Universidad Nacional Autónoma de México

Facultad de Ciencias

Caracterización analítica de la estructura morfológica de la mano para el aprendizaje automático

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

Licenciado en Ciencias de la Computación

PRESENTA:

Mauricio Riva Palacio Orozco

DIRECTOR DE TESIS:

Dr. Oscar Alejandro Esquivel Flores

Facultad de Ciencias, Ciudad Universitaria, 2024

1. Datos del alumno
Riva Palacio
Orozco
Mauricio
Universidad Nacional Autónoma de México
Facultad de Ciencias
Ciencias de la Computación
316666343

2. Datos del tutor
Dr. Oscar Alejandro
Esquivel
Flores

3. Datos del sinodal 1
Dr. José de Jesús
Galaviz
Casas

4. Datos del sinodal 2
Dr. Darío Emmanuel
Vázquez
Ceballos

5. Datos del sinodal 3
M.C.I.C. Odín Miguel
Escorza
Soria

6. Datos del sinodal 4
Dr. Jorge Luis
Ortega
Arjona

7. Datos del trabajo escrito
Caracterización analítica de la estructura morfológica de la mano para el aprendizaje automático
Facultad de Ciencias
2024

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que me han apoyado durante la realización de la licenciatura. Su apoyo y contribución han sido de gran importancia para el éxito de este proyecto.

En primer lugar, agradezco a mis padres y a mi hermana por su constante apoyo y motivación a lo largo de mi carrera. Su confianza en mí y su apoyo emocional han sido fundamentales para superar los desafíos que se presentaron en este camino.

También quiero agradecer a mis amigos, quienes estuvieron a mi lado y me brindaron su apoyo incondicional. Su amistad y aliento han sido un gran impulso para seguir adelante.

Mi agradecimiento se extiende a mis profesores de la carrera en Ciencias de la Computación por compartir sus conocimientos y orientación durante mi formación académica. Su dedicación y pasión por enseñar han dejado una huella duradera en mi desarrollo profesional.

Agradezco a la Facultad de Ciencias de la UNAM por brindarme la oportunidad de cursar mis estudios en un entorno académico excepcional. Agradezco a todos los miembros

de la facultad y al personal administrativo por su labor y compromiso con la educación de calidad.

También agradezco a mi asesor, Oscar Esquivel, por su orientación experta y valiosos aportes en el desarrollo de este trabajo de tesis. Su guía ha sido fundamental para llevar a cabo este proyecto de manera exitosa.

"Tlahuizcalpan"

Lugar donde nace la luz

Resumen

Este trabajo presenta una propuesta para la extracción de características específicas de datos obtenidos de imágenes mediante software de visión por computadora, con el objetivo de realizar aprendizaje automático sin depender de grandes conjuntos de datos, centrándose en la clasificación de gestos de la mano. Se emplearon herramientas como curvas Bezier, explorando sus propiedades de curvatura, rotación y longitud para esta extracción.

La propuesta fue evaluada mediante dos entrenamientos de modelos de aprendizaje automático. En el primero, donde se consideraron conjuntos de gestos con características similares, se logró una precisión promedio del 75 %. En el segundo, que adoptó un enfoque uno a uno para cada gesto, la precisión fue desde el 60 % hasta el 100 % en casos específicos.

Este estudio plantea la posibilidad de explorar nuevas herramientas para la extracción de características, con el fin de mejorar la precisión del aprendizaje automático sin depender exclusivamente de grandes conjuntos de datos.

Índice general

Índice de figuras

Capítulo 1

Introducción

La interacción persona-computadora, que abarca el estudio, diseño, implementación y análisis de hardware y software que facilitan la comunicación entre los usuarios finales y los sistemas informáticos, ha experimentado avances significativos desde su concepción en la década de 1980. Una de las principales áreas de investigación es el reconocimiento de gestos de la mano, un campo que tiene aplicaciones significativas en la interpretación del lenguaje de señas y el control de dispositivos a distancia. Esta tesis tiene como objetivo profundizar en este campo, específicamente en la caracterización analítica de la estructura morfológica de la mano para el aprendizaje automático.

La motivación de este estudio surge del deseo de superar los desafíos inherentes al modelado computacional del cuerpo humano, específicamente la mano. La mano representa un modelo complejo debido a su estructura intrincada y la gran variedad de gestos que puede realizar. A pesar de los avances en tecnología, la precisión en la detección y reconocimiento de los gestos de la mano sigue siendo un obstáculo importante, especialmente en el contexto de los traductores del lenguaje de señas.

El objetivo general de esta tesis es desarrollar y validar una metodología para la caracterización analítica de la estructura morfológica de la mano utilizando objetos

matemáticos, como matrices, triángulos y curvas Bézier. Los objetivos específicos incluyen la extracción de características distintivas de estos objetos matemáticos que representan una mano, el entrenamiento de una red neuronal para la clasificación de gestos de la mano y la evaluación de la precisión y eficacia de esta metodología en diferentes contextos.

