una capa de normalización. Las conexiones residuales ayudan a evitar el problema del gradiente de desvanecimiento en las redes profundas. La salida finaliza pasando a través de una red de retroalimentación hacia adelante [37].

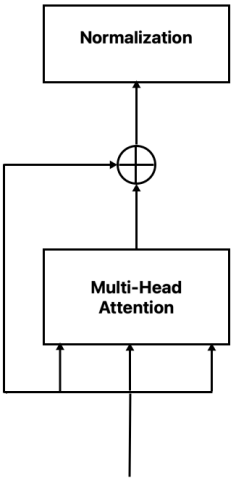


FIGURA 2.15: Entrada a modulo a capa de normalización, imagen tomada de [35].

2.4.3.10. Decoder

El trabajo del decodificador es generar texto, al igual que el codificador, el decodificador tiene capas ocultas similares. Sin embargo, a diferencia del codificador, la salida del decodificador se envía a una capa softmax para calcular la probabilidad de la siguiente palabra en la secuencia [37], la Fig. 2.16 muestra este modulo.

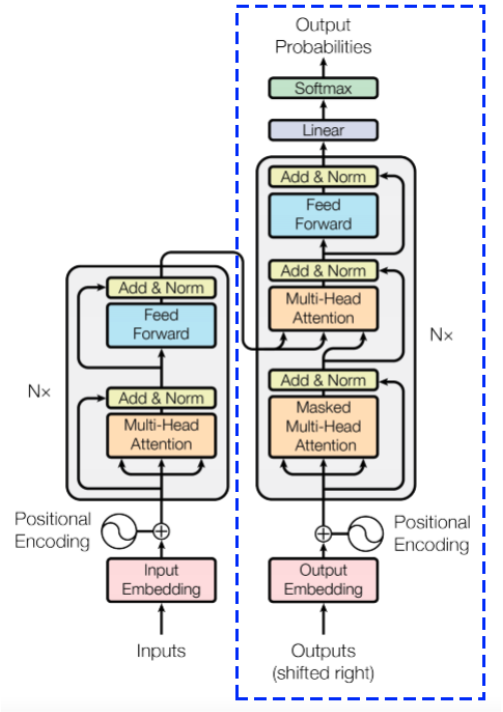


FIGURA 2.16: Modulo de decodificador en Transformador, imagen tomada de [37].

2.4.3.11. Decoder Input Embeddings & Positional Encoding

El decodificador es autorregresivo, lo que significa que predice valores futuros basados en valores anteriores. El decodificador predice el siguiente token en la secuencia mirando la salida del codificador y atendiendo automáticamente a su propia salida anterior. Al igual que se hace en el codificador, se agregan codificaciones posicionales al *embedding* de palabras para capturar la posición de los tokens en la oración [37].

2.4.3.12. Masking

Dado que el decodificador intenta generar la secuencia palabra por palabra, se utiliza una máscara de anticipación para indicar qué entradas no deben utilizarse [37]. Por ejemplo, al predecir el tercer token en la oración, solo se deben usar los tokens anteriores, es decir, el primer y el segundo token.

	On	the	river	bank
On	0.7	0	0	0
the	0.2	0.9	0	0
river	0.4	0.1	0.8	0
bank	0.3	0.4	0.3	0.8

FIGURA 2.17: Matriz de masking del modulo del decodificador, imagen tomada de [37].

2.4.3.13. Salida

Como se mencionó anteriormente, la salida de las capas ocultas pasa por una capa final de softmax. Si se tiene un vocabulario de 10 000 palabras, la salida del clasificador será un vector de longitud 10 000 donde el valor en cada índice es la probabilidad de que la palabra asociada con ese índice sea la siguiente palabra en la secuencia [37].

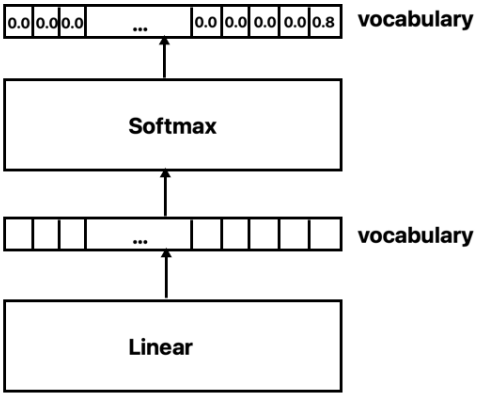


FIGURA 2.18: Capas finales en el modelo de Transformador, imagen tomada de [37].

Capítulo 3

Estado del Arte

En las siguientes secciones se identificaron los trabajos relacionados que fueron considerados para la revisión del estado del arte. En primer lugar, se hallaron dos sistemas principales de reconocimiento de lengua de señas; el reconocimiento aislado y el reconocimiento continuo. En ambos tipos de sistemas se detectó una metodología general compuesta por varias etapas, las cuales son revisadas a detalle en las siguientes secciones. Finalmente, se presenta una discusión donde se presentan áreas de oportunidad y limitaciones.

3.1. Reconocimiento Aislado

El reconocimiento aislado analiza con base en una secuencia de imágenes la seña que está presente. Mucha de la investigación realizada hasta el momento se enfoca en este problema, pues por dato de entrada se sabe que sólo hay una seña presenta, lo que reduce la complejidad en comparación del reconocimiento continuo. Por ejemplo en [9] se aborda el reconocimiento de los dígitos (el cual es un subcaso del reconocimiento aislado) de forma dinámica utilizando un modelo condicional de campos aleatorios y centrándose en descriptores manuales, además se afirma que identificando los movimientos de epéntesis se pueden obtener mejores resultados. Elpeltagy et al. [38] a pesar de no tener buenos resultados, describen que los mismos son prometedores porque están trabajando con un nuevo conjunto de datos, el enfoque que utilizaron para el reconocimiento fue a través de un modelo generado con un bosque aleatorio y análisis canónico.

En otro trabajo Ye et al. [39] detallaron un método para establecer el tamaño de las ventanas para generar clips, los cuales buscan definir una longitud aproximada por seña, luego trabajaron con flujos ópticos y datos de profundidad que se clasifican con una red

híbrida, la cual está compuesta por una red neuronal seguida de una red recurrente. En [40] se propuso un enfoque para el aumento de datos basado en los datos originales, lo cual es de gran interés en el momento del entrenamiento, en particular en condiciones donde no se tienen muchos datos y el cual también ayuda en la generalización del modelo; para la etapa de clasificación trabajan con modelos de inferencia probabilísticos y mecanismos de fusión. Zhang et al. [41] proponen un mecanismo para eliminar la información redundante con el fin de generar descriptores más relevantes como principal aporte, además utilizaron una red de fusión convolucional como clasificador.

Li et at. [42] es un trabajo que emplea mecanismos prototípicos de memoria y atención para las fases de extracción y reconocimiento de características. En [43] los autores se centran en el uso de la información relativa a la postura o a las articulaciones del esqueleto como rasgos descriptivos, los cuales se extraen mediante técnicas de aprendizaje profundo como redes neuronales convolucionales, para la fase de clasificación se realizan mecanismos de fusión y reglas de inferencia. Elakkiya y Vanitha [44] exploran el uso de componentes no manuales, en particular de las expresiones faciales como descriptores, que junto con otros generados en base a la región de la mano siguen unas reglas difusas para generar cúmulos de regiones similares que a través de un mecanismo de agrupamiento realizan tareas de clasificación.

En [45] a través de un método propuesto se detectaron las coarticulaciones del cuerpo, esto para diferenciar de forma correcta los distintos tipos de señas, lo que ayuda a definir un intervalo de duración aproximado por seña. Sin embargo, su aplicabilidad es un poco limitada ya que la publicación describe que sólo se trabaja con señas que emplean únicamente las manos para su realización, lo cual omite información de características no manuales. Kumar et al. [46] se centraron en la definición de una metodología que utiliza descriptores manuales y expresiones faciales, el método de clasificación que hace uso de técnicas bayesianas.

Ma et al. [47] presentaron una propuesta de utilizar un marco para el aprendizaje de conceptos, este se centra en el aprendizaje de detalles de pequeña granularidad para luego aprender ideas compuestas o complejas, en este caso los conceptos son las señas que están presentes en el vocabulario del conjunto de datos ocupado. En [48] se establece un enfoque que opera independientemente del señante y se basa en tres etapas algorítmicas distintas que operan en cascada: la primera para la detección de rostros y manos, la segunda para la extracción de características y la tercera para el reconocimiento de señas. En [49] a través de un sensor Leap Motion y un modelo cinemático se propone realizar el reconocimiento aislado de lengua de señas, la propuesta se probó con múltiples clasificadores y el conjunto de datos utilizado se colocó en un repositorio público.

En [50] se discute el reconocimiento de expresiones en un sistema experimental de apoyo a las personas sordas en una oficina cuando solicitan una tarjeta de identificación. Se propuso un método de procesamiento de flujo continuo de datos de imágenes e información de profundidad registrados por un dispositivo Kinect, posteriormente el vector de características ha sido inspirado en la investigación lingüística. La clasificación se probó utilizando los tres métodos más comúnmente usados para reconocer gestos dinámicos: k vecinos cercanos, modelos ocultos de markóv y redes bi de largo y corto término.

Tur y Keles [51] proporcionaron un marco de tres etapas que permite la clasificación de señas aisladas usando modelos de secuencia basados en modelos ocultos de markóv y en redes de largo y corto término. El primer módulo se utiliza para extraer características, el segundo módulo se utiliza para reducir la dimensión, si es necesario, y el tercer módulo sirve como clasificador de la secuencia. En [52] se introdujo un algoritmo eficiente para traducir el gesto de la mano de entrada en la lengua de señas india (ISL) en un texto y un discurso significativo en inglés. Para reconocer la postura de la mano, la región de la mano es segmentada con precisión y los rasgos de la mano son extraídos usando características de aceleración, Histograma de Gradientes Orientados y Patrones Binarios Locales. El sistema reúne los tres clasificadores de características entrenados usando la máquina de vectores de apoyo.

Hu et al. [53] se dedican al reconocimiento aislado de lengua de señas ocupando un enfoque difuso en el que las características no manuales juegan un papel importante. Para abordar el tema anterior, propusieron una Red de Mejora Global-local (GLE-Net). Su objetivo es mejorar las representaciones de características a partir de dos aspectos complementarios, el módulo de mejora global para la información contextual y el módulo de mejora local para las claves de grano fino. Con estos módulos se busca identificar patrones globales y locales.

3.2. Reconocimiento Continuo

Los sistemas de reconocimiento continuo buscan reconocer una secuencia de señas que generalmente están presentes en un video [4, 6, 54]. Este reconocimiento es el más completo por lo que es más adecuado para las necesidades de las aplicaciones de reconocimiento de lengua de señas en la vida real; por lo tanto, se ha prestado mayor atención a su investigación y se examinará más a fondo en el resto de esta sección.

En Wei et al. [55] detallaron un novedoso modelo de clasificación híbrido, que combina técnicas de visión con técnicas de procesamiento de lenguaje natural, en particular en la parte de visión trabaja con la generación de clips y redes neuronales convolucionales y en

el área de procesamiento de lenguaje natural trabaja con n-gramas. En [56] propusieron un modelo que ocupa enfoques de aprendizaje profundo de una manera interesante, dado que se hace uso tanto de redes neuronales convolucionales como de una red de corto y largo término para extraer características temporales, posteriormente se define un mecanismo de fusión de los resultados de los diferentes flujos de datos con los que se llevó a cabo una etapa del entrenamiento.

Kindiroglu et al. [57] presentan un enfoque basado en características temporales que son extraídas a partir de mapas de calor referentes a las articulaciones del cuerpo, además mediante el uso de una red neuronal convolucional realizan tareas de clasificación para el reconocimiento. Koller et al. [58] es uno de los trabajos más citados dentro del tema de investigación por la metodología presentada pero también porque es el considerado como la referencia del conjunto de datos RWTH-PHOENIX-2014 que es uno de los puntos de referencia. En un trabajo posterior Koller et al. [59] propusieron un modelo híbrido entre los métodos de aprendizaje profundo y los métodos de aprendizaje secuencial, en particular haciendo uso de las redes neuronales convolucionales y los modelos ocultos de Markov.

En Elakkiya y Selvamani [60] se presenta un enfoque de extracción de características a través de subunidades, es decir, descomponen el problema, en este caso las señas, en sus componentes más básicos y luego generalizan hasta lograr un reconocimiento adecuado. En [61] se propone el uso de una Red de Atención Jerárquica con Espacio Latente para el reconocimiento continuo de lengua de señas, que eliminó tanto la segmentación temporal propensa a errores como la síntesis de frases en los pasos de post-procesamiento. En otro trabajo de Huang et al. [62] se especifica el uso de mecanismos de atención para dirigir el proceso de extracción de características a áreas relevantes, con ello los autores proponen que los resultados mejoran en la fase de clasificación, que se realiza también con un método de reconocimiento de patrones basado en atención temporal.

Pu et al. [63] presenta un *framework* de aprendizaje profundo para el reconocimiento continuo de lengua de señas basado en red neuronal convolucional 3D y una red convolucional dilatada, con una estrategia de optimización iterativa. Utilizan el enfoque de clasificación temporal conexionista para alinear cada clip con su etiqueta de seña correspondiente dentro de la frase, y utilizan estas propuestas de alineación generadas (las llamadas seudo etiquetas) para afinar su extractor de características. En un trabajo posterior Pu et al. [64] ocuparon una arquitectura con una fase de extracción de características espacio-temporales a través de una red neuronal convolucional y una fase de modelado de secuencias con una red paralela de largo y corto plazo.

Cheng et al. [65] proponen una fase de extracción de características en la que a través de un módulo de codificación y decodificación de señas son capaces de realizar tareas

de reconocimiento a través de un clasificador conexionista temporal. Cihan Camgoz et al. [66] es una obra que no sólo abarca el tema del reconocimiento sino también el de la traducción, en particular en el reconocimiento se utilizan redes de transformación que están compuestas por redes neuronales convolucionales y módulos de atención. Bilge et al. [67] propone una metodología que sólo necesita pocos datos para la formación haciendo uso de un enfoque de aprendizaje de *zero-shot*, para la fase de extracción de características se ocupan las referentes a la región de la mano y el cuerpo, para la etapa de clasificación hacen uso de redes neuronales convolucionales y una red paralela de largo y corto plazo para modelar secuencias de diferente tamaño.

En [16] se explora un tema diferente al del reconocimiento de lengua de señas, lo que se busca es detectar si en una imagen o video los movimientos de manos son debido a una interacción que involucre lenguaje hablado o lengua de señas o por alguna otra razón, para ello genera diferentes flujos de datos de entrada basados en flujos ópticos e historia del movimiento, con los cuales extrae características a través de una red neural convolucional para finalmente realizar tareas de clasificación con redes neuronales para cada flujo de datos y un mecanismo de fusión de los resultados.

Zhou et al. [68] propone un módulo temporal para el modelado de secuencias de diferentes tamaños de escala basado en una red neuronal convolucional, que luego clasifica por medio de un clasificador temporal conexionista. En Camgoz et al. [18] declaran ser los primeros en proponer transformadores que permiten relaciones inter e intra contextuales, centrándose en descriptores de poses, expresiones faciales y manuales, además propone un clasificador que ocupa un modelo codificador-decodificador con auto atención. En [69] se aborda el estudio del reconocimiento continuo utilizando un modelo de convolución residual para encontrar descriptores y un módulo con una red con memoria de largo y corto término bidireccional para modelar secuencias de diferente longitud, ya en la etapa de clasificación se utiliza una red residual.

Zare y Zahiri [70] propusieron un método para hacer un sistema de reconocimiento independiente del señante y adaptarlo contra los cambios de distancia del mismo con respecto a la cámara usando características invariantes contra la transición, el cambio de escala y la rotación. La clasificación de los datos se lleva a cabo mediante tres clasificadores, incluyendo Bayes, k vecinos cercanos, y una red neuronal. Ravi et al. [71] proponen entrenar una arquitectura de 4 flujos, donde cada uno contiene una red neuronal convolucional, las cuales hacen uso de datos multimodales (espaciales, profundidad, temporales, manuales), dicha arquitectura fue probada con datos en tiempo real, aplicando un mecanismo de optimización que permite obtener buenos resultados aún en la falta de datos en la fase del entrenamiento.

Ko et al. [72] introdujeron un nuevo conjunto de datos de lengua de señas que se anotó manualmente en coreano y propuso un modelo de traducción del lenguaje de señas neurales basado en los modelos de traducción de secuencia a secuencia. En [19] aplicó los modelos de secuencia a secuencia con un mecanismo de atención a para hacer la traducción al idioma inglés, el trabajo hace énfasis en que la arquitectura propuesta es la principal aportación.

Moryossef et al. [73] propusieron una simple representación de flujo óptico humano para videos basada en la estimación de la pose para realizar una clasificación binaria por fotograma utilizando un modelo recurrente, de manera similar como en Borg et al. [16]. Compararon varias entradas posibles, como la estimación de la pose de cuerpo entero, la estimación de la pose parcial y los cuadros envolventes ocupados, y contrastaron su tiempo de inferencia para la detección de la lengua de señas en videoconferencia en tiempo real.

En [74] se propone un enfoque multicanal adversario para la producción de lengua de señas. Enmarcando la producción de lengua de señas como un juego de minimax, presentaron un discriminador adversario condicional que mide el realismo de las secuencias de señas generadas y dirige al generador hacia una producción articulada. También introdujeron la producción no manual de características para encapsular completamente los articuladores de la lengua de señas. En Xiao et al. [75] presentaron un novedoso método para reconocer y generar automáticamente secuencias de señas utilizando un modelo de generación condicional entrenado en datos de secuencias de señas. Implementaron un modelo de probabilidad de dos niveles para describir las secuencias de señas y un modelo de mezcla gaussiana para describir los gestos de las señas.

Así como [9] investigó los movimientos de epéntesis para segmentar señas en reconocimiento aislado, en el caso del reconocimiento continuo [7] determinan la localización de límites temporales entre las señas. Su enfoque emplea representaciones de redes neuronales convolucionales 3D con un refinamiento iterativo para discernir en casos ambigüos. En un trabajo posterior [8] se añade el uso de un mecanismo de pseudo etiquetado que se centra en los cambios de movimiento abruptos que ocurren en los datos, con esto se busca un mayor nivel de granularidad en los datos que ayudaron a obtener mejores resultados.

3.3. Etapas en el Reconocimiento de Lengua de Señas

La Fig. 3.1 muestra la metodología general identificada en los sistemas de reconocimiento de lengua de señas [4–6]. En particular, se presenta una etapa de preprocesamiento de

datos en la que se pueden realizar operaciones de mejora de imagen, segmentación de las regiones de interés y seguimiento posterior; una etapa de extracción de características que sean descriptivas, las cuales suelen estar relacionadas tanto con rasgos espaciales (posiciones, diámetros y formas geométricas), como con rasgos temporales (velocidad, trayectorias y diferencias en el tiempo y entre distintas regiones de interés). Por último, se aborda una etapa de reconocimiento a partir de los descriptores extraídos, donde se busca la correcta identificación de las señas gesticuladas y en el caso de los sistemas de reconocimiento continuo, la alineación de las mismas, es decir, que se identifique correctamente el punto donde comienza y termina una seña, e incluso los espacios donde no existen señas. En las siguientes subsecciones se abordan aspectos relacionados a cada tema en la revisión del estado del arte.

FIGURA 3.1: Metodología general en el reconocimiento de lengua de señas.