Nuestra hipótesis de investigación es que un enfoque basado en objetos matemáticos para el modelado de la mano permitirá una identificación de gestos más precisa y eficiente, mejorando la interacción persona-computadora, especialmente en el contexto del lenguaje de señas.

La metodología de este trabajo combina la utilización de herramientas de procesamiento de imágenes y técnicas de extracción de características. Las imágenes de los gestos de la mano se procesarán utilizando el software MediaPipe para la identificación de la mano, y las características extraídas se utilizarán para entrenar una red neuronal. Las técnicas de aprendizaje automático se aplicarán para el modelado de la mano y la identificación de gestos.

La tesis está organizada de la siguiente manera: en primer lugar, se presenta una revisión del estado del arte sobre el reconocimiento de gestos de la mano. A continuación, se detalla la metodología de investigación propuesta, desde la recolección de datos hasta el análisis y la interpretación. Posteriormente, se exponen y discuten los hallazgos de la investigación, subrayando su relevancia en el campo del reconocimiento de gestos de la mano. Finalmente, se presentan las conclusiones del estudio, seguidas de recomendaciones para investigaciones futuras en este campo.

Así, este estudio ofrece un enfoque innovador para el modelado de la mano y el reconocimiento de gestos, lo que se espera que tenga implicaciones significativas en la mejora

de la interacción persona-computadora, en particular para la comunidad de personas sordas y con discapacidades auditivas.

Capítulo 2

Antecedentes

La comunicación no verbal, como los gestos, juega un papel crucial en nuestras interacciones, representando alrededor del 65 % de nuestros mensajes. Los gestos pueden ser de mano, cabeza, rostro y cuerpo. La interacción humano-computadora (HCI) efectiva requiere sistemas de reconocimiento de gestos robustos y precisos, que se usan como alternativa a los dispositivos de HCI comunes, como mouse y teclados. Los sistemas de reconocimiento automático, como el reconocimiento de gestos con la mano, son áreas de investigación muy activas y significativas para la HCI [1].

Las técnicas de reconocimiento de gestos se basan en capturar la imagen del gesto de la mano humana mediante ordenador, lo que puede realizarse mediante reconocimiento basado en visión, utilizando cámaras web o de profundidad, y no requiere dispositivos especiales. También se pueden utilizar herramientas especiales como guantes alámbricos o inalámbricos que detectan los movimientos de la mano del usuario, y dispositivos de entrada de detección de movimiento (Microsoft Kinect, Leap Motion, etc.) que capturan los gestos y

movimientos de la mano [2].

Adicionalmente existen otras técnicas de reconocimiento de gestos basadas en la visión para extraer imágenes [3-5]. Otros estudios han implementado el uso de guantes con sensores de flexión para reconocer gestos [6-8]. Se ha demostrado que Leap Motion Controller tiene un alto potencial para aplicaciones de tipo dedo suave y ha sido utilizado para la captura de gestos [9-11].

El seguimiento de la mano *(tracking)* es una técnica que permite a la computadora rastrear la mano del usuario y separarla del fondo y de los objetos circundantes [2]. Los métodos de extracción de características se utilizan para extraer información útil de las imágenes que ayuda en el proceso de reconocimiento de gestos.

Diversos trabajos aplican métodos variados para esta tarea. Haitham, Alaa y Sabah [12] aplicaron la extracción de características de un contorno de la mano con fondo complejo. Weiguo et al. [3] usaron la Transformada de Fourier Discreta (DFT), Salunke y Bharkad [13] procesaron imágenes y extrajeron un histograma de gradientes de las mismas. Himadri et al. [14] se enfocó en la segmentación de las manos usando aproximación de polígonos y descomposición convexa aproximada.

Entre otras técnicas de extracción de características, Qingrui et al. [15] usaron el algoritmo de extracción de objetos salientes basado en Contraste Regional (RC), Hari et al. [16] aplicaron un algoritmo de detección de gestos, y Chenyang, Yingli y Matt [17] implementaron cinco estrategias de extracción de características diferentes. Jose et al. [18] usaron técnicas de procesamiento de imágenes digitales para eliminar el ruido y mejorar el contraste.

Las características extraídas cambian de una aplicación a otra. Algunas de las

características que podrían considerarse son: estado de los dedos, estado del pulgar, color de la piel, alineaciones de los dedos, y la posición de la palma [2]. Se han considerado y extraído varias características de los gestos de la mano que son altamente dependientes de la aplicación.

Cabe destacar que la extracción de características permite la clasificación y el reconocimiento de los gestos de la mano en una amplia gama de aplicaciones. Esto se logra utilizando una variedad de métodos que se adaptan a las necesidades específicas de cada aplicación, y que son capaces de identificar características únicas de la mano y los gestos que realiza. La elección del método y las características a extraer dependen del contexto y del objetivo de la aplicación.

Por otro lado, se tiene que el desarrollo de sistemas de reconocimiento de gestos de la mano, como las aplicaciones de lenguaje de señas, es extremadamente importante para superar las barreras de comunicación con personas no familiarizadas con este lenguaje. La tecnología que traduce automáticamente los movimientos de la mano en texto o habla audible puede ayudar a reducir esta barrera. Los sistemas de reconocimiento de gestos basados en visión tienen una amplia gama de aplicaciones, incluyendo comunicación, educación y rehabilitación, y pueden ser especialmente útiles en situaciones donde no se cuenta con un intérprete humano para el lenguaje de señas.

El reconocimiento de gestos de la mano es una aplicación que transforma los gestos de la mano del lenguaje de señas en salidas como texto o voz. Esta tecnología puede dividirse en sistemas basados en visión (donde los gestos son capturados por una o más cámaras) y sistemas basados en dispositivos (donde se utiliza un dispositivo de medida directa, como guantes electrónicos equipados con sensores)[19].

A pesar de que los sistemas basados en dispositivos son eficientes, su uso práctico es limitado debido a la necesidad de llevar un dispositivo engorroso para interactuar con el sistema. Este problema no se presenta en los sistemas basados en visión, que permiten una interacción más natural y tienen una amplia aplicación en escenarios al aire libre.

El reto para los sistemas basados en visión radica en cómo manejan conjuntos de datos compuestos por gestos manuales dinámicos en lenguaje de señas, como signos aislados y continuos. Muchos trabajos existentes se centran en reconocer gestos aislados, limitando su uso en aplicaciones del mundo real. Además, el desarrollo de reconocimiento de gestos manuales con sistemas basados en visión requiere el uso de métodos de extracción de características y discriminación más potentes[1].

Existen dos principales enfoques para los sistemas de reconocimiento de gestos de la mano: el basado en hardware y software especializado que interpreta la forma y el movimiento de las manos [20, 21], y el basado en el procesamiento de imágenes [22, 23]. Ambos se centran en extraer características de los gestos, ya sean rasgos estáticos o dinámicos, para luego realizar una clasificación [24, 25].

Los sistemas de traducción del lenguaje de señas también se dividen en dos grupos. El primero incluye sistemas que emplean hardware externo montado en un guante para detectar los rasgos de interés [26, 27]. Este hardware puede variar desde sensores de velocidad, unidades de movimiento inercial, hasta el uso de electromiografía en algunos casos [28]. Estos sistemas buscan determinar la posición y los movimientos que los usuarios de lenguaje de señas hacen para cada palabra, permitiendo así capturar los rasgos característicos de las mismas.

El segundo grupo de sistemas se centra en el análisis automatizado de imágenes y

tiene dos vertientes. La primer vertiente se basa en la detección de gestos estáticos donde no hay movimiento de las manos y la forma de estas es el rasgo principal para distinguir un gesto de otro [29, 30]. La segunda vertiente se enfoca en gestos dinámicos, donde el tipo de movimiento es una característica clave para la clasificación correcta de cada palabra.

Un factor común en ambos grupos de sistemas de traducción es la técnica de aprendizaje automático que utilizan. El objetivo principal es la correcta extracción de características de la mano, ya que una vez que se tiene una forma eficiente de procesar dichas características, se pueden generar grandes volúmenes de datos para su clasificación supervisada. Un área de desarrollo prometedora para estos sistemas es el uso de aprendizaje no supervisado a través de la extracción de características [31, 32] y la identificación de patrones morfológico-espaciales de la mano.

Finalmente, los sistemas de control de dispositivos a distancia se han vuelto cada vez más relevantes en los últimos años, aportando análisis y herramientas valiosas para la identificación de gestos. Estos sistemas se han implementado en una variedad de contextos, desde el control de dispositivos domésticos tradicionales [33], hasta su integración en sistemas avanzados de automóviles [34], e incluso en entornos médicos para minimizar el contacto físico con los instrumentos en un quirófano [35].

Capítulo 3

Trabajo Relacionado

En este capítulo se ofrece una breve descripción de los métodos y resultados más recientes relacionados con las técnicas de reconocimiento de los gestos de la mano. Aquí se presentan diferentes métodos y enfoques para el reconocimiento de los gestos de la mano y se discuten los principales problemas en esta área.

La interacción con máquinas basada en los gestos de las manos es una técnica natural de comunicación que las personas han adoptado en su vida cotidiana. En la actualidad, muchos investigadores del mundo académico y de la industria están estudiando diversas aplicaciones para hacer que las interacciones basadas en RGM sean más fáciles, naturales y cómodas sin necesidad de llevar ningún dispositivo adicional. Las aplicaciones de RGM van desde el control de juegos hasta el control de robots mediante visión por computadora, desde la realidad virtual hasta los sistemas domésticos inteligentes, desde los sistemas de seguridad hasta los sistemas de formación, y en muchos otros ámbitos[33-35].

El reconocimiento de gestos es una disciplina especializada que se encuentra en la intersección de la visión por computadora y la interpretación de imágenes. Su objetivo es traducir una imagen o una secuencia de imágenes, es decir, un video, en una descripción que posea un significado interpretativo.

Este capítulo explora a fondo el estado actual de las diversas técnicas empleadas en el reconocimiento de gestos de la mano. En particular, se enfoca en los enfoques avanzados que integran métodos sustentados en el aprendizaje automático, como las redes neuronales artificiales y la lógica difusa, entre otros.