3.4. Preprocesamiento

En los sistemas de reconocimiento la segmentación de las regiones de interés es un paso crítico, estas segmentaciones pueden hacerse de forma automática, semimanual y manual. Dado que las tareas de extracción y clasificación suelen ser complejas, muchas investigaciones optan por que esta parte del proceso se haga de forma manual y semimanual [44, 56, 57]. Enfoques recientes han decidido ocupar los datos en crudo para

posteriormente hacer uso de técnicas de aprendizaje profundo como las redes neuronales convolucionales para la extracción de las características.

Típicamente, en la etapa de preprocesamiento se realizan tareas como la aplicación de filtros para la mejora de los datos de entrada, el redimensionamiento, mejoras de contraste, filtros morfológicos, o la segmentación de las regiones de interés. La Fig. 3.2 muestra que la tarea más frecuente que se identificó en la revisión del estado del arte para la etapa del preprocesamiento es la de la segmentación de la región de la mano, lo que demuestra que se presta mucha más atención a las características manuales y que aún no hay una exploración amplia de las características no manuales, a pesar de que éstas son muy importantes para las lenguas de señas en la gramática de varias señas.

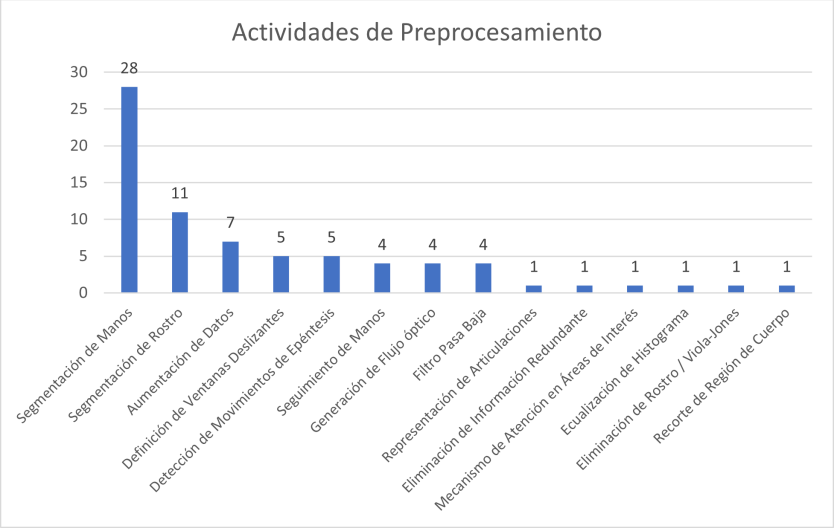


FIGURA 3.2: Actividades de preprocesamiento identificados en la revisión del estado del arte.

Con base a lo mencionado en secciones anteriores, puede notarse que la mayoría de las investigaciones se centran en la región de una o ambas manos [4, 9, 18, 38, 41, 43–45, 49, 52, 56–59, 67, 69], algunos más se centran en la región de la cabeza [4, 5, 18, 43, 50, 58, 73], sin embargo la estimación de la postura del cuerpo aún no se ha explorado completamente.

Un aspecto a considerar es la aumentación de datos [39–41, 50], para esta etapa se suelen generar los datos de forma artificial haciendo rotaciones o escalamientos con base en las imágenes originales, esto ha demostrado en los trabajos que se han ocupado que el proceso de entrenamiento es más robusto. Por otro lado, la estimación de los tamaños de las ventanas deslizantes de forma automática se ha abordado en algunos trabajos [39, 47, 55], donde el propósito es reducir el tiempo de procesamiento al no procesar todos los datos o bien delimitar aproximadamente la duración de cada seña que está presente en la secuencia de entrada, este enfoque no se explora en profundidad en la

mayoría de los trabajos y sin duda podría ser de gran ayuda a reducir la complejidad del reconocimiento de oraciones al dividir el problema en problemas más pequeños.

3.5. Extracción de Características

Esta etapa es crítica en el reconocimiento de lengua de señas, a diferencia de las actividades del preprocesamiento que son opcionales en muchos trabajos, esta etapa es forzosa. Para cualquier método de clasificación, lo que tomará como entrada son los vectores de características adquiridos en esta etapa. El método utilizado para la extracción de características debe de ser lo más robusto y fiable, independientemente de las modificaciones en las condiciones de brillo, la posición, el tamaño y la dirección del objeto en una imagen/vídeo [15, 76]. La Fig. 3.3 muestra que las características extraídas identificadas en el trabajo relacionado son de tipo espacial (E) o temporal (T), la mayoría de ellas están basadas en la región de la mano, situación que se refuerza por el hecho de que es la misma región que suele segmentarse con mayor frecuencia como se menciona en la subsección anterior.

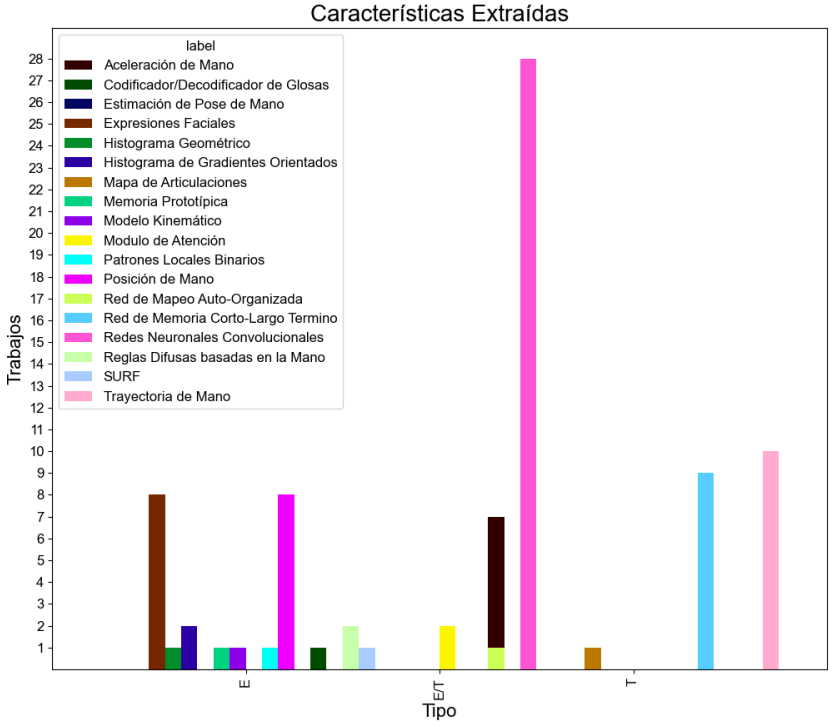


FIGURA 3.3: Características extraídas en la revisión del estado del arte.

Dado que el problema del reconocimiento continuo se considera un problema de representación de secuencias ordenadas, las características temporales son clave para el

buen funcionamiento de los sistema. Estas características temporales suelen basarse en la mano y en las articulaciones del cuerpo, en particular en su trayectoria y velocidad a lo largo del tiempo en los datos de entrada [4, 9, 42, 44–46, 50, 57, 58, 65, 77].

Recientemente, con los buenos resultados obtenidos en múltiples tareas de visión por computadora, las redes neuronales convolucionales se están empleando cada vez más como extractores de características espaciales y temporales [4, 5, 16, 18, 39–41, 43, 51, 53, 55, 56, 59, 62, 64–69, 78–80], un enfoque que es visualmente evidente en la Fig. 3.3 por un amplio margen respecto de las otras características extraídas que fueron identificadas.

3.6. Reconocimiento de Patrones

Se requiere un clasificador para el reconocimiento de lengua de señas para identificar correctamente a qué clase (en este caso, corresponden al vocabulario de las distintas señas presentes) pertenece un dato de entrada. A partir de un conjunto de datos, se obtienen los distintos vectores de características y se utilizan para realizar el entrenamiento del método de clasificación, el clasificador entrenado identifica la clase relacionada con las señas y muestra los textos o reproduce el audio según sea el caso [15, 76].

La Fig. 3.4 muestra cuatro tipos principales de técnicas de clasificación:

1. Redes con memoria de largo y corto término
2. Modelos Ocultos de Markov
3. Máquinas de Vectores de Soporte
4. Clasificador Temporal Conexionista

Pero cabe resaltar que el tipo de método de reconocimiento dependera del problema que se trate, ya sea reconocimiento aislado o reconocimiento continuo. Los Modelos Ocultos de Markov y la Máquina de Vectores de Soporte han sido empleados mayormente en trabajos de reconocimiento aislado, mientras que el otro par de métodos se han ocupado en trabajos de reconocimiento continuo.

Recientemente se ha investigado el uso de técnicas de aprendizaje profundo como las redes con memoria de largo y corto término [4, 5, 49, 56] o las redes neuronales convolucionales [4, 5, 49, 57, 73] que han demostrado ser exitosas. Sin embargo, un enfoque interesante que se ha propuesto es el de las redes híbridas [39–41, 44, 55, 56] en las que

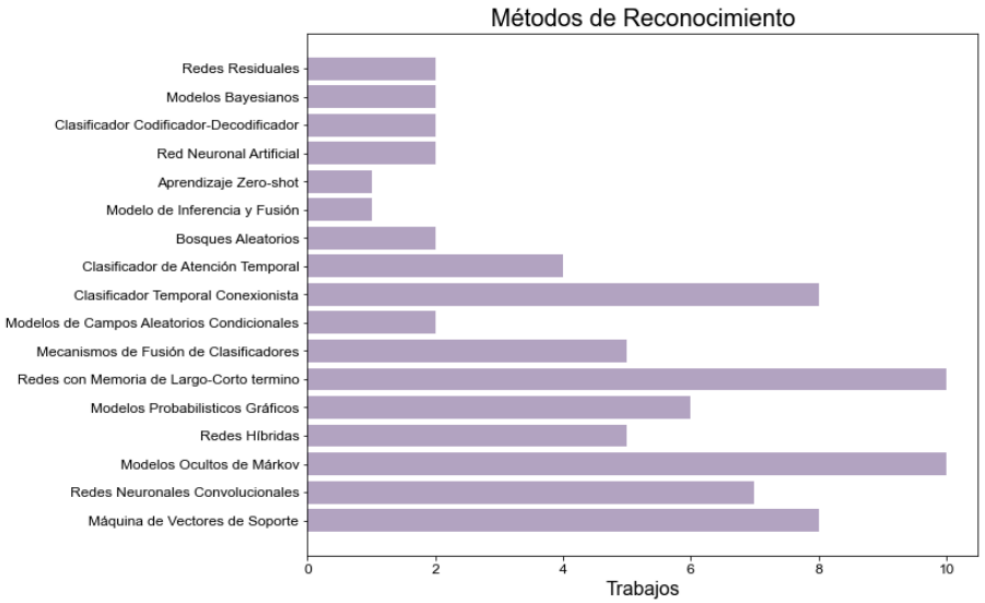


FIGURA 3.4: Métodos de reconocimiento identificados en la revisión del estado del arte.

es posible diseñar un modelo basado en descriptores visuales para clasificar y, por otra parte, diseñar otro modelo basado en factores lingüísticos o de procesamiento de lenguaje natural. El hecho de disponer de modelos híbridos también plantea la necesidad de proponer mecanismos de fusión que puedan unir y clasificar correctamente los resultados de ambos modelos, lo que ha dado lugar a otro tipo de clasificador que utilizan diferentes mecanismos de fusión [16, 41, 53] para determinar sus predicciones con respecto a los datos de entrada.

Como el reconocimiento continuo es un problema de ordenamiento de secuencias, los modelos ocultos de Markov también han demostrado ser relevantes para este tipo de problemas, por lo cual han sido empleados en varios trabajos [5, 50, 51, 59, 81]. Recientemente, se han empleado dos tipos de modelos de clasificación que han demostrado ser exitosos, los clasificadores conexionistas temporales [18, 64, 67, 68] y los modelos de atención que pueden ser ocupados en fusión con algún otro método o por sí mismos [40, 42]. Al final en este apartado hay mucha experimentación por parte de todos los autores, buscando unir distintos enfoques para obtener los mejores resultados.

3.7. Conjuntos de Datos

Muchos investigadores han llevado a cabo la captura de sus propios datos, en algunos casos estos datos se hacen públicos, pero en la mayoría de los casos sólo se especifican en las publicaciones de sus investigaciones. Como se muestra en la Fig. 3.5, hay cuatro

conjuntos de datos públicos principalmente utilizados en los sistemas de reconocimiento de lengua de señas:

- RWTH-PHOENIX-Weather 2014 [4, 18, 43, 44, 58, 59, 68] que corresponde a la lengua de señas alemana, en particular los datos son predicciones meteorológicas dadas en un programa de noticias
- SIGNUM [4, 58, 59] que también corresponde a la lengua de señas alemana
- ASLLVD [4, 44, 56, 60] que corresponde a la lengua de señas american, este conjunto de datos tiene registros preparados tanto para el reconocimiento aislado como para el continuo
- CSL que pertenece a la lengua de señas china [41, 55, 62, 64, 65, 68, 75], que también contiene datos tanto para el reconocimiento aislado como para el reconocimiento continuo, los datos de este conjunto fueron capturados con el dispositivo Kinect por lo que los datos contienen imágenes RGB, datos de profundidad y datos referentes a información de las articulaciones del cuerpo

También cabe resaltar que otro apartado que se visualiza en la figura, es el de los datos propios. Se considera de este tipo a todos los conjuntos de datos que hayan sido recabados por los propios investigadores, y que no hayan sido proporcionados de forma pública a través de algún repositorio o referencia.

3.8. Limitantes y Áreas de Oportunidad

A pesar de que esta revisión se enfoca principalmente en los trabajos que realizan reconocimiento continuo, en revisiones de otros autores se detalla que la mayor parte de la investigación realizada ha sido sobre el reconocimiento aislado por lo que se requiere más trabajo en esta área [4, 5]. Aunado a esto, a continuación se abordan algunos de las áreas de oportunidad detectadas en esta revisión de literatura.

- **Conjunto de datos estándar**. Los conjuntos de datos también son un problema para lograr el desarrollo de un sistema de reconocimiento de lengua de señas de alto rendimiento. Por el momento el conjunto de datos RWTH-PHOENIX-Weather 2014 [58] es el de referencia, sin embargo, como se detalló el vocabulario del mismo es limitado. El desarrollo de un conjunto de datos que sea estándar, que tenga un vocabulario amplio y que no sea capturado en un ambiente controlado aún es un problema abierto en estas investigaciones [10, 82].

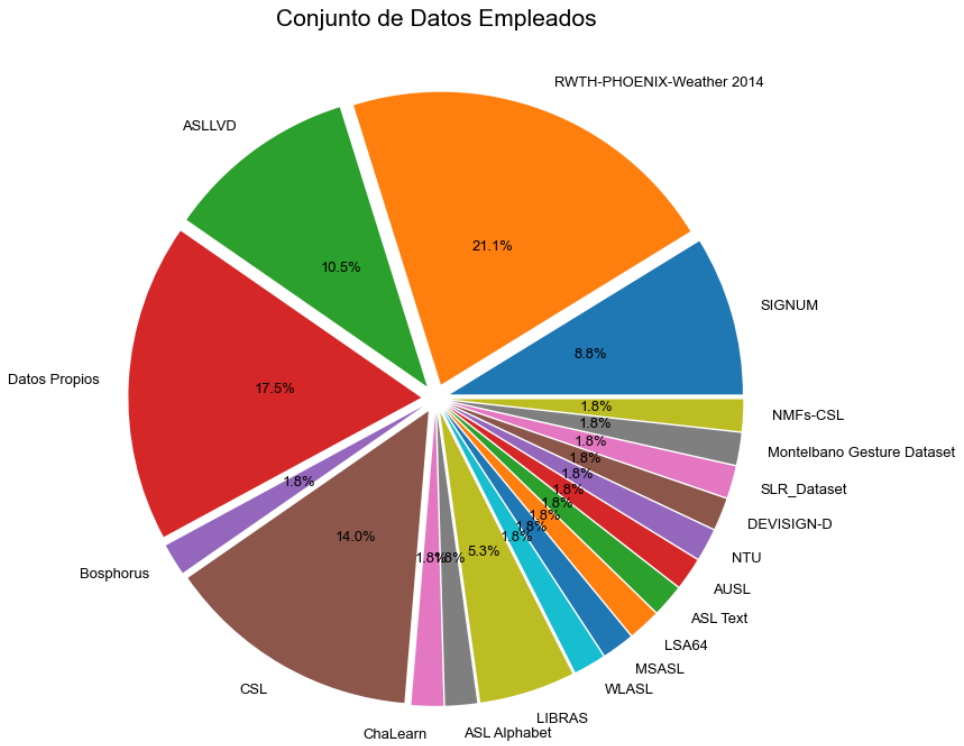


FIGURA 3.5: Conjuntos de datos ocupados que se identificaron en la revisión del estado del arte.

- **Extracción de características no manuales**. Esta revisión identificó que la extracción de características no manuales, es decir, las que se basan en las expresiones faciales, la postura corporal o partes específicas del cuerpo, no son exploradas de forma más exhaustiva. En los trabajos examinados, como se muestra en la Fig. 3.3, se observa que las características no manuales no se extraen con tanta frecuencia como las características manuales, y dado lo que se detalla en las reglas gramaticales de las lenguas de señas, se está omitiendo información que puede ser clave para mejorar los resultados de reconocimiento.

- **Detección transiciones entre señas**. Otro hallazgo clave de esta investigación es que la detección de movimientos de epéntesis puede ser útil en la estimación de la longitud de cada una de las señas presentes en una oración [15], en esta revisión analizamos su uso en la sección de extracción de características, el hecho de encontrar trabajos muestra que la investigación está en su inicio. Sin embargo, los trabajos encontrados sólo se centran en el reconocimiento aislado [9], por lo que es un área de oportunidad que puede ser explorada.

- **Adaptación a señantes.** Al igual que cualquier lengua, la comunicación correcta de todos los que conocen alguna lengua de señas, depende enteramente de las personas, en este caso de los señantes y como sucede con las lenguas habladas,

varias personas se comunican de distinta forma a pesar de ocupar el mismo lenguaje, esto puede deberse a que hablan más rápido, inventan palabras o modifican las existentes a través de modismos [3]. En la lengua de señas sucede lo mismo, un señante puede gesticular las señas más rápido con respecto a otro señante y también se pueden suscitar modificaciones en el vocabulario a través de modismos o nuevas palabras. La adaptación a distintos señantes y a las variantes que cada uno presente en su forma de comunicarse a través de alguna lengua de señas sigue siendo un problema abierto para el reconocimiento de lengua de señas.

Capítulo 4

Desarrollo Metodológico

En este capítulo se aborda el desarrollo metodológico de la investigación, en primer lugar se presenta el diagrama general de todas las fases que son necesarias para poder hacer un reconocimiento de lengua de señas y posteriormente se describen en detalle las tareas realizadas en cada una de ellas.

La Fig. 4.1 muestra la metodología definida para el proyecto de investigación. Como primer punto se obtiene un conjunto de datos con el que se realiza una etapa de preprocesamiento, en esa etapa se realiza la detección de transiciones entre señas. Posteriormente, se detectan las regiones de interés para su correcta segmentación. Con estas regiones, se realiza la extracción de las características con base a los descriptores manuales y no manuales presentes en las lenguas de señas. Finalmente, se realizan los experimentos pertinentes para la fase de clasificación.