Además, se presentan los métodos de preprocesamiento de imágenes que se utilizan para la segmentación de regiones de interés. Un ejemplo de esto es el delineado del contorno de la mano, que es esencial para la creación de la imagen de la mano.

3.1. Enfoque basado en apariencia

El enfoque basado en la apariencia explora la utilización de técnicas de visión por computadora para capturar y analizar imágenes o videos de las manos. Los trabajos de investigación han utilizado técnicas como la segmentación de la piel, la detección de contornos y la extracción de características visuales para interpretar los gestos. Estos métodos son menos dependientes del contexto y se centran principalmente en los atributos visibles, como la forma, la orientación y la posición de la mano.

Luca Ballan et al. [36] proponen utilizar puntos salientes aprendidos de forma discriminatoria en los dedos y estimar las asociaciones dedo-punto saliente simultáneamente con la estimación de la pose de la mano. Introducen una función objetivo diferenciable que también tiene en cuenta los bordes, el flujo óptico y las colisiones. Las evaluaciones cualitativas y cuantitativas muestran que el enfoque propuesto consigue resultados muy precisos en varias secuencias difíciles que contienen manos y objetos en acción.

C Nolker et al. [37] describen en su articulo un reconocimiento de gestos basado en las puntas de los dedos (GREFIT, por sus siglas en ingles, Gesture REcognition based on

FInger Tips), un sistema que reconoce posturas continuas de la mano a partir de imágenes de vídeo de nivel de gris (captura de posturas). El enfoque permite una identificación completa de todos los ángulos de las articulaciones de los dedos (haciendo, sin embargo, algunas suposiciones sobre los acoplamientos de las articulaciones para simplificar los cálculos). Esto permite una reconstrucción completa de la forma tridimensional (3-D) de la mano, utilizando un modelo de mano articulada con 16 segmentos y 20 ángulos de articulación. Este modelo se diseña de acuerdo con las dimensiones y posibilidades de movimiento de una mano humana natural. La mano virtual imita la mano del usuario con una precisión notable y puede seguir posturas a partir de imágenes en escala de grises a una frecuencia de imagen de 10 Hz.

Ambos trabajos anteriores se basan en la detección de la punta de los dedos para la construcción de imágenes de la mano.

Chengde Wan et al. [38] presentan un marco de regresión jerárquica para estimar las posiciones de las articulaciones de la mano a partir de imágenes de profundidad individuales basadas en las normales locales de la superficie. La regresión jerárquica sigue la topología arbórea de la mano desde la muñeca hasta la punta de los dedos. Proponen un bosque de regresión condicional que utiliza una nueva característica de diferencia normal. En cada etapa de la regresión, el marco de referencia se establece a partir de la normal local de la superficie o de las articulaciones de la mano estimadas previamente.

I. Oikonomidis et al. [39] en un primer trabajo presentan un enfoque en el cual tratan el problema como un problema de optimización, por un lado proponen un método que se basa en observaciones visuales sin marcadores para seguir la articulación completa de dos manos que interactúan entre sí de forma compleja y sin restricciones. En un segundo trabajo [40] presentan una solución novedosa de recuperar y rastrear la posición 3D, la orientación y la

articulación completa de una mano humana a partir de observaciones visuales sin marcadores obtenidas por un sensor Kinect, buscando los parámetros del modelo de mano que minimicen la discrepancia entre la apariencia y la estructura 3D de las instancias hipotéticas de un modelo de mano y las observaciones reales de la mano.

3.2. Enfoque basado en la extracción de características

En los últimos años la extracción de características ha tenido un gran impacto en los trabajos de reconocimiento de gestos de la mano sobresaliendo el análisis de componentes principales (PCA), desde esta perspectiva el objetivo del reconocimiento de gestos de la mano es clasificar los datos de los gestos de la mano representados por algunas características en un número predeterminado de clases de gestos.

El trabajo de Ansar et al. [41] presenta un novedoso método de reconocimiento de los gestos de la mano para los electrodomésticos inteligentes utilizando sensores de imagen. El modelo propuesto se divide en seis pasos. En primer lugar, se realiza un preprocesamiento para eliminar el ruido de los fotogramas de vídeo y redimensionar cada fotograma a una dimensión específica. En segundo lugar, se detecta la mano mediante un modelo de red neuronal de convolución basado en un detector de disparo único (SSD-CNN, por sus siglas en ingles, Single Shot Detector-based Convolution Neural Network model). En tercer lugar, se localizan puntos de referencia en la mano mediante el método del esqueleto. En cuarto lugar, se extraen características basadas en trayectorias de puntos, diferenciación de cuadros, histogramas de orientación y nubes de puntos 3D. En quinto lugar, las características se optimizan mediante lógica difusa y, por último, se utiliza el clasificador H-Hash para la clasificación de los gestos de la mano. El sistema se ha probado en dos conjuntos de datos de referencia, a saber, el conjunto de datos de manos IPN y el conjunto de datos Jester. La

precisión del reconocimiento en el conjunto de datos de manos IPN es del 88,46 % y en el conjunto de datos Jester es del 87,69 %. Los usuarios pueden controlar sus electrodomésticos inteligentes, como la televisión, la radio, el aire acondicionado y la aspiradora, utilizando el sistema propuesto.