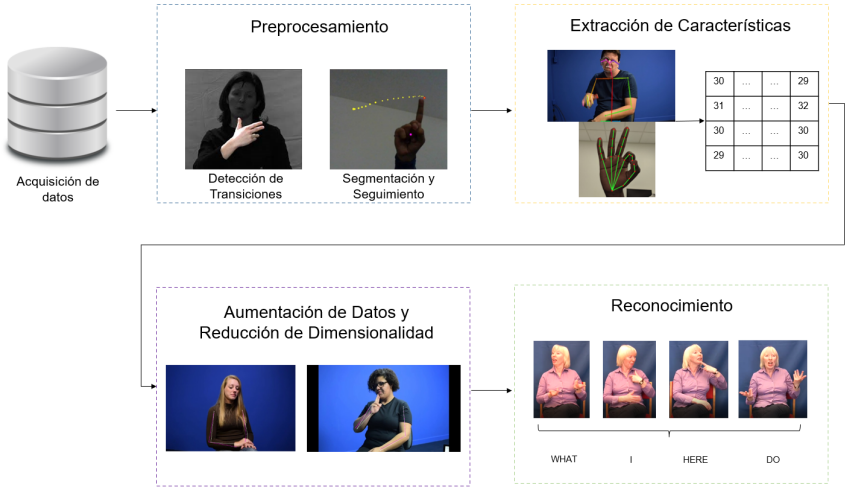


FIGURA 4.1: Metodología general propuesta.

4.1. Preprocesamiento

Como se revisó en el capítulo del Marco Teórico las lenguas de señas tienen componentes manuales y no manuales, siendo el primero el que brinda la mayor información, por tal razón una región relevante a tener en consideración es la de las manos. La primera tarea abordada en la etapa de preprocesamiento es la extracción y anotación manual de imágenes, estas fueron extraídas respecto a un subconjunto de videos de la muestra total del conjunto de datos LIBRAS.

Con dicho subconjunto de datos se ocupa el método de detección de objetos YOLOv5 para reconocer la región de las manos, dicho sistema fue elegido por estar en el estado del arte para el problema de detectar objetos. El problema del reconocimiento de la lengua de señas es muy específico en cuanto a la postura que toman las manos durante la gesticulación de las señas, situación que fue tomada en cuenta en la anotación de las imágenes.

Con el propósito de no generar un modelo sobre entrenado y de que no se necesite un tiempo considerable para las tareas del entrenamiento y evaluación con YOLOv5, sólo se considera un subconjunto de videos de los cuales también se extraen un número específico de imágenes. Lo primero que se realizó fue definir la submuestra con un tamaño de 50 videos, para la selección de estos videos se siguió un muestreo sistemático [83]. El muestreo sistemático M se define en la ecuación 4.1:

$$M = (i, i + k, i + 2k, ..., i + (n - 1)k) \tag{4.1}$$

donde k es el tamaño de incremento para la selección de cada uno de los elementos de la submuestra, este valor se calcula como N/n, donde N es el tamaño total de la muestra y n el tamaño de la submuestra; por último i es un número que se selecciona de forma aleatoria en el rango $[1 - k]$. Para el caso de esta investigación, los valores que se ocuparon fueron $k = 23$ y $i = 10$, los cuales fueron identificados de forma empírica.

Posteriormente con los videos de la submuestra se realizó la extracción de las imágenes de *fotogramas* específicos, para la selección de estos se definió un intervalo de 15 segundos, es decir, en cada video se extrajeron todas las imágenes que fueran posibles empezando con el *fotograma* relacionado a los 15 segundos y haciendo incrementos por el mismo tiempo hasta llegar al final del video. Después de realizar esta operación se extrajeron un total de 614 imágenes.

Con este subconjunto se procedió a la anotación manual respecto a las regiones de las manos y la cabeza, esto se hizo ocupando la herramienta gratuita *Computer Vision*

Annotation Tool [84], que permite definir cuadros envolventes de las regiones que se seleccionen y exportar las anotaciones en diferentes formatos disponibles. En la Fig. 4.2 se muestra un resumen de todo esta etapa.

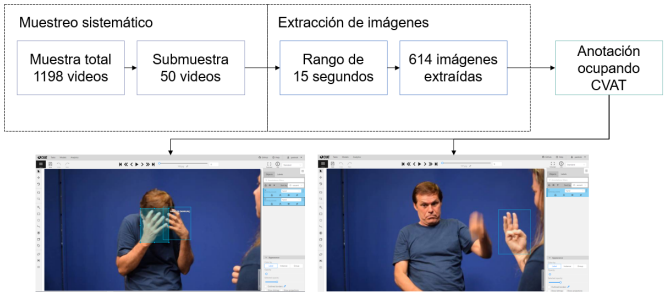


FIGURA 4.2: Actividades realizadas en la etapa del preprocesamiento.

4.1.1. Detección de regiones de interés

Tres regiones de interés fueron definidas: cabeza, manos y postura del cuerpo, las cuales fueron definidas con base en la hipótesis de esta investigación, en particular, estas regiones son empleadas para el apartado de la extracción de características manuales y no manuales. En las siguientes secciones se describe el proceso que se siguió para su detección.

4.1.1.1. Manos y Cabeza

Con el modelo de YOLOv5 obtenido (Ver Apéndice A para más detalles), este sistema aplica una red neuronal convolucional a la imagen completa. Esta red divide la imagen en regiones y predice cuadros delimitadores y probabilidades para cada región. Estos cuadros delimitadores están ponderados por las probabilidades predichas.

Este modelo tiene varias ventajas sobre los sistemas basados en clasificadores, ya que mira la imagen completa en el momento del entrenamiento, por lo que sus predicciones se basan en el contexto global de la imagen. También hace predicciones con una única red de evaluación a diferencia de otros sistemas que requieren miles para una sola imagen, esto lo hace extremadamente rápido.

YOLOv5 está disponible de forma gratuita en el repositorio GitHub y codificado en el lenguaje de programación Python y el *fotogramawork* PyTorch. El repositorio fue descargado en un proyecto de Google Colab y para el proceso de entrenamiento y evaluación, el total de los datos se dividió en tres grupos, uno de entrenamiento, de validación y de prueba; para esto se siguió la práctica en las investigaciones de detección de objetos de ocupar los porcentajes de 70 % para entrenamiento, 20 % para validación y 10 % para

pruebas; además de esto los datos se dimensionaron a una resolución de 416x416 píxeles, esto último porque YOLOv5 obtiene mejores resultados en imágenes con resoluciones que sean múltiplos de 32. Estas operaciones se realizaron a través del sitio roboflow [85] que selecciona de forma aleatoria los datos para cada subconjunto de datos y rescala las imagenes a su nueva resolución, en la Fig. 4.3 se puede apreciar la cantidad de imágenes consideradas en cada uno de los tres subconjuntos generados.

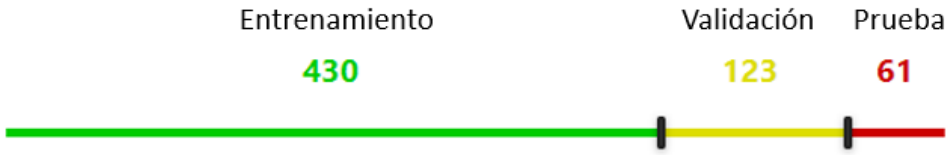


FIGURA 4.3: Distribución de datos para generar los 3 subconjuntos a ocupar con YOLOv5.

4.1.1.2. Postura del Cuerpo

Para la estimación de la postura del cuerpo se decidió ocupar el sistema MediaPipe [27], enfoque que ha demostrado tener buenos resultados en comparación otros trabajos ampliamente ocupados como OpenPose, la principal diferencia que tiene MediaPipe es que está optimizado para ser empleado en dispositivos móviles.

MediaPipe es una plataforma para datos multimedia que emplea soluciones de aprendizaje automático, la cual está desarrollada para realizar múltiples tareas de reconocimiento y segmentación de regiones de objetos (generalmente personas pero también puede ser empleado en objetos). Esta plataforma está desarrollada para trabajar de forma rápida incluso en *hardware* común, pues tiene la posibilidad de ser empleada hasta en dispositivos móviles, los cuales en muchas ocasiones suelen tener recursos limitados.

Este sistema de estimación de la postura del cuerpo además de detectar los puntos claves relacionados con las articulaciones del cuerpo, también es capaz de estimar puntos claves referentes a la región de las manos, pies y del rostro, en total es capaz de estimar 135 puntos claves. Para este trabajo se habilitan las opciones de que se estimen también los puntos de la región de las manos y del rostro, ambos complementan los resultados obtenidos a través del modelo YOLOv5, además de que los puntos faciales dan una estimación de que expresiones se están gesticulando, componente clave en las caracteristicas no manuales en cualquier lengua de señas.

En la Fig. 4.4 se aprecian los primeros resultados que se obtuvieron con este sistema, como se puede apreciar los resultados son bastantes certeros.

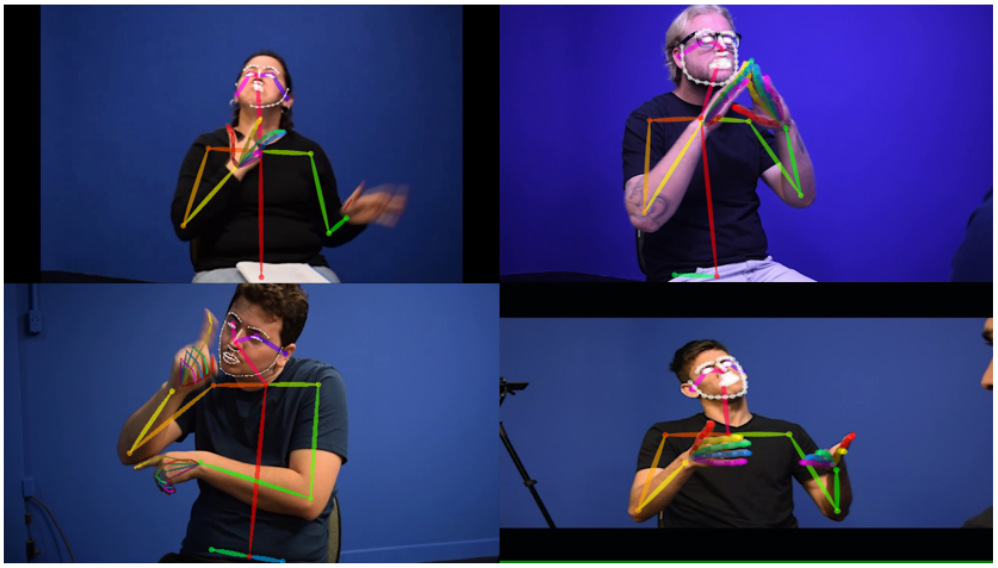


FIGURA 4.4: Resultados de la estimación de la postura del cuerpo.

4.1.2. Detección de transiciones entre señas

Para la problemática de detectar transiciones entre señas, se hace uso de una red dinámica bayesiana. Estos modelos probabilísticos han demostrado ser muy útiles en problemas de aprendizaje de secuencias [6, 7]. Este tipo de redes han sido empleadas en diversas aplicaciones, desde optimización en riego en aplicaciones de agricultura [86], síntesis de población [87], consumo de energía de forma inteligente [88], manejo de residuos en temas de construcción [89] o para el análisis de riesgo de enfermedades [90].

Para el problema en cuestión se propone el diseño de dos redes dinámicas bayesianas. El primer modelo no toma en cuenta datos de la mano menos empleada (secundaria) ni de características no manuales. Este modelo puede apreciarse en la Fig. 4.5, donde los nodos Px y Py hacen referencia a las coordenadas de la posición, el nodo V es la velocidad aproximada, el nodo F es un promedio de los descriptores referentes a la forma, todos estos datos respecto a la mano más empleada (dominante) y el nodo T es el referente a la predicción de la transición. El índice va desde $i = 0 \ldots n$ donde n es el número de fotogramas considerados por cada registro.

Aprender la parte gráfica (es decir, la estructura) de estos modelos a partir de datos es un problema NP-difícil [91]. Se han realizado muchos estudios sobre este tema, lo que ha dado lugar a tres familias diferentes de enfoques: (1) métodos basados en restricciones que apuntan a identificar independencias condicionales en un conjunto de datos y transformar esta información en grafos, (2) métodos basados en puntajes, que optimizan una función de objetivo global mediante la exploración heurística del espacio de soluciones y (3) métodos de búsqueda locales o métodos híbridos que se ocupan de la identificación de

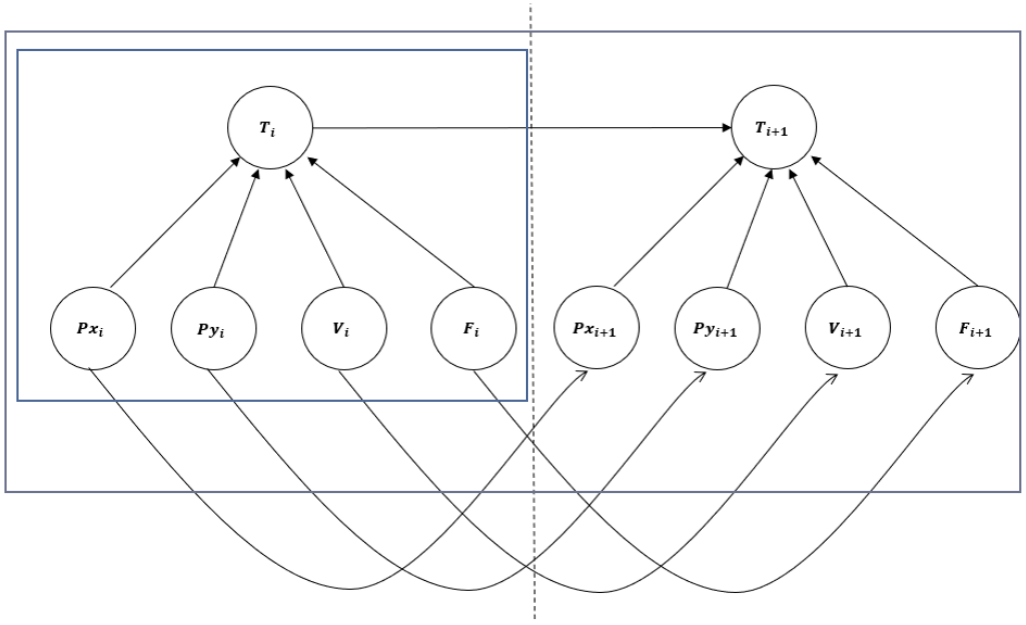


FIGURA 4.5: Modelo de red dinámica bayesiana simplificado.

estructuras locales y la optimización del modelo global restringida con esta información local. Esta última familia de métodos es capaz de escalar a distribuciones con más de miles de variables.

Debido a la complejidad inducida por la adición de la dimensión temporal, el aprendizaje de la estructura DBN también es una tarea compleja. Los algoritmos existentes son adaptaciones de algoritmos de aprendizaje de estructura de redes bayesianas basados en puntajes [91] pero a menudo están limitados cuando el número de variables es alto.

Al igual que el aprendizaje de la estructura, el aprendizaje de parámetros de las RDB es más complejo respecto de las redes bayesianas que no consideran al aspecto temporal. Se debe crear tanto la red inicial B_0 (que especifica una distribución sobre el estado inicial del proceso) como el modelo de transición B_t definido sobre $B(t, t+1)$, que especifica las probabilidades de transición entre los estados del proceso temporal [29].

La red inicial B_0 siguió un algoritmo de aprendizaje de estructura en vez de un enfoque frecuentista, empleado previamente para la creación de las tablas de distribución de probabilidades condicionales (CPDs). El algoritmo ocupado para esta operación es *Dynamic Max-Min Parent Children* [91], el cual es un método de búsqueda local, en particular es una adaptación que permite encontrar estructuras de redes dinámicas bayesianas con más de 10 variables ocupando un algoritmo de búsqueda voraz.

La técnica para el aprendizaje de parámetros de la red dinámica bayesiana, la cual es una variación del método estimación de máxima verosimilitud (MLE) para datos continuos [92] fue ocupado. El método de MLE se aplica de forma iterativa sobre una serie de

instancias (videos) que están relacionadas a lo largo de un lapso de tiempo, al final se obtienen como resultado los parámetros que maximizan las probabilidades de las CPDs. Con los parámetros de la red inicial B_0 y la red de transición B_t se puede inicializar una red dinámica bayesiana con la cual se realizaran una serie de inferencias.

Para la realización de los experimentos con estas redes, en primera instancia se aplicó la normalización de los datos de entrada, para ello se empleo el método de *Min-Max* [93], el cual transforma los valores del vector de características en un rango [0-1]. La idea detrás de la normalización de valores en los vectores de características es que las variables tengan una misma escala y así todas puedan contribuir equitativamente en el ajuste del modelo con lo cual se evite crear sesgos. La formula de la ecuación 4.2 es la que sigue el método, donde x es el valor a normalizar y X es el vector de características.

$$x_{scaled} = \frac{x - min(X)}{max(X) - min(X)} \tag{4.2}$$

4.2. Extracción de características

Con el proposito de poder crear un espacio de características que tome en cuenta descriptores manuales y no manuales y haciendo uso de las regiones de interés que se han identificado, se procede a la etapa de extracción de características. A continuación se describen las características que se consideran para cada región y la forma en la que se calculan.

4.2.1. Posición de las manos

Los valores que se ocupan en este apartado son las coordenadas x e y con respecto al punto que representa el centroide del cuadro envolvente el cual es estimado con YOLOv5 [94]. Estos valores se extraen para cada una de las manos, en caso de que no se detecte alguna mano o no esté presente en escena, los valores que se emplean para cada coordenada serán los de 0. En la Fig. 4.6 se aprecia un ejemplo del cuadro envolvente que se estima y del cual se calcula el centroide.

4.2.2. Expresiones Faciales

MediaPipe es una plataforma para datos multimedia que emplea soluciones de aprendizaje automático, la cual está desarrollada para realizar múltiples tareas de reconocimiento y segmentación de regiones de objetos (generalmente personas pero también puede ser

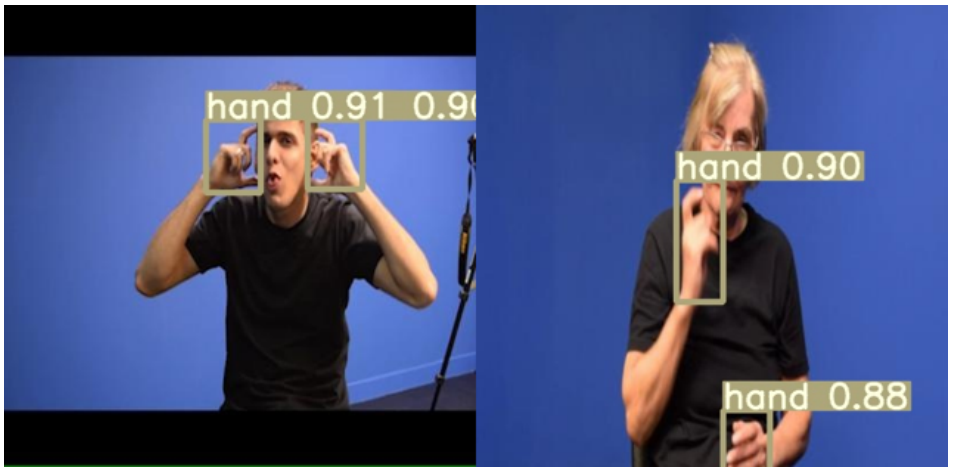


FIGURA 4.6: Ejemplo de estimación de cuadro envolvente en la región de las manos.

empleado en objetos). Esta plataforma está desarrollada para trabajar de forma rápida incluso en *hardware* común, pues tiene la posibilidad de ser empleada hasta en dispositivos móviles, los cuales en muchas ocasiones suelen tener recursos limitados. En la Fig. 4.7 se aprecian ejemplos de las tareas que se pueden realizar con MediaPipe.