Jalal et al. [42] proponen un novedoso sistema de estimación de la pose humana (HPE, por sus siglas en ingles, human pose estimation) y de clasificación de eventos sostenibles (SEC, por sus siglas en ingles, sustainable event classification) para el que han diseñado un modelo de palo pseudo-2D. Para extraer las características de la silueta humana de cuerpo entero, se proponen varias características como la energía, el seno, los movimientos de las distintas partes del cuerpo y una vista cartesiana 3D de gradientes de suavizado características. Las características extraídas para representar los puntos clave de la postura humana incluyen la apariencia 2D rica punto angular, y autocorrelación multipunto. Tras la extracción de los puntos clave, aplican una clasificación jerárquica y un modelo de optimización mediante la optimización de rayos y un algoritmo K-ary tree hashing.

A. Kapitanov et al. [43] presentan un enorme conjunto de datos HaGRID (por sus siglas en ingles, HAnd Gesture Recognition Image Dataset) para sistemas de reconocimiento de gestos con las manos (HGR). Este conjunto de datos contiene 552.992 muestras divididas en 18 clases de gestos. Las anotaciones consisten en cuadros delimitadores de las manos con etiquetas de gestos y marcas de las manos protagonistas. El conjunto de datos propuesto permite construir sistemas HGR, que pueden utilizarse en servicios de videoconferencia, sistemas de domótica, el sector de la automoción, servicios para personas con deficiencias auditivas y del habla, etc. Nos centramos especialmente en la interacción con dispositivos para manejarlos. Además, proponen las líneas de base para las tareas de detección de manos y clasificación de gestos.

D. Kumar et al. [44] proponen un sistema basado en visión para el reconocimiento del gesto estático de la mano. Permite reconocer el gesto en variación de iluminación, rotación, posición y tamaño de las imágenes gestuales. El sistema propuesto consta de tres fases: preprocesamiento, extracción de características y clasificación. La fase de preprocesamiento implica un proceso de mejora, segmentación, rotación y filtrado de la imagen. Para obtener una imagen gestual invariante de la rotación, en este trabajo proponen una técnica novedosa que consiste en hacer coincidir la 1ª componente principal de los gestos de la mano segmentados con ejes verticales. En la fase de extracción de características, este trabajo extrae secuencias de contorno localizadas y características basadas en bloques y propone una novedosa mezcla de características (o características combinadas) para una mejor representación del gesto estático de la mano. Las características combinadas se aplican como entrada al clasificador de máquina de soporte vectorial multiclase para reconocer el gesto estático de la mano.

Jing Li et al. [45] proponen un sistema de reconocimiento de gestos estáticos que combina información de profundidad y datos de esqueleto para clasificar los gestos. A través de la fusión de características, los gestos de los dígitos de la mano del 0 al 9 pueden ser reconocidos con precisión y eficiencia. Según los resultados experimentales, el sistema de reconocimiento de gestos propuesto es eficaz y robusto, y es invariable ante fondos complejos, cambios de iluminación, inversión, distorsión estructural, rotación, etc.

Haitham et al. [46] exploran la utilidad de dos métodos de extracción de características, el contorno de la mano y el algoritmo de momentos complejos, para resolver el problema del reconocimiento de gestos de la mano, identificando las principales ventajas y desventajas de cada método. Construyen una red neuronal artificial con el fin de clasificar mediante el algoritmo de aprendizaje de retropropagación. El sistema propuesto presenta un algoritmo

de reconocimiento para reconocer un conjunto de seis gestos estáticos específicos de la mano: Abrir, Cerrar, Cortar, Pegar, Maximizar y Minimizar. La imagen del gesto de la mano pasa por tres etapas, preprocesamiento, extracción de características y clasificación. En la etapa de preprocesamiento se aplican algunas operaciones para extraer el gesto de la mano de su fondo y preparar la imagen del gesto de la mano para la etapa de extracción de características y posteriormente la clasificación.

Chahid et al. [47] proponen un método de extracción de características basado en la matriz de pesos de posición para la clasificación multiclase, con el fin de mejorar la interpretación de las señales biomédicas. Este método se valida en el reconocimiento de señales de electromiograma de superficie (sEMG, por sus siglas en ingles) para ocho gestos diferentes de la mano. El conjunto de datos CapgMyo [48] utilizado consiste en señales sEMG de alta densidad a través de 128 canales adquiridos de 9 sujetos intactos.