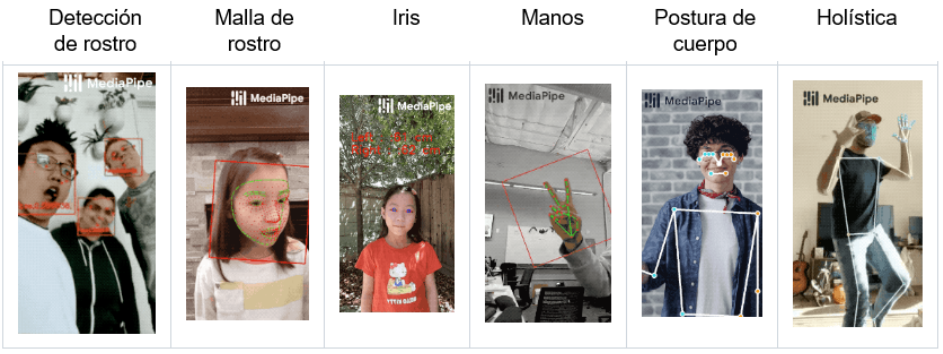


FIGURA 4.7: Ejemplos de tareas que se pueden realizar con la plataforma MediaPipe.

MediaPipe estima una serie de puntos claves tanto de la región del rostro como de las articulaciones del cuerpo de una persona. Los puntos claves que MediaPipe estima respecto de la región del rostro y las expresiones faciales son los que se muestran en la Fig. 4.8.

En el apartado de las expresiones faciales se calculan las distancias entre diversos puntos clave. Este enfoque es con base al que se presenta en el trabajo de [58] y por los puntos clave que retorna MediaPipe.

La distancia que se calcula es la distancia euclidiana, la cual se define en la ecuación 4.3, donde p y q son los dos puntos y $n = 2$, ya que ambos puntos se encuentran en un plano cartesiano. Los puntos que se consideran para calcular sus distancias entre los pares: (0-17), (61-291), (0-94), (52-159) y (282-386).

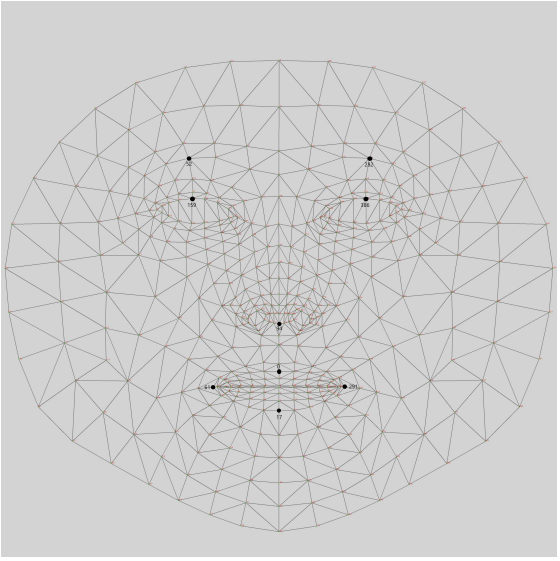


FIGURA 4.8: Puntos clave brindados por MediaPipe relacionados con las expresiones faciales.

$$d\left(p,q\right)=\sqrt{\sum_{i=1}^{n}\left(q_i-p_i\right)^2} \tag{4.3}$$

4.2.3. Forma de Manos y Brazos

De forma similar al proceso realizado en las expresiones faciales y ocupando los puntos clave que retorna MediaPipe, pero ahora para la región de las manos se calculan las distancias euclidianas entre los puntos definidos. En la Fig. 4.9 se aprecian los puntos claves para la región de las manos y en la Tabla 4.1 se detallan los puntos considerados.

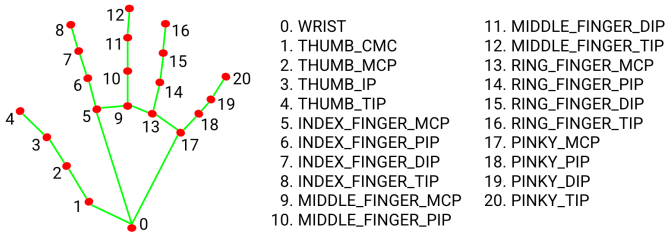


FIGURA 4.9: Puntos clave referentes a la región de las manos soportados por MediaPipe.

Para el caso de la región de los brazos no se calculan las distancias entre pares de puntos, lo que se hace en su lugar es tomar los valores de sus coordenadas y así se ocupa su información espacial. La Fig. 4.10 muestra los puntos claves que retorna MadiaPipe y la Tabla 4.1 detalla los puntos que se consideran.

Tabla 4.1: Puntos clave considerados de la región de las manos.

Región	Puntos
Muñeca - Dedo Medio MCP	0-9
Muñeca - Pulgar MCP	0-2
Muñeca - Meñique MCP	0-17
Meñique MCP - Punta Meñique	17-20
Dedo Anular MCP - Punta Dedo Anular	13-16
Dedo Medio MCP - Punta Dedo Medio	9-12
Dedo Índice MCP - Punta Dedo Índice	5-8
Dedo Pulgar MCP - Punta Dedo Pulgar	2-4

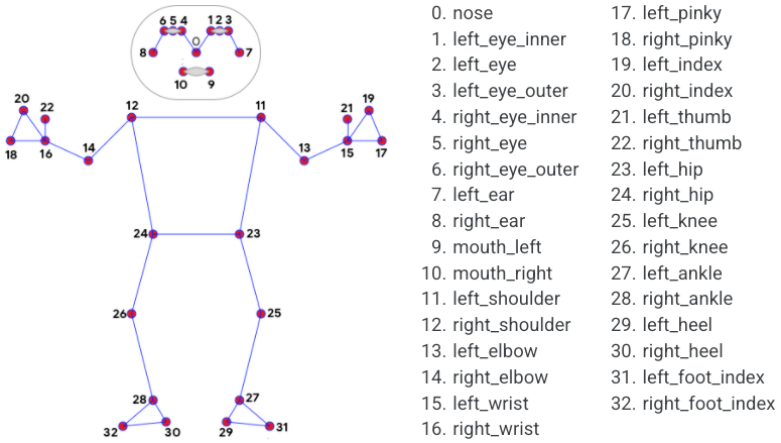


Figura 4.10: Puntos clave referentes a la postura del cuerpo por MediaPipe.

Tabla 4.2: Puntos clave considerados de la región de los brazos.

Región	Puntos
Hombro Izquierdo	11
Codo Izquierdo	13
Muñeca Izquierda	15
Hombro Derecho	12
Codo Derecho	14
Muñeca Derecho	16

4.2.4. Velocidad Aproximada de las Manos

El muestreo sobre los fotogramas de cada video empieza en el fotograma número 3, esto también se realiza con el propósito de poder medir el cambio de posición del centroide del cuadro envolvente de la región de las manos. Con dicho cambio y el tiempo transcurrido en ese tamaño de ventana se puede calcular la velocidad aproximada de las manos. La ecuación 4.4 es ocupada para el calculo del valor de velocidad, donde d viene dada por la distancia euclidiana entre el punto del centroide del cuadro envolvente del muestreo

previo, respecto al punto del centroide del muestreo actual y t es el tiempo transcurrido en el tamaño de ventaneo de 3 fotogramas.

$$v = d/t \tag{4.4}$$

Como parte de la propuesta se realiza la extracción de características no manuales con base a la estimación de la cabeza y la mirada, las cuales no han sido estudiadas en profundidad [17].

Tanto para la estimación de la cabeza como de la mirada se utilizó el toolkit de OpenFace [95], el cual se encuentra de forma pública y con licencia de libre uso para propósitos de investigación. A continuación se describe brevemente de qué forma se realiza la estimación para cada región, además, se detalla la información que será ocupada como descriptor.

4.2.5. Estimación de Cabeza

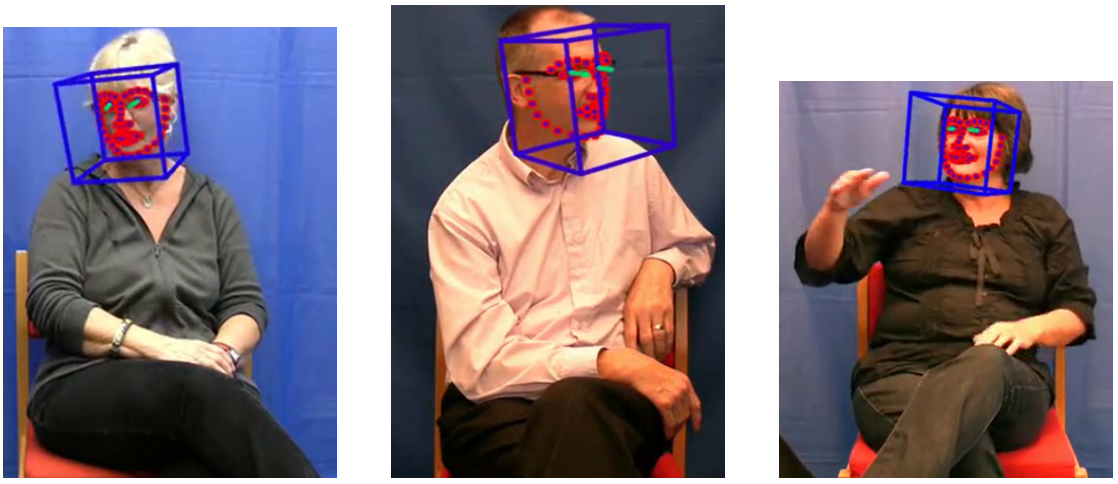
En el apartado de la estimación de la cabeza OpenFace ocupa un arquitectura de aprendizaje profundo, la cual emplea una serie de convoluciones [96] y una etapa de reducción de dimensionalidad de los datos de entrada en su proceso de entrenamiento, este modelo ayuda a estimar los puntos claves referentes a la región del rostro. Con el resultado obtenido se aborda el problema de la perspectiva tomando n puntos [97], el cual suele ser empleado para calibrar cámaras y estimar ángulos de rotación de un objeto respecto de un origen, en este caso el dispositivo de captura. Para la resolución de este problema se emplea la solución presentada el trabajo de [98].

Una vez realizado todo este proceso, OpenFace da la posibilidad de obtener múltiples valores, en particular los que se propone utilizar y que serán analizados en la etapa de reconocimiento de lengua de señas son:

- Coordenadas (x, y, z) de la postura de la cabeza respecto a la cámara que en este caso es el origen
- Rotación (x, y, z) en radianes respecto a la cámara

4.2.6. Estimación de Mirada

OpenFace utiliza un campo neuronal local restringido [99, 100] para detectar los puntos clave de los párpados, el iris y la pupila. Se usa la pupila detectada y la ubicación del



(A) (B) (C)

FIGURA 4.11: Ejemplos de estimación de cabeza y mirada a través de OpenFace.

ojo para calcular el vector de la mirada del ojo individualmente para cada ojo. Se traza una línea desde el origen de la cámara a través del centro de la pupila en el plano de la imagen y se calcula su intersección con la esfera del globo ocular. Esto da la ubicación de la pupila en coordenadas de cámara 3D. El vector desde el centro del globo ocular 3D hasta la ubicación de la pupila es el vector de mirada estimado.

En este caso una vez realizado todo este proceso, los valores que retorna OpenFace y que se propone ocupar son:

- Coordenadas (x, y, z) del vector estimado para cada ojo
- Ángulo de dirección con valores (x, y) en radianes, promediada para los dos ojos

En la Fig. 4.11 se aprecian múltiples ejemplos del resultado de la estimación de la cabeza y la mirada, los resultados son acertados en todos los casos, razón por la que no se es necesario realizar ajustes adicionales.

En la Fig. 4.12 se aprecia un esquema sobre las características extraídas.

FIGURA 4.12: Características extraídas para la tarea del reconocimiento de lengua de señas.

4.3. Reconocimiento de lengua de señas

Se definió para el reconocimiento de lengua de señas la metodología compuesta de las siguientes actividades:

1. Preprocesamiento de palabras
2. Anotación de clases para las instancias
3. Generar conjuntos de entrenamiento, validación y prueba
4. Definición de modelo de reconocimiento (establecer parámetros y topología)
5. Entrenamiento y evaluación con el modelo generado

En el primer punto las 'palabras' son las etiquetas de cada seña, lo que termina representando las clases para el problema del reconocimiento, con el fin de evitar meter caracteres basura en la etapa siguiente, se realiza una etapa donde se eliminan espacios y se pasan todas las palabras a minúsculas.

Para la anotación de las instancias se toman en cuenta los archivos eaf que vienen en todos los registros considerados, estos archivos tienen las anotaciones y también la delimitación temporal de cada seña. Tomando en cuenta el fotograma asociado a cada instancia se comprueba si está dentro del tiempo que se está gesticulando una seña, en caso afirmativo se asigna la etiqueta de la seña, en caso contrario se asigna una clase *blank_transition*, la cual sirve para definir transiciones y estados de reposo. Los últimos tres pasos de la metodología propuesta se describiran a detalle en la sección de Experimentos y Resultados.

4.3.1. Aumentación de datos

La aumentación de datos en el reconocimiento de lengua de señas ha sido un enfoque que se ha explorado recientemente en varios trabajos [101–103]; distintos métodos han sido propuestos, desde aquellos que generan datos sintéticos basados en transformaciones espaciales de regiones de interés [102, 103] hasta aquellos que se basan en inyectar instancias existentes con bases a métricas de evaluación [101].

En la investigación se optó por emplear un enfoque que genere instancias sintéticas basadas en transformaciones espaciales. En particular, se eligieron dos transformaciones propuestas en el trabajo de [103], dichas aumentaciones buscan ayudar a prevenir el

sobreajuste y maximizar la capacidad de generalización en el reconocimiento de los modelos a emplear. Las aumentaciones espaciales en este caso son sobre los datos referentes al cuerpo y dichos datos son empleados en la fase de entrenamiento del modelo pero no en la etapa de prueba en los experimentos.

De acuerdo a lo definido en [103], los parámetros de las aumentaciones fueron seleccionados aleatoriamente de una distribución normal, pero se mantienen consistentes para todos los fotogramas en cada instancia de una seña. Así, pues, las dos aumentaciones implementadas son las siguientes:

1. **Rotación en el plano.** Todas las coordenadas de las articulaciones en cada fotograma son rotadas por un ángulo aleatorio θ de hasta 13 grados en la siguiente forma: $f_{rotate}(x, y) = ((x - 0{,}5)cos\theta - (y - 0{,}5)sin\theta + 0{,}5, (y - 0{,}5)cos\theta + (x - 0{,}5)sin\theta + 0{,}5)$

2. **Rotación en secuencia de las articulaciones.** Las coordenadas de las articulaciones ambos brazos son pasadas de forma secuencial (para mantener un orden), dichos puntos claves son rotados ligeramente con respecto al punto procesado en cuestión en la iteración de la secuencia. La probabilidad de cada articulación de ser rotada es de $3 : 10$ y el ángulo de rotación se elije de forma aleatoria en el intervalo de $[-4, 4]$. Esto busca simular pequeñas variaciones en la ejecución de cada seña, lo cual tiene también como objetivo no cambiar el significado semántico.

En la Fig. 4.13 se muestran ejemplos de aumentaciones de datos tomando la información que se estimó con base a la postura del cuerpo. En la Fig. 4.13 a) se aprecia con líneas punteadas los puntos que corresponden a los puntos originales y con una línea sólida la rotación de dichos puntos, como es claro en la imagen, esta aumentación considera también variaciones en la región de la cabeza a diferencia de la Fig. 4.13 b) donde sólo se considera la región de los brazos.

4.3.2. Red BiLSTM

En el trabajo de [15] se identificó que las técnicas que se han ocupado para problemas de secuencias son las más empleadas; hecho que es lógico dado que se está tratando con problemas que consideran la dimensión temporal. Los métodos seleccionados son una red bidireccionales con memoria de largo y corto plazo (BiLSTM) y un Transformador.

Como toda red neuronal se tiene que establecer una topología y sus parámetros para su proceso de entrenamiento y la generación del modelo de inferencias. En este trabajo se propuso ocupar la topología que se aprecia en la Fig. 4.14, en la cual se establece que

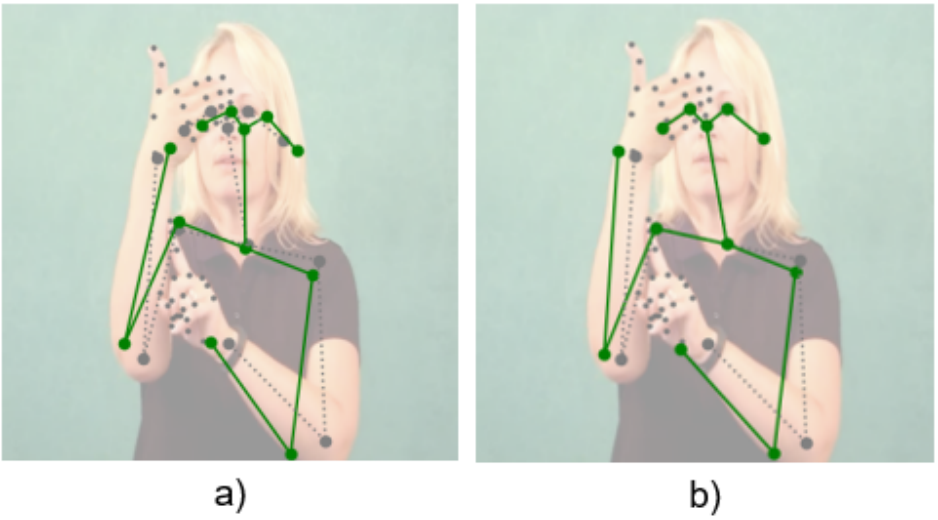


FIGURA 4.13: Ejemplo de aumentaciones de datos, a) rotación en el plano y b) rotación en secuencia de las articulaciones.

se tienen datos de entrada en forma de secuencias (videos), que pasan por la capa de la red BiLSTM, la salida funge como entrada de una capa de una red completamente conectada que tiene como salida una capa *softmax*, que es la función de pérdida empleada para la optimización en el proceso de entrenamiento; esta función es la más empleada en problemas de clasificación.

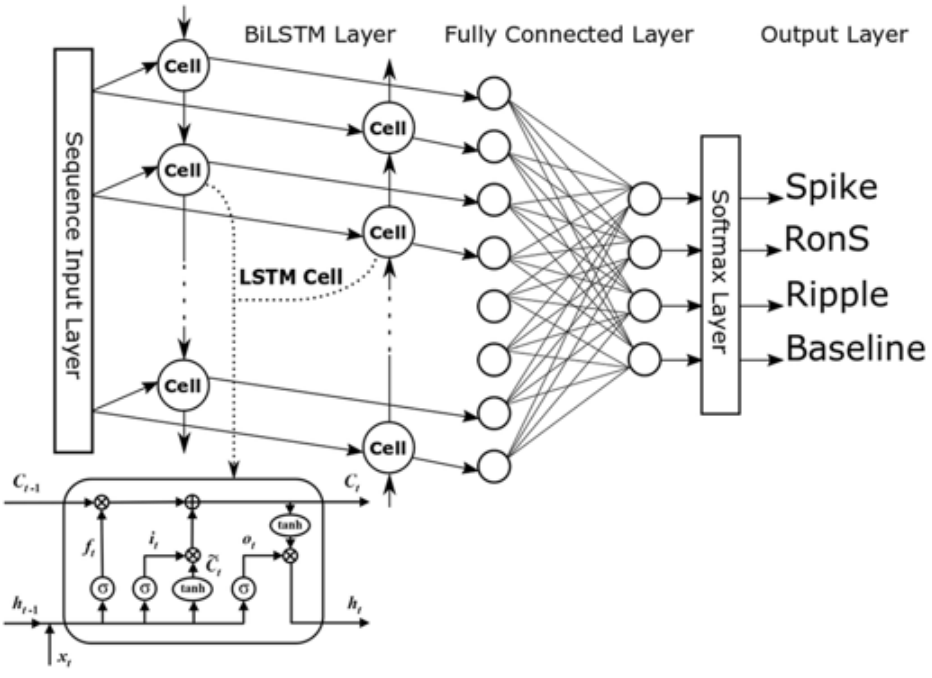


FIGURA 4.14: Arquitectura de la red BiLSTM empleada para el proceso de reconocimiento de lengua de señas.