3.3. Enfoque basado en modelos

Este método se basa en la utilización de un sistema de marcadores, cuyo objetivo es adaptar un modelo tridimensional de una mano a estos marcadores detectados para obtener una representación de la postura. Es necesario subrayar que estos modelos sirven como entrada para un algoritmo de aprendizaje automático, diferenciándose de la creación de un modelo de mano en sí mismo. A pesar de que a menudo es necesario limpiar los datos, debido a errores en las lecturas cuando la mano se oculta o se mueve rápidamente, este enfoque tiene la ventaja de requerir escaso entrenamiento, lo que lo hace eficaz para identificar posturas desconocidas. No obstante, debido a que gran parte del procesamiento ocurre en tiempo de ejecución, puede resultar demasiado demandante para aplicaciones en tiempo real [49].

James M. Rehg y Takeo Kanade [50] describen un sistema de seguimiento de manos basado en modelos, denominado DigitEyes, que puede recuperar el estado de un modelo de mano de 27 grados de libertad a partir de imágenes ordinarias en escala de grises a velocidades de hasta 10 Hz. La detección pasiva del movimiento de la mano y las extremidades humanas es importante para una amplia gama de aplicaciones, desde la interacción persona-ordenador hasta la medición del rendimiento deportivo. Los mecanismos articulados de alto grado de libertad, como la mano humana, son difíciles de rastrear debido a su amplio espacio de estados y a la compleja apariencia de su imagen.

Tagliasacchi et al. [28] proponen un sistema de rastreo de la mano en tiempo real utilizando una cámara RGBD (Color y profundidad) como entrada, usando una optimización basada en el algoritmo ICP (*Iterative Closest Point*) en conjunto con una base de datos de poses de manos posibles. EL modelo que utilizan para representar la mano consiste en una serie de cilindros optimizados con 26 grados de libertad.

Fan Shang et al [51] proponen un algoritmo con el cual detectan 21 puntos claves en la mano para su representación y modelado. Su algoritmo se divide en dos partes: la detección de la mano utilizando una imagen con la palma orientada a la cámara, y el modelado y mapeo de 21 puntos colocados en la imagen recortada de la mano obtenida en el punto anterior.

En la siguiente sección mencionaremos algunos trabajos relacionados respecto a como las redes neuronales pueden identificar una mano o un gesto de una mano, mediante algunos de los modelos de los trabajos previamente mencionados que sirven como entrada a estos sistemas que utilizan redes neuronales.

3.4. Redes neuronales artificiales

Una red neuronal artificial (RNA) está compuesta por una serie de elementos de procesamiento altamente interconectados (también llamados neuronas) que operan conjuntamente para resolver problemas específicos [52]. Las RNA pueden configurarse para resolver problemas como el reconocimiento de patrones o la extracción de datos mediante modelos basados en el aprendizaje. Las RNA también tienen capacidades como el aprendizaje adaptativo, la autoorganización y las operaciones en tiempo real mediante un hardware especial.

Nolker [37] utiliza un enfoque por capas basado en RNA para detectar las puntas de los dedos. Los vectores de las puntas de los dedos se obtienen y se transforman en ángulos de las articulaciones de los dedos en un modelo de mano articulada. Para cada dedo, se entrena una red distinta con los mismos vectores de características. La entrada de cada red es un vector de tamaño 35, mientras que la salida es sólo bidimensional. Lee [53] utiliza el modelo de Markov oculto (HMM) para el reconocimiento de gestos utilizando características de forma. El estado del gesto se determina tras estabilizar el componente de la imagen como dedos abiertos en fotogramas consecutivos.

Wang [54] propuso un potente enfoque basado en el flujo óptico para el reconocimiento de la acción humana utilizando modelos de aprendizaje. En este enfoque de flujo óptico, las partes ocultas de la imagen también se etiquetan. Este algoritmo basado en el margen máximo puede aplicarse al reconocimiento de gestos.

Zhan [55] propone un algoritmo para el reconocimiento de gestos de la mano en tiempo real utilizando redes neuronales convolucionales (CNN). La CNN propuesta alcanza una precisión media del 98,76 % en el conjunto de datos que comprende 9 gestos de la mano

y 500 imágenes para cada gesto.

Jimin y Shangbo [56] proponen un método de reconocimiento basado en una estrategia que combina las redes neuronales convolucionales 2D con la fusión de características. Los fotogramas clave originales y los fotogramas clave de flujo óptico se utilizan para representar las características espaciales y temporales respectivamente, que luego se envían a la red neuronal convolucional 2D para la fusión de características y el reconocimiento final. Para garantizar la calidad del gráfico de flujo óptico extraído sin aumentar la complejidad de la red, utilizan el método de orden fraccionario para extraer el gráfico de flujo óptico, combinando el cálculo fraccionario y el aprendizaje profundo.

Jhaung et al. [57] propone un sistema dinámico de gestos de la mano basado en un radar FMCW de 60 GHz que puede utilizarse para el control de dispositivos sin contacto; reciben las señales de radar de los gestos de la mano y las transforman en dominios comprensibles para el ser humano, como el alcance, la velocidad y el ángulo. Con estos datos, personalizan el sistema a diferentes escenarios. Proponen un modelo de aprendizaje profundo de extremo a extremo (red neuronal y memoria a corto plazo), que extrae las señales de radar transformadas en características y clasifica las características extraídas en etiquetas de gestos de la mano. Dentro de su esfuerzo de recopilación de datos de entrenamiento, utilizan una cámara solo para apoyar el etiquetado de datos de gestos de la mano. La precisión del modelo puede alcanzar el 98 %.