4.3.3. Transformadores

Diversos métodos de reconocimiento se han ocupado en todos los trabajos relacionados, desde redes neuronales convolucionales (CNN), redes con memoria de largo y corto termino (LSTM), enfoques de atención, unidades bidireccionales de almacenamiento recurrente (BGRU) o las redes bidireccionales con memoria de corto y largo termino (BiLSTM). Otro método ha demostrado tener buenos resultados en problemas con datos de secuencias son los transformadores [17].

Como el modelo descrito en el trabajo [35] fue empleado para tareas de procesamiento de lenguaje natural (NLP), se tienen que hacer ajustes para trabajar con imágenes. Dado que el vector de características que se genera con todas las características extraídas no es posible ocupar la adaptación hecha para el reconocimiento de objetos ocupando transformadores presentado en [104], pues en dicho enfoque los *embeddings* se van generando al procesar las imágenes a través de regiones locales.

Por tal razón se considera la adaptación presentada en [103], donde también se parte de un vector de características previamente extraídas; el ajuste que se realiza a diferencia del enfoque original es que en lugar de ocupar las fórmulas descritas para crear los valores de la codificación posicional se ocupa un arreglo con valores aleatorios con longitud igual al número de características. Además de ello, se modifica el modulo del decodificador, donde se descartan los espacios de *query* y de *key*. Los parámetros del modelo se presentaran en el apartado de los experimentos.

Capítulo 5

Experimentos y Resultados

El siguiente capítulo se aborda la descripción de los conjuntos de datos que fueron empleados en la investigación, con dichos conjuntos se realizó el diseño de una serie de experimentos, los cuales son descritos a continuación, en particular, se detallan los referentes a el reconocimiento de la región de interés de la mano y del reconocimiento de lengua de señas. Ocupando las métricas presentadas, se presentan y discuten los resultados que fueron obtenidos, tanto de forma aislada como en comparación con el estado del arte.

5.1. Conjunto de Datos

A continuación se describen los dos conjuntos de datos con los que se realizaron experimentos a lo largo de esta investigación.

5.1.1. LIBRAS

El conjunto de datos LIBRAS [11] es de libre acceso, fue elaborado por la Universidad Federal de Santa Catarina, donde la lengua de señas que está presente es la brasileña. Este conjunto de datos tiene las siguientes características:

- 540 registros de conversaciones entre dos personas.
- En su mayoría cada registro consta de 4 videos; una toma frontal a cada uno de los participantes en la conversación, una toma frontal a ambos y un toma con un ángulo aéreo.
- 36 personas fueron parte de la captura de los datos.

- Se consideraron 3 grupos de edad para los datos, personas con edades entre 18-29 años, 30-59 años y personas con más de 60 años.
- Cada video tiene una resolución de 640x428 píxeles y una duración de entre 3 minutos hasta 20 minutos.
- Cada registro tiene un archivo xml de formato eaf que corresponde a la anotación de los datos.
- Algunos de los temas que se abordan en las conversaciones son referentes a: bebidas, fechas, familia, frutas, verduras, tecnología, tránsito, conversaciones, tema libre sólo por mencionar algunos.

En la Fig. 5.1 se pueden visualizar un ejemplo de los datos y de cada una de las tomas que se acaban de describir.

FIGURA 5.1: Ejemplos pertenecientes al conjunto de datos LIBRAS.

Este conjunto de datos se generó como parte de un proyecto de preservación de la cultura de la comunidad sorda, por lo cual el vocabulario de los temas presentes tiene la ventaja de ser amplio en comparación con los conjuntos de datos que se identificaron en la revisión del estado arte, eso representa dos hechos, en primer lugar, que existen situaciones realistas y que contienen un vocabulario amplio en los datos, por ejemplo, una conversación entre dos personas.

Con el conjunto de datos, la siguiente tarea que se realizó fue la de depurar los datos para definir el conjunto de datos que formaran parte de la etapa experimental. En la Fig. 5.2 se aprecia que del total de 540 registros; de forma inicial se descartaron 40 registros, esto porque no contenían videos, sólo tenían los archivos de anotación.

De los 500 registros restantes sólo se tomaron en cuenta los videos referentes a las tomas frontales de cada uno de los participantes, es decir, se descartaron los videos que corresponden a las tomas frontales y a la toma aérea. Los videos de la toma aérea se descartaron dado que es imposible realizar la extracción de descriptores con base en la

región de la cara. Una vez descartados los registros y videos denotados, el número total de videos con los que se cuenta es de 1198, estos videos representan la muestra total.

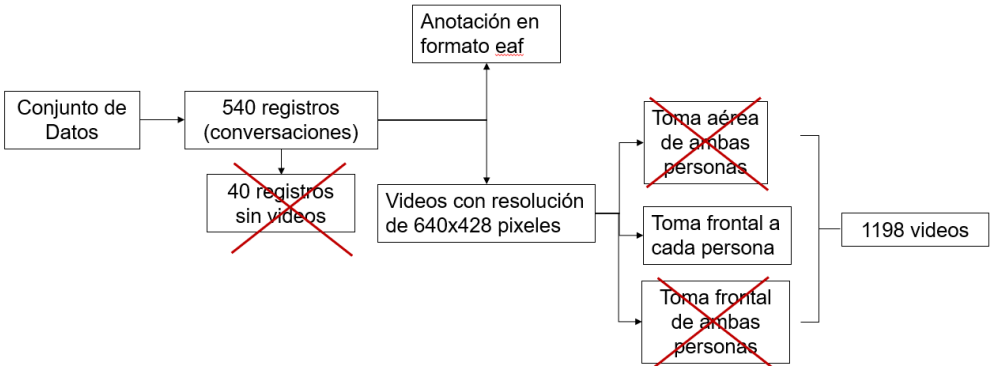


FIGURA 5.2: Definición de la muestra a ocupar en la fase de experimentación del conjunto de datos LIBRAS.

5.1.2. WASL

El conjunto de datos WASL [105] que se componen de datos provenientes de sitios web educativos respecto de la lengua de señas americana (ASL) y de tutoriales de la misma lengua que se encuentran en YouTube; todos estos son almacenados para realizar el reconocimiento de lengua aislado (a nivel de palabra). Este conjunto de datos tiene las siguientes características:

- Se tienen 34,404 videos que contienen 3,126 glosas
- Todos los registros se almacenan en un archivo json, donde cada uno de los registros contiene la url donde está almacenado el video original, la glosa (palabra), le delimitación temporal de la glosa y el cuadro envolvente de la región de la mano dominante, entre otras características.
- Otra característica relevante es el énfasis que se pone en la diversidad de los señantes, pues se señala que cualquier modelo de reconocimiento debe de ser capaz de ser robusto a las variaciones intra-señantes (apariencia de cada señante y estilo y velocidad en la gesticulación de cada señante).
- Los videos tienen una duración que va de 0.36 a 8.12 segundos, siendo la media de 2.41.
- Se definen cuatro experimentos principales, donde se seleccionan las top-K glosas, con los valores posibles de $k = \{100, 300, 1000, 2000\}$. Estos experimentos ocupan los subconjuntos que se generan para cada valor posible de k: WASL100, WASL300, WASL1000 y WASL2000.

En la Fig. 5.3 se muestra un ejemplo de los datos descritos anteriormente.

FIGURA 5.3: Ejemplos pertenecientes al conjunto de datos WASL.

5.2. Resultados en la Detección de Regiones de Interés

Con los datos definidos se procedió a realizar los experimentos de la detección de regiones de interés, en particular a detectar la región de las manos a través del modelo que fue entrenado con el subconjunto de imágenes muestreadas y detallado en el Capítulo previo.

Los parámetros que deben de definirse del modelo YOLOv5 para el proceso del entrenamiento son el numero de épocas, el cual se estableció en 400, esto se definió de forma empírica a través de varios experimentos. Además, se define el tamaño de la imagen en una resolución de 416x416 píxeles y el tamaño del *batch* se dejó en 16.

Una vez entrenado el modelo se evaluaron los resultados obtenidos. Dos métricas fueron empleadas para ello, la primera de ellas es *Intersection over Union* (IoU) [94] que mide la superposición entre 2 regiones, una región viene delimitada por el cuadro envolvente de la anotación que se hizo manualmente y la otra región está dada por el cuadro envolvente que se predijo a través del modelo entrenado. Está métrica nos da un intervalo de confianza sobre la predicción realizada, en la Fig. 5.4 se aprecia como se calcula esta métrica.

La segunda métrica es la del promedio de la precisión obtenida (mAP) [94], para ello se hace uso de la primera métrica descrita del IoU; se define un intervalo de confianza y se establece que cualquier predicción que sea inferior a dicho intervalo es una predicción errónea y cualquier valor igual o mayor a este intervalo definido se toma como una predicción correcta. Al final se realiza la relación de las predicciones correctas sobre el número total de los datos de prueba y esa es la precisión obtenida para el modelo.

Se obtienen las predicciones de los distintos intervalos de confianza que son empleados en trabajos de detección de objetos, los cuales son el intervalo de confianza de

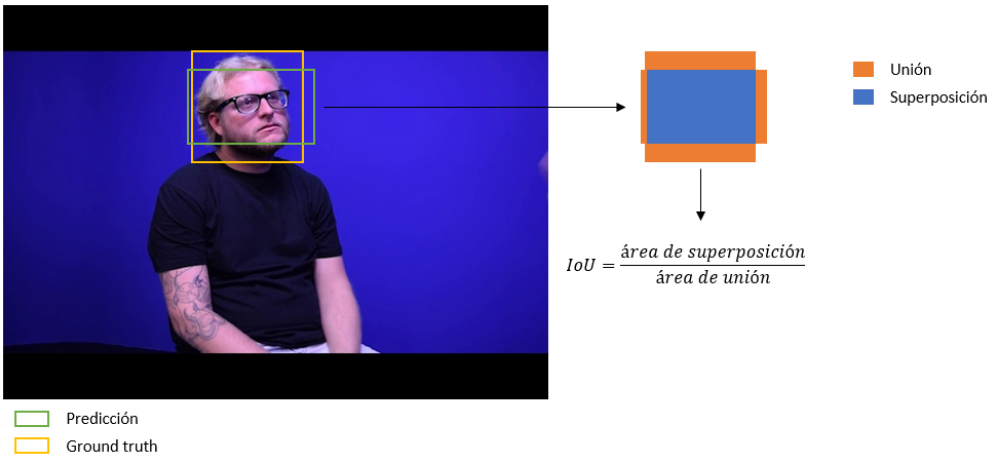


FIGURA 5.4: Descripción gráfica sobre cómo se calcula la métrica IoU.

0.5 (mAP@0,5) y el promedio de las precisiones obtenidas en el intervalo [0,5 − 0,95] (mAP@0,5 − 0,95) con un tamaño de incremento de 0,05. En la Tabla 5.1 se muestran los resultados de mAP y en la Fig. 5.5 se muestran el proceso de entrenamiento a través de las épocas definidas mediante gráficas.

TABLA 5.1: Resultados obtenidos la detección de las regiones de las manos y a cabeza.

Región	Métrica	Resultado
Manos	mAP@0,5	96,22 %
Manos	mAP@0,5 − 0,95	62,22 %

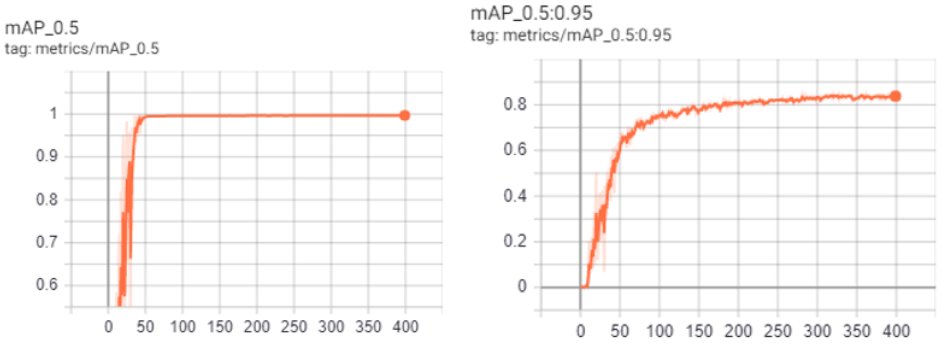


FIGURA 5.5: Gráficas del proceso de entrenamiento a través de las épocas definidas.

Como se aprecia, los resultados obtenidos son bastantes certeros. Las imágenes donde hubo menor certeza en las predicciones presentan múltiples oclusiones, por ejemplo, las piernas, los brazos cuando estos están cruzados, la región de la cabeza; por otro lado esta región de la cabeza presenta varios movimientos volviendo la definición de la zona más borrosa. A pesar de esto último, los resultados en ambos casos son buenos, en la Fig. 5.6 se muestran ejemplos de algunas inferencias que se realizaron con el modelo entrenado.

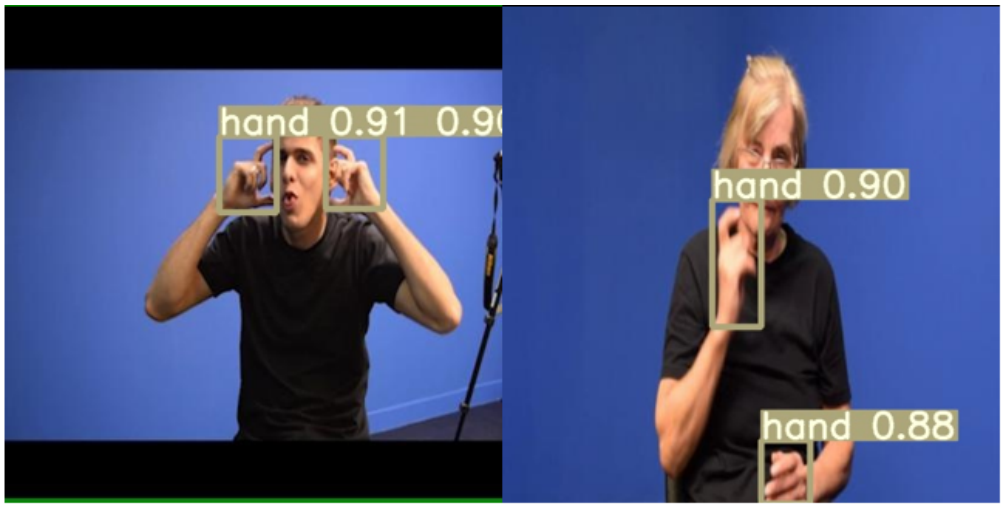


FIGURA 5.6: Inferencias realizadas con los modelos generados, en la fila superior se muestran ejemplos de la región de la cabeza y en la fila inferior de la región de las manos. El número que se muestra es el intervalo de confianza IoU calculado.

5.3. Resultados en el Reconocimiento de Lengua de Señas

Para el diseño de experimentos se toman en cuenta los siguientes pasos:

1. Los conjuntos de datos empleados son LIBRAS [11] y WASL [105]
2. Se consideran 50 videos de LIBRAS y 153 videos de WASL
3. Los videos tienen una duración de 1 segundo hasta 2 minutos aproximadamente
4. Se considera la relación de 70 %-15 %-15 % para obtener los conjuntos de entrenamiento, validación y de prueba, respectivamente
5. Una vez obtenidos los vectores de características, dichos valores son normalizados por el método de estandarización *z-score* [93]
6. Se aplica el método de análisis de componentes principales (PCA) [93] para descartar las características menos discriminatorias

A excepción de la RDB que fue implementada en el lenguaje de programación R ocupando el entorno de desarrollo R studio, el resto de los algoritmos o métodos empleados fueron implementados con el lenguaje de programación Python. En particular, los métodos de reconocimiento fueron mediante el *framework* PyTorch.

En ambos conjuntos de datos y con ambos métodos se realizó una validación cruzada con un valor de *folds*: $k = 3$. Para dicha validación se toma el 70 % que corresponde a

el conjunto de entrenamiento y en él se realizan las tres particiones que se siguen en el algoritmo.

Tanto el conjunto de validación como en el conjunto de prueba no se afectan; además, se asegura que las instancias que se distribuyen en los diversos conjuntos no sean contiguos, es decir que no sean fotogramas obtenidos de forma consecutiva a través del ventaneo definido. En fotogramas consecutivos por muy rápidos que sean los movimientos presentes, no ocurren cambios grandes en todos los atributos y esto podría verse como que se están considerando muchas instancias redundantes en el modelo.

Esto también daría pauta como trabajo futuro a que pudiera implementarse un método donde se analice la varianza en los atributos en instancias contiguas y en algún momento optar por eliminar algunas instancias con el propósito de acelerar el proceso del entrenamiento.

Ambos métodos para el conjunto de datos consideran el ajuste automático de la tasa de aprendizaje a través de un *scheduler*, se tiene una tolerancia de 5 épocas, donde en caso de no presentarse mejora en la función de pérdida, se procede a realizarse el ajuste $new\_lr = lr * 0{,}1$.

5.3.1. Reducción de Dimensionalidad

A pesar de contar con un espacio de características extraídas de forma puntual, lo que ocasiona que sea de una dimensionalidad reducida (44 descriptores) en comparación con otros enfoque que emplean métodos de aprendizaje profundo como extractores de características [6, 15, 17] se realizó una etapa de reducción de dimensionalidad. Con el propósito de preservar únicamente las características que aporten el mayor grado de varianza entre ellas, es decir, que presenten la menor redundancia de información posible, se emplea el algoritmo de análisis de componentes principales (PCA).

Para aplicar el algoritmo de PCA, uno de los parámetros requeridos es el número de componentes, el cual generalmente se establece de forma explicita. Sin embargo, se omite definir dicho parámetro, en su lugar se ocupa un método que lo identifica para ello hay que definir el porcentaje de variabilidad que se busca preservar y se calcula de forma automática el número de componentes [106].

En este caso se eligió esta segunda opción con un 95 % de variabilidad, lo que dió como resultado un total de 20 componentes calculados de forma automática. En la Fig. 5.7 se visualiza a forma de ejemplo, los tres componentes principales que mayor varianza aportan y su relación entre ellos.

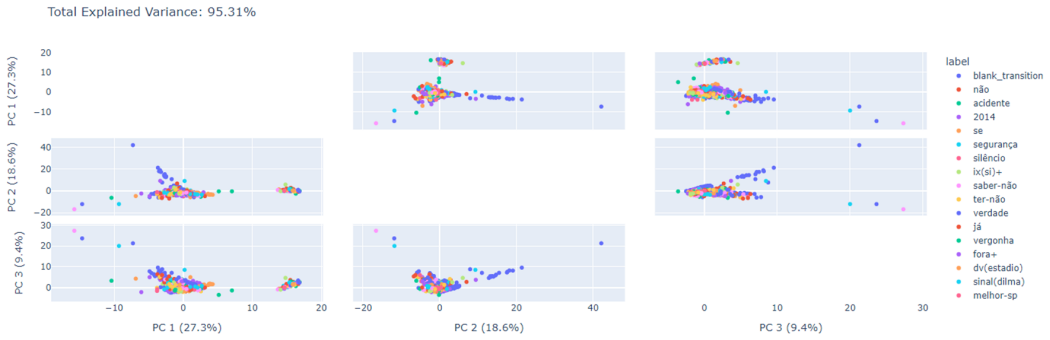


FIGURA 5.7: Relación entre los principales componentes empleando el método de PCA.

5.3.2. Submuestreo de Clase Dominante

Al contar con datos que provienen de conversaciones, se generan muchos estados de reposo y los datos que vienen a nivel de oración también vienen con transiciones entre las distintas clases, por tal motivo, la clase *blank_transition* tiene más instancias que cualquier otra clase, es decir, se tiene un conjunto de datos desbalanceado.

Al hacer el análisis del espacio de características generado para LIBRAS mediante la frecuencia de las clases en todos los registros, se observa en la Fig. 5.8 que efectivamente existe una clase altamente mayoritaria (*blank_transition*) que tiene totalmente desbalanceado el conjunto de datos.

Un aspecto a resaltar es que existen múltiples clases que sólo tienen una instancia; situación que busca mitigarse con la aumentación de los datos para no tener problemas con el modelo obtenido en proceso del entrenamiento.

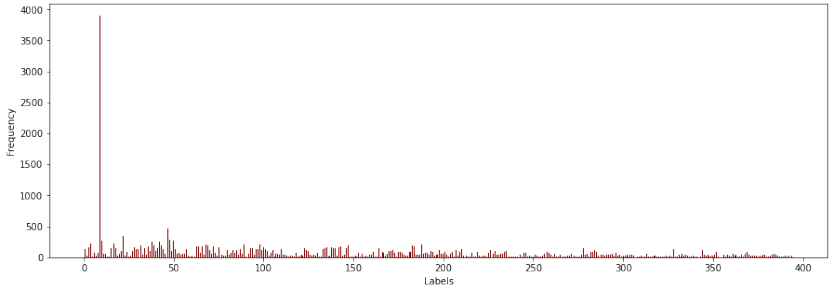


FIGURA 5.8: Visualización de número de instancias por clase con datos de LIBRAS.

Los métodos de remuestreo están diseñados para cambiar la composición de un conjunto de datos de entrenamiento para una tarea de clasificación con un conjunto de datos desbalanceado. La mayor parte de la atención de los métodos de remuestreo se centra en el sobremuestreo de la clase minoritaria.

En este caso dado que muchas de las clases minoritarias tienen sólo una instancia, no se realizo el sobremuestreo de ellas, pues se podría generar una copia de los datos, lo que

eventualmente podría ocasionar un sobre ajuste en el proceso de entrenamiento, por tal razón se explora un método de submuestreo.

Un método clásico para el submuestreo de una clase dominante es el de los enlaces de Tomek, sin embargo esta técnica se centra más en los traslapes entre las distintas clases y no hace un submuestreo tan notorio. La selección unilateral (OSS), es otra técnica de submuestreo que combina enlaces de Tomek y la regla del vecino más cercano condensado (CNN) [107].

Específicamente, los enlaces de Tomek son puntos ambiguos en el límite de la clase y se identifican y eliminan en la clase mayoritaria. Luego, se usa el método CNN para eliminar ejemplos redundantes de la clase mayoritaria que están lejos del límite de decisión [107]. De esta forma se puede esperar que se remuevan más instancias de la clase mayoritaria.

Finalmente, se agrega un último método llamado AllKNN, que emplea una metodología donde busca a los vecinos cercanos y ve en las fronteras si la mayoría son de la clase dominante, en tal caso puede eliminar las instancias.

5.3.3. Resultados de la red BiLSTM en LIBRAS

Los parámetros empleados por la red BiLSTM son los siguientes:

- Épocas: 50
- Tasa de aprendizaje: 0.003
- Capas ocultas: 2
- Celdas en las capas ocultas: 128

Dado que ocupando la red BiLSTM se procesan todas las instancias una por una y que se tiene un desbalance de la clase *blank_transition* (transiciones y estados de reposo) se emplean los métodos de submuestreo para la clase mayoritaria OSS, Tomek y allKNN. Además de ello, también se realizan experimentos sin aplicar el método OSS, es decir, sin hacer submuestreo. Los resultados obtenidos se muestran en la Tabla 5.2.

Los resultados fueron obtenidos al aplicar ambos métodos de aumentación de los datos; en dicho proceso por cada instancia se generó una nueva siguiendo cada método, es decir, por cada instancia se generaron dos nuevas.

Dado que se aprecia una influencia en la aumentación de datos en la mejora de los resultados, se exploró si alguna de ellas brinda información más discriminatoria, así que

Tabla 5.2: Resultados en el reconocimiento de lengua de señas ocupando una red BiLSTM ocupando datos aumentados por los dos métodos definidos con LIBRAS.

Método submuestreo	Exactitud	SD
Sin submuestreo	88.23 %	±4,89
Tomek	87.88 %	±5,51
AllKNN	86.56 %	±5,33
OSS	86.33 %	±5,33

se define un experimento donde sólo se generan datos sintéticos ocupando el método de rotación rotación en secuencia de las articulaciones. Los resultados se muestran en la Tabla 5.3; en ellos se puede visualizar que la exactitud se incrementó y la SD se redujo, por la misma razón en los experimentos que se realizaron posteriormente sólo se consideró ese método de aumentación de datos.

Tabla 5.3: Resultados en el reconocimiento de lengua de señas ocupando una red BiLSTM ocupando datos aumentados con el método de rotación en secuencia de las articulaciones con LIBRAS.

Método submuestreo	Exactitud	SD
Tomek	94.33 %	±3,81
Sin submuestreo	94.31 %	±4,38
AllKNN	93.85 %	±4,79
OSS	93.29 %	±4,41

Como se mencionó en la sección de la etapa de preprocesamiento, el modelo generado con la RDB brinda estimaciones respecto de si una instancia parece ser una transición o no, ocupando los mismo datos que se enuncian al inicio de esta sección, se realizan las estimaciones de todas las instancias y se añaden al vector de características empleado en los experimentos recien descritos.

Además, para estos nuevos experimentos, se creó un objeto encargado de almacenar el modelo que obtiene la mejor exactitud en el conjunto de validación. Por tal motivo en dicho objeto se almacena el modelo como uno de sus atributos y se tienen una serie de métodos, donde la principal corrobora si la función de pérdida es menor en la siguiente época, si es así, guarda el modelo; en caso contrario no guarda el modelo.

Con estos cambios, se generan nuevos resultados, los cuales son mostrados en la Tabla 5.3, de forma llamativa se puede apreciar que todavía se dio una pequeña mejora, la cual no es despreciable considerando que ya de por sí se había alcanzado una exactitud alta. También es interesante que se puede observar que la SD se disminuyó considerablemente, lo cual parece estar relacionado con el hecho de no considerar el modelo de la última época.

TABLA 5.4: Resultados en el reconocimiento de lengua de señas ocupando una red BiLSTM ocupando datos aumentados con el método de rotación en secuencia de las articulaciones y la detección de transiciones con LIBRAS.

Método submuestreo	Exactitud	SD
Sin submuestreo	96.72 %	±0,58
Tomek	96.59 %	±0,26
AllKNN	95.75 %	±0,21
OSS	95.75 %	±1,46

5.3.4. Resultados del Transformador en LIBRAS

A continuación se realizaron experimentos con el método de reconocimiento Transformador, para ello, como se definió, los datos de entrada se preprocesan para formar instancias que correspondan a secuencias completas para cada una de las clases. Es decir, se toman todas las instancias que correspondan a una clase y se compactan en una sola; por tal motivo se generan arreglos de distintas longitudes en cada una de las características. La Fig. 5.9 muestra un ejemplo del preprocesamiento explicado para adecuar los datos al formato correcto requerido por el Transformador.

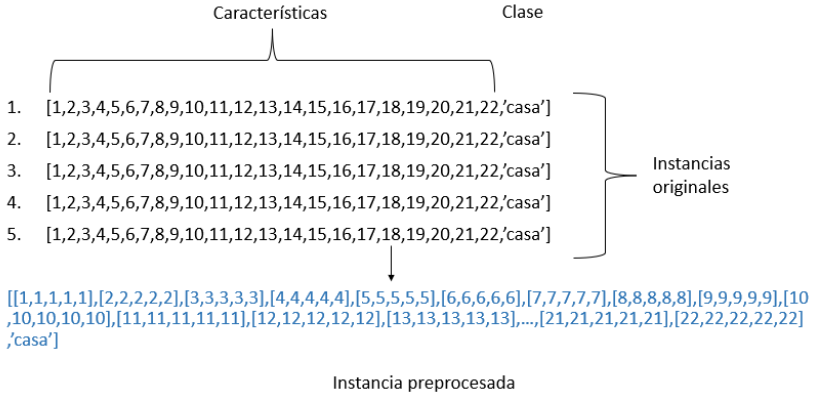


FIGURA 5.9: Preprocesamiento de los datos para tener el formato de entrada correcto para el método Transformador.

Al hacer esto, se reduce considerablemente el desbalanceo que existe por lo que en estos experimentos ya no se aplica ningún método de submuestreo. Al igual que con los experimentos recien descritos, los valores se normalizan por el método $z - score$ y se asegura que los datos estén estratificados. Los parámetros que se ocupan para la red y el proceso de entrenamiento, los cuales fueron identificados a través de forma empírica son los siguientes:

- Épocas: 150
- Tasa de aprendizaje: 0.001

- Número de cabezas en el modulo de atención: 11

Los resultados obtenidos considerando la aumentación de datos, las inferencias de las transiciones y el almacenamiento del mejor modelo fueron de una exactitud de **96.65 %** para el conjunto de entrenamiento y de un **92.48 %** en el conjunto de prueba. En la Fig. 5.10 se observa el proceso de entrenamiento a través de las épocas definidas y la exactitud y el valor de la función de pérdida.

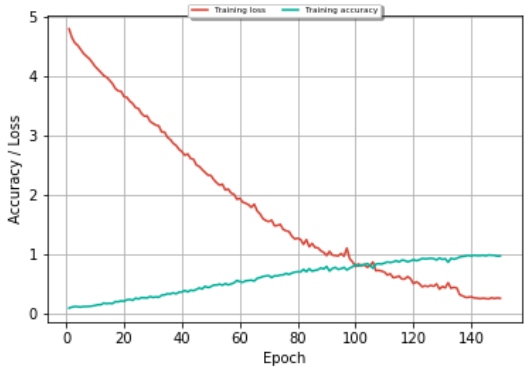


FIGURA 5.10: Proceso de entrenamiento con el método de reconocimiento Transformador.

En la Fig. 5.11 se muestra cómo se realiza el ajuste en el proceso de entrenamiento de la tasa de aprendizaje mediante el mecanismo descrito.

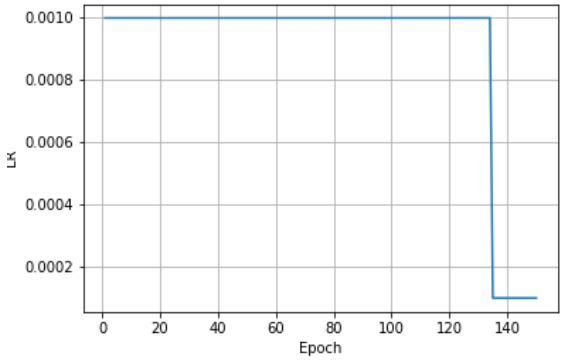


FIGURA 5.11: Proceso de ajuste de tasa de aprendizaje de forma dinámica.

A continuación se muestra en la Tabla 5.5 la comparación con el trabajo relacionado, en la misma se puede visualizar que los resultados obtenidos son competitivos por ambos métodos.

Tabla 5.5: Comparación de los resultados con trabajo relacionado con el conjunto LIBRAS.

Autor	Método	Exactitud
Trabajo propuesto	BiLSTM	96.65 %
Trabajo propuesto	Transformador	92.48 %
Amaral et al. [108]	LSTM	88.4 %
Passos et al. [109]	SVM	88.12 %

5.3.5. Resultados de la red BiLSTM en WASL

Con el conjunto de datos WASL, se realizaron únicamente experimentos con la red BiLSTM por cuestiones de tiempo, para ellos se consideran los mismos parámetros de la red empleados en los experimentos realizados con el conjunto de datos LIBRAS, los cuales son los siguientes:

- Épocas: 50
- Tasa de aprendizaje: 0.003
- Capas ocultas: 2
- Celdas en las capas ocultas: 128

Por la misma cuestión del tiempo sólo se consideraron videos referentes a 30 etiquetas, como se mencionó, restan videos referentes a 70 etiquetas más para poder establecer que los resultados están en igualdad de condiciones que el trabajo relacionado. En estos experimentos se almacena el mejor modelo como previamente se describió, así como los conjuntos de entrenamiento, validación y prueba, sin embargo, no llegó a incluirse la información referente de las transiciones. Tomando todo esto en cuenta, la Tabla 5.6 muestra los resultados obtenidos.

Tabla 5.6: Resultados en el reconocimiento de lengua de señas ocupando una red BiLSTM ocupando datos aumentados con el método de rotación en secuencia de las articulaciones y la detección de transiciones con el conjunto WASL.

Método submuestreo	Exactitud	SD
Tomek	87.48 %	±0,35
Sin submuestreo	87.24	±0,66
OSS	86.96 %	±0,35
AllKNN	81.59	±0,38

Con el fin de mostrar un punto de referencia, se listan los trabajos que tienen los mejores resultados con ese conjunto de datos en la Tabla 5.7. Es claro que conforme se vayan

agregando más clases y datos, lo más probable es que la exactitud se decremente, no obstante, observando los resultados obtenidos hasta el momento, parece que el método propuesto podría obtener finalmente resultados sólo por debajo de los trabajos de [110] y de [111]. Como trabajo futuro resta averiguar esto y además agregar la información referente a las transiciones, la cual como se observó con el conjunto de LIBRAS, podría ayudar a obtener una pequeña mejora.

Tabla 5.7: Resultados obtenidos con el conjunto de WASL ocupando 100 etiquetas que son el estado del arte.

Autores	Exactitud
Hu et al. [110]	83.30 %
Maruyama et al. [111]	81.38 %
Li et al. [105]	65.89 %
Bohavcek et al. [103]	63.18 %

5.4. Discusión

Al considerar los resultados obtenidos se retoman y responden las preguntas de investigación definidas en el Capítulo 1:

1. ¿Qué mejora de rendimiento se obtiene al hacer uso de características multimodales, así como de la correcta identificación de transiciones entre señas en el reconocimiento de lengua de señas?

Como se observa en la Tabla 5.5 los resultados sobre el conjunto de datos LIBRAS mejoran los previamente obtenidos por un margen considerable en el caso del mejor resultado.

En la Tabla 5.6 se muestran los resultados obtenidos con el conjunto WASL, dado que restan glosas (70) por considerar y que eso representa que el modelo empieza a confundir instancias, es normal esperar que los resultados desciendan. No obstante, se aprecia de forma clara como se tiene un rendimiento competitivo con el trabajo relacionado.

Además, como se mostró en el caso del conjunto de LIBRAS, el uso de la información de las transiciones entre señas ayudó a incrementar de forma nada despreciable la exactitud que había sido alcanzada sin esos datos.

Por estas razones descritas previamente, se puede concluir que se puede obtener un rendimiento competitivo mediante el uso de características multimodales y el empleo de información de transiciones entre señas como descriptores.

2. ¿Cuáles de las características extraídas proveen información relevante para el propósito de distinguir entre señas que son similares?

Como se mencionó anteriormente, las características que prevalecieron a la etapa de la reducción de dimensionalidad son las referentes a los componentes manuales, esto dado que son las que mayor información brindan.

Algunos componentes no manuales (relacionados con expresiones faciales e información de la intención de la mirada) se mantuvieron, lo que también indica que se obtiene información útil de ese tipo de características.

Finalmente, como se mostró en el rendimiento alcanzado en los experimentos finales, la información contenida en la estimación de transiciones entre señas también fue relevante.

3. ¿La identificación de transiciones entre señas puede ayudar a delimitar la duración de todas las señas presentes en una conversación compuesta de varias señas?

Aunque ésta pregunta se planteó al inicio de la investigación como una opción para hacer uso de la información de la estimación de las transiciones entre señas, al final se optó por emplear dicha información como un descriptor que ayudara a caracterizar de mejor forma las instancias relacionadas a movimientos de transiciones o estados de reposo.

Dado que la RDB demostró tener un buen rendimiento como se mostró en el apartado de la evaluación de dicho método en este Capítulo, también podría hacerse uso de dichos datos para poder segmentar las señas. Una metodología sencilla a seguir sería buscar los límites de instancias consideradas como señas y las consideradas como transiciones. Con ello se podrían descartar todas las instancias contiguas marcadas como transiciones.

4. ¿Es posible tener un mejor rendimiento en la tarea del reconocimiento de lengua de señas tomando en cuenta características basadas en la mirada, forma de labios y posición de la cabeza de un señante?

En la pregunta de investigación 2 se mencionó que uno de las fuentes de información relevante que se identificó fue la pertinente a algunos componentes no manuales.

En el Capítulo se definió que dichos componentes son relevantes en la gramática de toda lengua de señas y con ello se planteó la hipótesis de que en la tarea del reconocimiento de lenguas mediante el uso de técnicas computacionales podrían ser de utilidad.

Con datos de mayor resolución y mejores técnicas para lidiar con las oclusiones que están presentes en los datos de entrada para éste problema, seguramente, el resto de los descriptores que en primera instancia parecieron no ser útiles en éste proyecto demostrarían una mayor relevancia.

Capítulo 6

Conclusiones

El objetivo principal de este trabajo de investigación es proponer una metodología de extracción de características manuales y no manuales para el reconocimiento de lengua de señas; esta fue evaluada estadísticamente sobre la exactitud que nos dieron diferentes métodos de reconocimiento y comparado con lo que se ha reportado en la literatura, se obtuvieron resultados competitivos.

En el Capítulo 3 se elaboró una revisión de los métodos y técnicas utilizadas para la extracción y reconocimiento de lengua de señas, tanto de tipo aislado como de tipo continuo, donde se logró identificar que no han sido tan empleadas las características relacionadas con los componentes no manuales dentro de las lenguas de señas.

Además, se apreciaron otros dos hechos claves; en primer lugar la mayoría de la investigación ha sido con datos de tipo aislado (a nivel de palabra), situación entendible dado que los datos de tipo continuo conllevan una mayor complejidad, no obstante, estos últimos son los datos presentes en situaciones diarias y por ende el trabajo sobre de ellos es de vital importancia para la comunidad sorda.

En segundo lugar, los estudios recientes se han enfocado en demasía en el uso de métodos de reconocimiento que aplican aprendizaje profundo. Si bien dichos trabajos han obtenido resultados relevantes, el uso de los mismos conllevan problemas que no han de perderse de vista, como lo son: la necesidad de contar con grandes cantidades de datos, los cuales no siempre es posible tener en problemas tan específicos; la mayoría de ellos emplea los datos crudos sin ningún tipo de segmentación, lo cual aunque evitar el diseño de una procedimiento para aislar regiones de interés, al considerar toda la información presente en las imágenes o videos, también mete ruido o datos ajenos a la problemática.