3.5. Propuestas de traductores del lenguaje de señas

Una gran motivación detrás del reconocimiento de los gestos de la mano es la traducción del lenguaje de señas, existen múltiples trabajos que proponen diferentes técnicas

de aprendizaje, detección, clasificación, etc., en torno al lenguaje de señas. Estos trabajos van en torno al apoyo de la comunidad para mejorar la comunicación entre las personas.

Daniel Hernandez en su tesis [58] aborda la detección de algunas palabras en la lengua de señas mexicana sin el uso de dispositivos adicionales como guantes. Los experimentos llevados acabo en ese trabajo muestran que es posible la detección de algunos gestos o palabras del lenguaje de señas, y que se necesitan características específicas de manos, brazos, cara y cuerpo.

Zheng et al. [59] proponen dos adaptaciones a los modelos neurales tradicionales de LS (Lenguaje de Señas) utilizando módulos optimizados relacionados con la tokenización. En primer lugar, introducen un algoritmo de compresión de la densidad del flujo de fotogramas (FSDC, por sus siglas en inglés, Frame Stream Density Compression) para detectar y reducir los fotogramas similares redundantes, lo que acorta eficazmente las frases de signos largas sin perder información. Después sustituyen el codificador tradicional en un módulo de traducción automática neural (NMT, por sus siglas en inglés, Neural Machine Translation) por una arquitectura mejorada, que incorpora una unidad de convolución temporal (T-Conv) y una unidad GRU (por sus siglas en inglés, Gated Recurrent Unit) bidireccional jerárquica dinámica (DH-BiGRU) de forma secuencial. El componente mejorado tiene en cuenta la información de tokenización temporal para extraer información más profunda con un consumo de recursos razonable. Los experimentos con el conjunto de datos RWTH-PHOENIX-Weather 2014T muestran que el modelo propuesto supera a la línea de base del estado del arte hasta una ganancia de puntuación de 1,5+ BLEU-4.

Junfu et al. [60], proponen una red de alineación con optimización iterativa para el reconocimiento de la lenguaje de señas continua débilmente supervisada. El marco consta de dos módulos: una red residual convolucional 3D (3D-ResNet) para el aprendizaje de

características y una red codificadora-decodificadora con clasificación temporal conexionista (CTC) para el modelado de secuencias. Los dos módulos anteriores se optimizan de forma alternativa. En la red de aprendizaje de secuencias con codificador-decodificador se incluyen dos decodificadores, el decodificador LSTM (por sus siglas en inglés, Long Short-Term Memory) y el decodificador CTC. Ambos decodificadores se entrenan conjuntamente mediante el criterio de máxima verosimilitud con una restricción de alineación dinámica temporal suave (soft-DTW). La ruta de deformación, que indica la posible alineación entre los clips de vídeo de entrada y las palabras de signo, se utiliza para afinar la 3D-ResNet como etiquetas de entrenamiento con pérdida de clasificación. Tras el ajuste fino, se extraen las características mejoradas para la optimización de la red de aprendizaje de secuencias de codificador-decodificador en la siguiente iteración. El algoritmo propuesto se evalúa en dos puntos de referencia de reconocimiento del lenguaje de signos continuo a gran escala.

Prasad et al. [61] proponen una metodología innovadora que fusiona conceptos de visión por computadora y aprendizaje automático para desarrollar una aplicación capaz de predecir con precisión el equivalente textual de las señales en lengua de signos. En su estudio, exploraron el uso de máquinas de soporte vectorial (SVM, por sus siglas en inglés Support Vector Machine), un algoritmo de aprendizaje automático supervisado para clasificar patrones de señas. Utilizaron el sensor de la cámara Kinect de Microsoft Xbox 360 para capturar el lenguaje de señas con alta resolución y mayor profundidad de píxeles. Mediante un conjunto de ejemplos de entrenamiento, cada uno etiquetado con el valor textual correspondiente a la señal, evaluaron cómo el algoritmo SVM construye un modelo capaz de clasificar nuevos ejemplos no incluidos en el entrenamiento. Además, analizaron la eficacia del algoritmo de reconocimiento de patrones en la transformación escalar de características invariantes (SIFT, por sus siglas en inglés Scalar Invariant Feature Transform) y evaluaron la precisión de la clasificación realizada por la SVM.

Wang et al. [62] nos dicen que para utilizar la dinámica a largo plazo sobre una secuencia de signos aislada, proponen una representación basada en la matriz de covarianza para fusionar de forma natural la información procedente de fuentes multimodales. Para abordar el inconveniente inducido por la métrica riemanniana comúnmente utilizada, la proximidad de las matrices de covarianza se mide en la variedad de Grassmann. Sin embargo, la métrica de Grassmann inherente no puede aplicarse directamente a la matriz de covarianza. Resuelven este problema evaluando y seleccionando los vectores singulares más significativos de las matrices de covarianza de las secuencias de signos. La representación compacta resultante se denomina matriz de covarianza de Grassmann. Finalmente, la métrica de Grassmann se utiliza como kernel para la máquina de vectores de soporte, lo que permite el aprendizaje de los signos de forma discriminativa. Para validar el método propuesto, recopilaron tres conjuntos de datos sobre el lenguaje de signos, en los que las evaluaciones muestran que el método propuesto supera a los métodos del estado del arte tanto en precisión como en coste computacional.