En ese estudio y análisis bibliográfico se pudo determinar que los movimientos de epéntesis son un componente clave que no se ha explorado a profundidad en las investigaciones previas. Estos movimientos están relacionados con toda gesticulación que no está asociada a la realización de una seña. La oportuna y correcta identificación de estos movimientos se propuso que ayudaría a filtrar instancias no relevantes.

Tomando en cuenta lo previamente estipulado, se definió una metodología en donde se abordó cada punto, en primer lugar se realizó la identificación y segmentación de regiones de interés con las cuales se hizo la extracción de características tanto de componentes manuales como no manuales. Dentro de la misma hay un apartado donde se estiman no sólo movimientos de epéntesis sino también estados de reposo a través de una RDB, dichas inferencias al final se consideraron como descriptores en la etapa de reconocimiento. Posteriormente, en la etapa de reconocimiento se emplearon dos métodos que se identificó que han dado buenos resultados en problemas de secuencias de datos: Una red BiLSTM y un Transformador.

Dos conjuntos de datos fueron considerados: LIBRAS y WASL, el primero contiene datos de tipo continuo, por otro lado, el segundo conjunto consta de datos de tipo aislado. A pesar de que ambos conjuntos tienen un número considerable de datos siguen conteniendo un número reducido de glosas. En LIBRAS, además, al tener datos de tipo continuo se contaban con muchas instancias relacionadas con transiciones o estados de reposo. De hecho, como se mostró, el conjunto de datos no sólo estaba desbalanceado, también había muy pocas instancias de algunas glosas. Por tal razón, se agregaron dos etapas más a la metodología: la aumentación de datos considerando rotaciones sobre las articulaciones de los señantes y el submuestreo de las instancias de la clase dominante.

A pesar de que se consideró un conjunto pequeño (44) de características, como parte de la experimentación se investigó si una técnica de reducción de dimensionalidad podría ayudar a alcanzar un mejor rendimiento en la etapa de clasificación, para ello se consideró el método de PCA.

Posteriormente, se hizo el diseño de diversos experimentos y se obtuvieron resultados competitivos en la etapa de clasificación, en la tabla 5.5 se visualiza que el mejor resultado que se obtuvo (96,65 %) con la red BiLSTM en el conjunto LIBRAS superó en más de 6 % al trabajo identificado con la mejor exactitud. EL resultado de 92,48 % obtenido con el Transformador también superó al trabajo relacionado por una diferencia de 4 %.

Dado que el trabajo relacionado con LIBRAS es escaso, se optó por ocupar otro conjunto de datos (WASL); por cuestiones de tiempo no se logró completar el experimento que reporta el trabajo relacionado de WASL100, es decir, considerar 100 glosas en lugar de las 30 que se tomaron en cuenta en esta investigación. A pesar de ello los resultados

obtenidos (87,48 %) fueron de relevancia ya que demostraron que la metodología diseñada es capaz de funcionar con datos de tipo continuo y de tipo aislado sin verse afectada en su rendimiento.

Las principales conclusiones son las siguientes:

- La información extraída de componentes no manuales también aporta datos relevantes para la etapa del reconocimiento de lengua de señas. Dichos datos ayudan a diferenciar señas donde la información manual es similar.
- Los datos de las inferencias concernientes a las transiciones entre señas o estados de reposo y su posterior uso como descriptor son de utilidad para la tarea del reconocimiento. El contenido de estas inferencias ayuda a descartar (o clasificar) instancias no relacionadas con alguna seña.
- Una matriz de características con una dimensionalidad pequeña en comparación con aquellas que emplean enfoques de aprendizaje profundo demostró ser más que suficiente para obtener resultados competitivos. Un espacio de características compacto no sólo es una ventaja a la hora de la extracción de características en cuanto al tiempo computacional requerido sino también en la etapa del entrenamiento de los modelos de reconocimiento.

6.1. Trabajo Futuro

Como trabajo futuro hay muchos ajustes que se pueden realizar con el propósito de hacer más sólidos los resultados obtenidos, en primer lugar y de forma obvia hay que culminar los experimentos para el conjunto de datos WASL, por lo menos los pertenecientes al subconjunto WASL100. Los resultados que se obtuvieron fueron alentadores para mostrar la robustez de la metodología presentada en el uso de datos aislados y continuos, no obstante, la total consideración de los datos de WASL100 darán una comparación equitativa con el trabajo relacionado.

Por otro lado se pueden considerar nuevos métodos para la aumentación de datos, etapa que terminó siendo clave para la obtención de los resultados finales. Enfoques basados en la simulación de pequeñas variaciones en las regiones de los dedos o las expresiones faciales podría ser explorados.

Dado que la etapa de la reducción de dimensionalidad también fue relevante, métodos alternos a PCA podrían ser explorados. Un método que genera igual proyecciones de tipo lineal al igual que PCA es el Escalado Multidimensional. Además de ellos, dado que

es difícil que haya una linealidad en el espacio de características, enfoques de proyección no lineales como el Mapeo de Sammon o IsoMap también serían opciones interesantes de analizar.

Finalmente, el uso de más métodos de reconocimiento podrían brindar buenos resultados. En la investigación se hizo uso de una adaptación del Transformador definido originalmente en [35], sin embargo, el método de Transformador de Visión [104] tal vez sería una opción más idónea por los datos de entrada con los que se trabaja. Otro enfoque que ha sido últimamente ocupado para problemas de reconocimiento y que podría estudiarse son las CapsNet [5, 6, 17].

6.2. Publicaciones

Los siguientes trabajos son productos logrados durante el desarrollo de este trabajo de investigación:

- Sánchez-Ruiz, D., Olvera-López, J. A. & Olmos-Pineda, I. Reconocimiento de lengua de señas como medio para un mundo más inclusivo. Contactos, Revista De Educación en Ciencias E Ingeniería, (120), 35-46, 2021.
- Sánchez-Ruiz, D., Olvera-López, J. A. & Olmos-Pineda, I. Metodología y avances para el Reconocimiento Continuo de Lengua de Señas. Lenguaje, conocimiento, y tecnología educativa: avances recientes, 2021. ISBN: 978-607-525-761-7.
- Sánchez-Ruiz, D., Olvera-López, J. A. & Olmos-Pineda, I. Sign Language Recognition through Manual and Non-Manual Features. Research in Computing Science Journal, 151 (12), 2022. ISSN 1870-4069.
- Sánchez-Ruiz, D., Olvera-López, J. A. & Olmos-Pineda, I. Word-Level Sign Language Recognition via Handcrafted Features. IEEE Latin America Transactions, 21 (7), 839-848, 2023.

Referencias

[1] WHO. Sordera y pérdida de la audición, 2021. URL https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[2] Roland Pfau, Markus Steinbach, and Bencie Woll. *Sign language: An international handbook*, volume 37. Walter de Gruyter, 2012.

[3] CONAPRED. Lengua mexicana de señas, 2017.

[4] Suhail Muhammad Kamal, Yidong Chen, Shaozi Li, Xiaodong Shi, and Jiangbin Zheng. Technical Approaches to Chinese Sign Language Processing: A Review. *IEEE Access*, 7:96926–96935, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2929174. URL https://ieeexplore.ieee.org/document/8764391/.

[5] Ankita Wadhawan and Parteek Kumar. Sign Language Recognition Systems: A Decade Systematic Literature Review. *Archives of Computational Methods in Engineering*, (0123456789), dec 2019. ISSN 1134-3060. doi: 10.1007/s11831-019-09384-2. URL https://doi.org/10.1007/s11831-019-09384-2http://link.springer.com/10.1007/s11831-019-09384-2.

[6] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Sign language recognition: A deep survey. *Expert Systems with Applications*, page 113794, 2020.

[7] Katrin Renz, Nicolaj C Stache, Samuel Albanie, and Gül Varol. Sign language segmentation with temporal convolutional networks. *arXiv preprint arXiv:2011.12986*, 2020.

[8] Katrin Renz, Nicolaj C Stache, Neil Fox, Gül Varol, and Samuel Albanie. Sign segmentation with changepoint-modulated pseudo-labelling. *arXiv preprint arXiv:2104.13817*, 2021.

[9] Ananya Choudhury, Anjan Kumar Talukdar, Manas Kamal Bhuyan, and Kandarpa Kumar Sarma. Movement Epenthesis Detection for Continuous Sign Language

Recognition. *Journal of Intelligent Systems*, 26(3):471–481, 2017. ISSN 03341860. doi: 10.1515/jisys-2016-0009.

[10] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, and Tessa Verhoef. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 16–31, 2019.

[11] Ronice Müller Quadros, Deonísio Schmitt, Juliana Lohn, and Tarcísio de Arantes Leite. Corpus de libras, 2012.

[12] WHO. *World Report on Hearing*. 2021. ISBN 978-92-4-002048-1.

[13] World Federation of Deaf. Our work. https://wfdeaf.org/our-work, 2021. URL https://wfdeaf.org/our-work.

[14] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudrealt, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, aug 2019.

[15] R. Elakkiya. Machine learning based sign language recognition: a review and its research frontier. *Journal of Ambient Intelligence and Humanized Computing*, (0123456789), 2020. ISSN 18685145. doi: 10.1007/s12652-020-02396-y. URL https://doi.org/10.1007/s12652-020-02396-y.

[16] Mark Borg and Kenneth P. Camilleri. Sign Language Detection "in the Wild" with Recurrent Neural Networks. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1637–1641. IEEE, may 2019. ISBN 978-1-4799-8131-1. doi: 10.1109/ICASSP.2019.8683257. URL https://ieeexplore.ieee.org/document/8683257/.

[17] Oscar Koller. Quantitative survey of the state of the art in sign language recognition. *arXiv preprint arXiv:2008.09918*, 2020.

[18] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel Transformers for Multi-articulatory Sign Language Translation. pages 1–18, 2020. URL http://arxiv.org/abs/2009.00299.

[19] Nikolaos Arvanitis, Constantinos Constantinopoulos, and Dimitrios Kosmopoulos. Translation of sign language glosses to text using sequence-to-sequence attention models. *Proceedings - 15th International Conference on Signal Image Technology*

and Internet Based Systems, SISITS 2019, pages 296–302, 2019. doi: 10.1109/SITIS.2019.00056.

[20] Franco Ronchetti. *Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas*. PhD thesis, Facultad de Informática, 2017.

[21] Jörg Zieren, Ulrich Canzler, Britta Bauer, and Karl-Friedrich Kraiss. Sign Language Recognition. In *Advanced Man-Machine Interaction*, number 231135, pages 95–139. Springer-Verlag, Berlin/Heidelberg, 2006. ISBN 1424407796. doi: 10.1007/3-540-30619-6_3.

[22] Edward S Klima and Ursula Bellugi. *The signs of language*. Harvard University Press, 1979.

[23] Clayton Valli and Ceil Lucas. *Linguistics of American sign language: An introduction*. Gallaudet University Press, 2000.

[24] Scott K Liddell and Robert E Johnson. American sign language: The phonological base. *Sign language studies*, 64(1):195–277, 1989.

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[26] Krishnendu Kar. *Mastering Computer Vision with TensorFlow 2.x: Build advanced computer vision applications using machine learning and deep learning techniques*. Packt Publishing, 2020.

[27] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for perceiving and processing reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019.

[28] Ankur Ankan and Abinash Panda. *Mastering probabilistic graphical models using python*. Packt Publishing Ltd, 2015.

[29] Luis Sucar and Miriam Martínez-Arroyo. Aprendizaje de clasificadores bayesianos dinámicos. 2011.

[30] Heung-Il Suk, Bong-Kee Sin, and Seong-Whan Lee. Recognizing hand gestures using dynamic bayesian network. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6. IEEE, 2008.

[31] Ara V Nefian, Luhong Liang, Xiaobo Pi, Xiaoxing Liu, and Kevin Murphy. Dynamic bayesian networks for audio-visual speech recognition. *EURASIP Journal on Advances in Signal Processing*, 2002(11):1–15, 2002.

[32] Habes Alkhraisat and Saqer Alshrah. American sign language pattern recognition based on dynamic bayesian network. *International Journal of Computer Science and Information Security (IJCSIS)*, 14(3), 2016.

[33] David Foster. *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media, 2019.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] Uday Kamath, Kenneth Graham, and Wael Emara. *Transformers for Machine Learning: A Deep Dive*. Chapman and Hall/CRC, 2022.

[37] Cory Maklin. Transformers explained, 2022. URL https://medium.com/@corymaklin/transformers-explained-610b2f749f43.

[38] Marwa Elpeltagy, Moataz Abdelwahab, Mohamed E. Hussein, Amin Shoukry, Asmaa Shoala, and Moustafa Galal. Multi-modality-based Arabic sign language recognition. *IET Computer Vision*, 12(7):1031–1039, 2018. ISSN 17519640. doi: 10.1049/iet-cvi.2017.0598.

[39] Yuancheng Ye, Yingli Tian, Matt Huenerfauth, and Jingya Liu. Recognizing american sign language gestures from within continuous videos. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June: 2145–2154, 2018. ISSN 21607516. doi: 10.1109/CVPRW.2018.00280.

[40] Wenjin Tao, Ming C. Leu, and Zhaozheng Yin. American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence*, 76 (September):202–213, 2018. ISSN 09521976. doi: 10.1016/j.engappai.2018.09.006. URL https://doi.org/10.1016/j.engappai.2018.09.006.

[41] Shujun Zhang, Weijia Meng, Hui Li, and Xuehong Cui. Multimodal spatiotemporal networks for sign language recognition. *IEEE Access*, 7:180270–180280, 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2959206.

[42] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring Cross-Domain Knowledge for Video Sign Language Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6204–6213. IEEE, jun 2020. ISBN 978-1-7281-7168-5. doi: 10.1109/CVPR42600.2020.00624. URL https://ieeexplore.ieee.org/document/9157542/.

[43] Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. A deep learning approach for analyzing video and skeletal features in sign language recognition. *IST 2018 - IEEE International Conference on Imaging Systems and Techniques, Proceedings*, pages 1–6, 2018. doi: 10.1109/IST.2018.8577085.

[44] R. Elakkiya and V. Vanitha. Interactive real time fuzzy class level gesture similarity measure based sign language recognition using artificial neural networks. *Journal of Intelligent and Fuzzy Systems*, 37(5):6855–6864, 2019. ISSN 18758967. doi: 10.3233/JIFS-190707.

[45] P. K. Athira, C. J. Sruthi, and A. Lijiya. A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario. *Journal of King Saud University - Computer and Information Sciences*, (xxxx):0–10, 2019. ISSN 22131248. doi: 10.1016/j.jksuci.2019.05.002. URL https://doi.org/10.1016/j.jksuci.2019.05.002.

[46] Pradeep Kumar, Partha Pratim Roy, and Debi Prosad Dogra. Independent Bayesian classifier combination based sign language recognition using facial expression. *Information Sciences*, 428:30–48, 2018. ISSN 00200255. doi: 10.1016/j.ins.2017.10.046.

[47] Xiang Ma, Lin Yuan, Ruoshi Wen, and Qiang Wang. Sign language recognition based on concept learning. *I2MTC 2020 - International Instrumentation and Measurement Technology Conference, Proceedings*, pages 1–6, 2020. doi: 10.1109/I2MTC43012.2020.9128734.

[48] Marco Fagiani, Emanuele Principi, Stefano Squartini, and Francesco Piazza. Signer independent isolated Italian sign recognition based on hidden Markov models. *Pattern Analysis and Applications*, 18(2):385–402, 2015. ISSN 14337541. doi: 10.1007/s10044-014-0400-z.

[49] Vincent Hernandez, Tomoya Suzuki, and Gentiane Venture. Convolutional and recurrent neural network for human activity recognition: Application on American sign language. *PLOS ONE*, 15(2):e0228869, feb 2020. ISSN 1932-6203. doi: 10.1371/journal.pone.0228869.

[50] Tomasz Kapuscinski and Marian Wysocki. Recognition of signed expressions in an experimental system supporting deaf clients in the city office. *Sensors (Switzerland)*, 20(8), 2020. ISSN 14248220. doi: 10.3390/s20082190.

[51] Anil Osman Tur and Hacer Yalim Keles. Evaluation Of Hidden Markov Models Using Deep CNN Features In Isolated Sign Recognition. pages 1–16, jun 2020. URL https://arxiv.org/abs/2006.11183http://arxiv.org/abs/2006.11183.

[52] T. Raghuveera, R. Deepthi, R. Mangalashri, and R. Akshaya. A depth-based Indian Sign Language recognition using Microsoft Kinect. *Sadhana - Academy Proceedings in Engineering Sciences*, 45(1), 2020. ISSN 09737677. doi: 10.1007/s12046-019-1250-6. URL https://doi.org/10.1007/s12046-019-1250-6.

[53] Hezhen Hu, Wengang Zhou, Junfu Pu, and Houqiang Li. Global-local Enhancement Network for NMFs-aware Sign Language Recognition. 1(1):1–18, 2020. URL http://arxiv.org/abs/2008.10428.

[54] Nikolaos M Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, and Petros Daras. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, 2021.

[55] Chengcheng Wei, Wengang Zhou, Junfu Pu, and Houqiang Li. Deep Grammatical Multi-classifier for Continuous Sign Language Recognition. *International Conference on Multimedia Big Data (BigMM)*, pages 435–442, 2019. doi: 10.1109/BigMM.2019.00027.

[56] Al Amin Hosain, Panneer Selvam Santhalingam, Parth Pathak, Jana Kosecka, and Huzefa Rangwala. Sign Language Recognition Analysis using Multimodal Data. sep 2019. URL http://arxiv.org/abs/1909.11232.

[57] Ahlet Alp Kindiroglu, Ogulcan Ozdemir, and Lale Akarun. Temporal Accumulative Features for Sign Language Recognition. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[58] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015. ISSN 1090235X. doi: 10.1016/j.cviu.2015.09.013. URL http://dx.doi.org/10.1016/j.cviu.2015.09.013.

[59] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via

Hybrid CNN-HMMs. *International Journal of Computer Vision*, 126(12): 1311–1325, dec 2018. ISSN 0920-5691. doi: 10.1007/s11263-018-1121-3. URL https://doi.org/10.1007/s11263-018-1121-3http://link.springer.com/10.1007/s11263-018-1121-3.

[60] R. Elakkiya and K. Selvamani. Subunit sign modeling framework for continuous sign language recognition. *Computers and Electrical Engineering*, 74: 379–390, 2019. ISSN 00457906. doi: 10.1016/j.compeleceng.2019.02.012. URL https://doi.org/10.1016/j.compeleceng.2019.02.012.

[61] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 2257–2264, 2018.

[62] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9):2822–2832, sep 2019. ISSN 1051-8215. doi: 10.1109/TCSVT.2018.2870740. URL https://ieeexplore.ieee.org/document/8466903/.

[63] Junfu Pu, Wengang Zhou, and Houqiang Li. Dilated Convolutional Network with Iterative Optimization for Continuous Sign Language Recognition. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 885–891, California, jul 2018. International Joint Conferences on Artificial Intelligence Organization. ISBN 9780999241127. doi: 10.24963/ijcai.2018/123. URL https://www.ijcai.org/proceedings/2018/123.

[64] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative Alignment Network for Continuous Sign Language Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4165–4174, 2019.

[65] Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai. Fully convolutional networks for continuous sign language recognition, 2020.

[66] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10020–10030. IEEE, jun 2020. ISBN 978-1-7281-7168-5. doi: 10.1109/CVPR42600.2020.01004. URL https://ieeexplore.ieee.org/document/9156773/.

[67] Yunus Can Bilge, Nazli Ikizler-Cinbis, and Ramazan Gokberk Cinbis. Zero-Shot Sign Language Recognition: Can Textual Data Uncover Sign Languages? pages 1–14, jul 2019. URL http://arxiv.org/abs/1907.10292.

[68] Hao Zhou, Wengang Zhou, and Houqiang Li. Dynamic pseudo label decoding for continuous sign language recognition. *Proceedings - IEEE International Conference on Multimedia and Expo*, 2019-July:1282–1287, 2019. ISSN 1945788X. doi: 10.1109/ICME.2019.00223.

[69] Yanqiu Liao, Pengwen Xiong, Weidong Min, Weiqiong Min, and Jiahao Lu. Dynamic Sign Language Recognition Based on Video Sequence with BLSTM-3D Residual Networks. *IEEE Access*, 7:38044–38054, 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2904749.

[70] Ali Asghar Zare and Seyed Hamid Zahiri. Recognition of a real-time signer-independent static Farsi sign language based on fourier coefficients amplitude. *International Journal of Machine Learning and Cybernetics*, 9(5):727–741, 2018. ISSN 1868808X. doi: 10.1007/s13042-016-0602-3.

[71] Sunitha Ravi, Maloji Suman, P. V.V. Kishore, Kiran Kumar E, Teja Kiran Kumar M, and Anil Kumar D. Multi modal spatio temporal co-trained CNNs with single modal testing on RGB–D based sign language gesture recognition. *Journal of Computer Languages*, 52(April):88–102, 2019. ISSN 25901184. doi: 10.1016/j.cola.2019.04.002. URL https://doi.org/10.1016/j.cola.2019.04.002.

[72] Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. Neural Sign Language Translation Based on Human Keypoint Estimation. *Applied Sciences*, 9(13):2683, jul 2019. ISSN 2076-3417. doi: 10.3390/app9132683. URL https://www.mdpi.com/2076-3417/9/13/2683.

[73] Amit Moryossef, Ioannis Tsochantaridis, Roee Aharoni, Sarah Ebling, and Srini Narayanan. Real-Time Sign Language Detection using Human Pose Estimation. pages 1–13, aug 2020. URL http://arxiv.org/abs/2008.04637.

[74] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Adversarial Training for Multi-Channel Sign Language Production. aug 2020. URL http://arxiv.org/abs/2008.12405.

[75] Qinkun Xiao, Minying Qin, and Yuting Yin. Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks*, 125:41–55, 2020. ISSN 18792782. doi: 10.1016/j.neunet.2020.01.030. URL https://doi.org/10.1016/j.neunet.2020.01.030.

[76] Mohammed Mustafa. A study on Arabic sign language recognition for differently abled using advanced machine learning classifiers. *Journal of Ambient Intelligence and Humanized Computing*, (0123456789), 2020. ISSN 18685145. doi: 10.1007/s12652-020-01790-w. URL https://doi.org/10.1007/s12652-020-01790-w.

[77] Heike Brock, Felix Law, Kazuhiro Nakadai, and Yuji Nagashima. Learning Three-dimensional Skeleton Data from Sign Language Video. *ACM Transactions on Intelligent Systems and Technology*, 11(3), 2020. ISSN 21576912. doi: 10.1145/3377552.

[78] Nicholas Wilkins, Beck Cordes Galbraith, and Ifeoma Nwogu. Modeling Global Body Configurations in American Sign Language. pages 2–6, 2020. URL http://arxiv.org/abs/2009.01468.

[79] Kiran Kumar, PVV Kishore, Teja Kiran Kumar, and Anil Kumar. 3d sign language recognition with joint distance and angular coded color topographical descriptor on a 2–stream cnn. *Neurocomputing*, 372:40–54, 2020.

[80] Ashwini M. Deshpande and Snehal R. Kalbhor. Video-based marathi sign language recognition and text conversion using convolutional neural network. *Lecture Notes in Electrical Engineering*, 569:761–773, 2020. ISSN 18761119.

[81] Walaa Aly, Saleh Aly, and Sultan Almotairi. User-Independent American Sign Language Alphabet Recognition Based on Depth Image and PCANet Features. *IEEE Access*, 7:123138–123150, 2019. doi: 10.1109/access.2019.2938829.

[82] Soraia Silva Prietch, Polianna dos Santos Paim, Ivan Olmos-Pineda, Josefina Guerrero García, and Juan Manuel Gonzalez Calleros. The human and the context components in the design of automatic sign language recognition systems. In *Iberoamerican Workshop on Human-Computer Interaction*, pages 369–380. Springer, 2019.

[83] Bernat Requena Serra. Muestreo sistemático, Oct 2020. URL https://www.universoformulas.com/estadistica/inferencia/muestreo-sistematico/.

[84] CVAT.ai Corporation. Computer vision annotation tool (cvat), November 2023. URL https://doi.org/10.5281/zenodo.10076023.

[85] Go from raw images to a trained computer vision model in minutes. URL https://roboflow.com/.

[86] Brett Drury, Jorge Valverde-Rebaza, Maria-Fernanda Moura, and Alneu de Andrade Lopes. A survey of the applications of bayesian networks in agriculture. *Engineering Applications of Artificial Intelligence*, 65:29–42, 2017.

[87] Lijun Sun and Alexander Erath. A bayesian network approach for population synthesis. *Transportation Research Part C: Emerging Technologies*, 61:49–62, 2015.

[88] Zhichao Tian, Binghui Si, Xing Shi, and Zigeng Fang. An application of bayesian network approach for selecting energy efficient hvac systems. *Journal of Building Engineering*, 25:100796, 2019.

[89] Amal Bakshan, Issam Srour, Ghassan Chehab, Mutasem El-Fadel, and Jalal Karaziwan. Behavioral determinants towards enhancing construction waste management: A bayesian network analysis. *Resources, Conservation and Recycling*, 117: 274–284, 2017.

[90] K Leerojanaprapa, W Atthirawong, W Aekplakorn, and K Sirikasemsuk. Applying bayesian network for noncommunicable diseases risk analysis: Implementing national health examination survey in thailand. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 904–908. IEEE, 2017.

[91] Ghada Trabelsi, Philippe Leray, Mounir Ben Ayed, and Adel Mohamed Alimi. Dynamic mmhc: A local search algorithm for dynamic bayesian network structure learning. In *International Symposium on Intelligent Data Analysis*, pages 392–403. Springer, 2013.

[92] Roderick JA Little and Mark D Schluchter. Maximum likelihood estimation for mixed continuous and categorical data with missing values. *Biometrika*, 72(3): 497–512, 1985.

[93] Sinan Ozdemir and Divya Susarla. *Feature Engineering Made Easy: Identify unique features from your dataset in order to build powerful machine learning systems.* Packt Publishing Ltd, 2018.

[94] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[95] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 59–66. IEEE, 2018.

[96] Amir Zadeh, Yao Chong Lim, Tadas Baltrusaitis, and Louis-Philippe Morency. Convolutional experts constrained local model for 3d facial landmark detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2519–2528, 2017.

[97] Shiqi Li, Chi Xu, and Ming Xie. A robust o (n) solution to the perspective-n-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 34(7): 1444–1450, 2012.

[98] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390. IEEE, 2011.

[99] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 354–361, 2013.

[100] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.

[101] Junfu Pu, Wengang Zhou, Hezhen Hu, and Houqiang Li. Boosting continuous sign language recognition via cross modality augmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1497–1505, 2020.

[102] Chan-Il Park and Chae-Bong Sohn. Data augmentation for human keypoint estimation deep learning based sign language translation. *Electronics*, 9(8):1257, 2020.

[103] Matyáš Boháček and Marek Hrúz. Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 182–191, 2022.

[104] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[105] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469, 2020.

[106] Thomas Minka. Automatic choice of dimensionality for pca. *Advances in neural information processing systems*, 13, 2000.

[107] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[108] Lucas Amaral, Victor Ferraz, Tiago Vieira, and Thales Vieira. Skelibras: A large 2d skeleton dataset of dynamic brazilian signs. In *Iberoamerican Congress on Pattern Recognition*, pages 184–193. Springer, 2021.

[109] Wesley L Passos, Gabriel M Araujo, Jonathan N Gois, and Amaro A de Lima. A gait energy image-based system for brazilian sign language recognition. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(11):4761–4771, 2021.

[110] Hezhen Hu, Weichao Zhao, Wengang Zhou, Yuechen Wang, and Houqiang Li. Signbert: pre-training of hand-model-aware representation for sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11087–11096, 2021.

[111] Mizuki Maruyama, Shuvozit Ghose, Katsufumi Inoue, Partha Pratim Roy, Masakazu Iwamura, and Michifumi Yoshioka. Word-level sign language recognition with multi-stream neural networks focusing on local regions. *arXiv preprint arXiv:2106.15989*, 2021.

Apéndice A

Entrenamiento de YOLOv5

La versión de YOLOv5 que fue empleada para el proyecto viene pre-entrenada con un conjunto de datos de diversos objetos, sin embargo, entre ellos no vienen manos y mucho menos con las características que se requieren (oclusiones, posición y velocidad de los objetos en la región de interés) para los conjuntos de datos LIBRAS y WASL. Es por ello que como se especificó en el Capítulo 4, se hizo el entrenamiento de un subconjunto de imágenes que fue adquirido de forma sistemática, proceso que igual fue especificado en el mismo Capítulo. En el siguiente Apéndice se enuncia el proceso seguido para la obtención del modelo de YOLOv5 para la identificación de la región de las manos.

Crear un modelo personalizado de YOLOv5 para detectar sus objetos es un proceso iterativo de recopilar y organizar imágenes, etiquetar los objetos de interés, entrenar el modelo, implementarlo y como paso opcional se puede ocupar ese modelo implementado para recopilar ejemplos de casos extremos para repetir y mejorar todo el proceso.

A.1. Crear conjunto de datos

Los modelos YOLOv5 deben entrenarse con datos etiquetados para poder aprender clases de objetos en esos datos. Hay dos opciones para crear su conjunto de datos antes de comenzar a entrenar:

- Ocupar la plataforma Roboflow [85] para crear el conjunto de datos con formato YOLO
- Hacer todo el proceso de forma manual

En este caso, se optó por la primera opción ya que la plataforma Roboflow simplifica bastante todo el proceso.

A.1.1. Recolectar imágenes

El modelo aprenderá con diversos ejemplos, por lo que tiene que ser entrenado con imágenes similares a las que verá en el contexto del problema para el que será empleado. Lo ideal es recopilar una amplia variedad de imágenes de la misma configuración (cámara, ángulo, iluminación, etc.). Dicho paso fue logrado en esta investigación mediante la creación de un conjunto de imágenes que fueron seleccionados de forma sistemática del conjunto de datos LIBRAS.

A.1.2. Crear las etiquetas

Una vez que se han recopilado las imágenes que serán empleadas para el entrenamiento, se deben de anotar los objetos de interés para crear una verdad fundamental (*Ground Truth*) de la que pueda aprender el modelo a entrenar. Para este punto se ocupa la herramienta de código abierto CVAT [84], con dicha herramienta se pueden delimitar los cuadros envolventes de las regiones de interés.

Una vez que se hace la anotación de los cuadros, dichas anotaciones se exportan con el formato de YOLO. Este formato contiene un archivo de texto por imagen (que contiene las anotaciones y una representación numérica de la etiqueta) y un mapa de etiquetas que asigna los ID numéricos a cadenas legibles por humanos. Las anotaciones están normalizadas para que se encuentren dentro del rango [0, 1], lo que hace que sea más fácil trabajar con ellas incluso después de escalar o estirar las imágenes.

A.1.3. Preparar el conjunto de datos para YOLOv5

Ya sea que se etiqueten las imágenes con Roboflow o no, la plataforma se puede usar para convertir el conjunto de datos al formato YOLO, crear un archivo de configuración YAML YOLOv5 y alojarlo para importarlo a un script de entrenamiento.

Para realizar eso se necesita cargar el conjunto de datos en un espacio de trabajo público en la plataforma de Roboflow, etiquetar las imágenes sin anotaciones y luego generar y exportar una versión del conjunto de datos en formato YOLOv5 Pytorch.

YOLOv5 realiza aumento de datos en línea durante el entrenamiento, por lo que no recomendamos aplicar ningún paso de aumentación en Roboflow para entrenar con YOLOv5. Sin embargo, se recomienda aplicar los siguientes pasos de preprocesamiento (Fig. A.1):

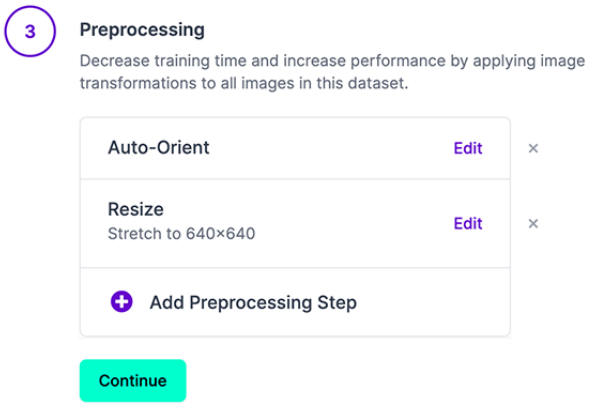


FIGURA A.1: Preprocesamiento de datos en Roboflow.

- Orientación automática: para eliminar la orientación EXIF (Orientación de la cámara en relación con la escena capturada) de sus imágenes.
- Cambiar tamaño (Estirar): al tamaño de entrada cuadrado de su modelo (640x640 es el valor predeterminado de YOLOv5).

Generar una versión brindará una instantánea de un momento determinado del conjunto de datos para que siempre se pueda regresar y comparar las futuras ejecuciones de entrenamiento del modelo con él, incluso si agrega más imágenes o cambia su configuración más adelante.

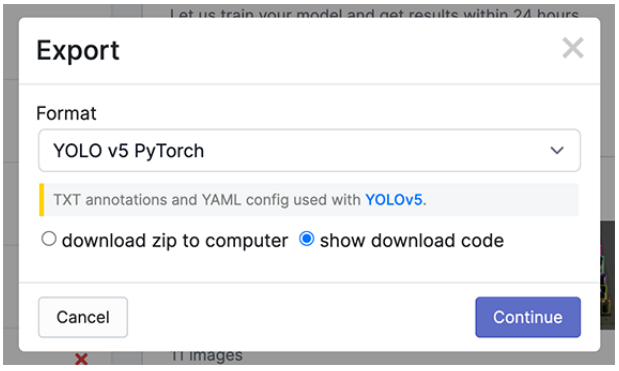


FIGURA A.2: Exportación del conjunto de datos en el formato YOLOv5.

Finalmente, hay que exportar en formato YOLOv5 Pytorch, luego, copiar el fragmento en el script de entrenamiento (Fig. A.2) o la libreta para descargar el conjunto de datos (Fig. A.3).

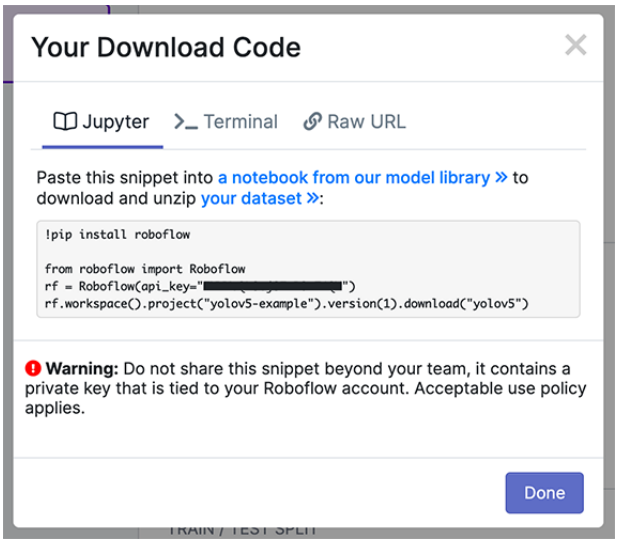


FIGURA A.3: Descarga del código en una libreta de trabajo.

A.2. Selección del modelo

Una vez que se cuentan con los datos en el formato requerido, se tiene que seleccionar un modelo previamente entrenado para comenzar a entrenar. Se puede consultar la tabla README en el repositorio[1] del framework YOLOv5 para obtener una comparación completa de todos los modelos (Ver Fig. A.4). Aquí seleccionamos YOLOv5s, el segundo modelo más pequeño y rápido disponible.

FIGURA A.4: Modelos disponibles para el entrenamiento del framework YOLOv5.

A.3. Entrenamiento

Para este momento, se procede a realizar el entrenamiento del modelo YOLOv5s especificando el conjunto de datos que se exportó previamente, el tamaño del lote, el tamaño de la imagen y –weights yolov5s.pt previamente entrenado (recomendado) o –weights ” –cfg yolov5s.yaml inicializado aleatoriamente (no recomendado).

[1] https://github.com/ultralytics/yolov5#pretrained-checkpoints

Es común que se clone el proyecto del framework YOLOv5 en una libreta de trabajo en Google Colab, se puede tomar de referencia la que brindan de referencia los desarrolladores[2]. Aunado a las parámetros que solicita el script de entrenamiento hay que subir el mismo espacio de trabajo el archivo que contiene los datos con los cuales entrenar el framework.

Durante el proceso de entrenamiento, los pesos previamente entrenados se descargan automáticamente desde la última versión de YOLOv5. A continuación se muestra un ejemplo del entrenamiento por tres épocas en el conjunto de datos en LIBRAS.

```
$ python train.py --img 640 --batch 16 --epochs 3 --data libras.yaml
--weights yolov5s.pt
```

El archivo de resultados results.csv se actualiza después de cada época y luego se representa como results.png (a continuación) una vez finalizado el entrenamiento. También se pueden graficar los resultados de cualquier archivo results.csv manualmente:

```
from utils.plots import plot_results
plot_results('path/to/results.csv')
```

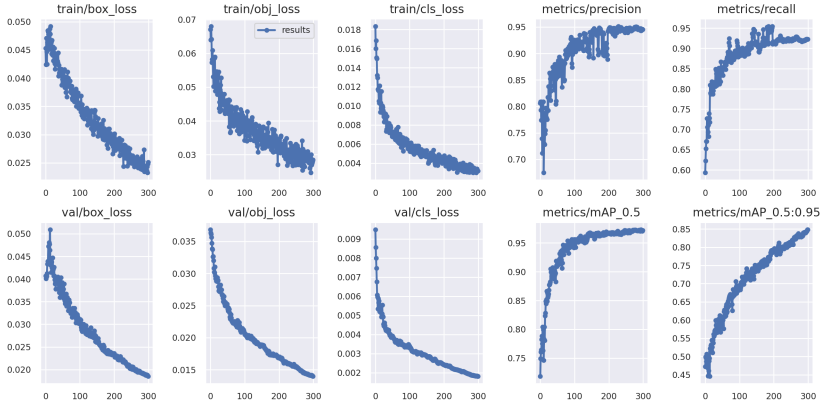


FIGURA A.5: Imagen que muestra los resultados del entrenamiento.

Por último, el framework genera un archivo que tiene los pesos de la red neuronal profunda, el cual es el modelo entrenado y que es ocupado para la etapa de las detecciones.

Para poder cargar el archivo sin tener que realizar todo el proceso del entrenamiento se realiza lo siguiente:

[2] https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb

```
# YOLOv5 PyTorch HUB Inference
import torch
model = torch.hub.load('ultralytics/yolov5', 'libras.pt',
force_reload=True, trust_repo=True)
im = 'libras1.jpg'
results = model(im)  # inferencia
results.print()
```