Muneer Al-Hammadi et al. [63] proponen un sistema novedoso para el reconocimiento dinámico de gestos de la mano utilizando múltiples arquitecturas de aprendizaje profundo para la segmentación de la mano, las representaciones de características locales y globales, y la globalización y el reconocimiento de características de la secuencia. El sistema propuesto se evalúa en un conjunto de datos, que consiste en 40 gestos dinámicos de la mano realizados por 40 sujetos en un entorno no controlado.

Capítulo 4

Composición, modelado y detección de la mano

En los últimos años, la investigación en visión por computadora ha experimentado una expansión rápida y exitosa. Este éxito se ha logrado gracias a dos factores principales. Por un lado, se ha producido una adopción y adaptación de métodos de aprendizaje automático, lo cual ha contribuido significativamente a los avances en el campo [64]. Por otro lado, se ha dedicado un esfuerzo considerable al desarrollo de nuevas representaciones y modelos específicos para abordar los desafíos de la visión por computadora, lo que ha llevado al diseño de soluciones eficaces.

Entre los avances destacados, se ha logrado un progreso significativo en la detección de objetos, donde se han desarrollado algoritmos capaces de identificar y localizar objetos con una precisión cada vez mayor [65]. Además, la segmentación semántica ha avanzado notablemente, permitiendo la identificación y clasificación precisa de diferentes regiones en una imagen. Otra área de avance ha sido el reconocimiento facial, donde los sistemas de visión por computadora han alcanzado niveles de precisión cercanos al rendimiento humano. También se han realizado avances en la detección y seguimiento de personas en vídeos, lo que ha permitido aplicaciones como la vigilancia inteligente y la conducción autónoma [55].

Dado un conjunto de clases de objetos, la detección de objetos consiste en determinar la ubicación y la escala de todas las instancias de objetos, si las hay, que están presentes en una imagen. Así, el objetivo de un detector de objetos es encontrar todas las instancias de objetos de una o más clases de objetos dadas, independientemente de la escala, la ubicación, la pose, la vista con respecto a la cámara, las oclusiones parciales y las condiciones de iluminación.

En muchos sistemas de visión por computadora, la detección de objetos es la primera tarea que se realiza, ya que permite obtener más información sobre el objeto detectado y sobre la escena. Una vez que se ha detectado una instancia de objeto (por ejemplo, un rostro), es posible obtener más información, incluyendo: reconocer la instancia específica (por ejemplo, identificar el rostro del sujeto), rastrear el objeto en una secuencia de imágenes (por ejemplo, rastrear el rostro en un vídeo), y extraer más información sobre el objeto (por ejemplo, para determinar el sexo del sujeto), mientras que también es posible inferir la presencia o la ubicación de otros objetos en la escena (por ejemplo, una mano puede estar cerca de una cara y a una escala similar) y estimar mejor otra información sobre la escena (por ejemplo, el tipo de escena, interior o exterior, etc.), entre otra información contextual.

La detección de objetos se ha utilizado en muchas aplicaciones, siendo las más populares: la interacción persona-computadora (HCI), la robótica (por ejemplo, los robots de servicio), la electrónica de consumo (por ejemplo, los teléfonos inteligentes), la seguridad (por ejemplo, el reconocimiento, el seguimiento), la recuperación (por ejemplo, los motores de búsqueda, la gestión de fotos), y el transporte (por ejemplo, la conducción autónoma y asistida). Cada una de estas aplicaciones tiene diferentes requisitos, entre ellos: tiempo de procesamiento (fuera de línea, en línea o en tiempo real), robustez a las oclusiones, invariabilidad a las rotaciones (por ejemplo, rotaciones en el plano) y detección bajo cambios

de pose. Mientras que muchas aplicaciones consideran la detección de una sola clase de objeto (por ejemplo, caras) y desde una sola vista (por ejemplo, caras frontales), otras requieren la detección de múltiples clases de objetos (humanos, vehículos, etc.), o de una sola clase desde múltiples vistas (por ejemplo, vista lateral y frontal de vehículos). En general, la mayoría de los sistemas sólo pueden detectar una única clase de objeto a partir de un conjunto restringido de vistas y poses.

Dentro del área de estudio de la interacción persona-computadora, un tema que ha tenido un gran impacto en los últimos años es poder identificar partes del cuerpo humano dentro de una imagen, existen algunas bibliotecas de software que hacen esta identificación posible y con gran precisión como son: OpenPose [66], MediaPipe [20], Lens Studio [21], Banuba [67], entre otros. En este trabajo para la detección de la mano utilizaremos el software de MediaPipe.

4.1. Detección e identificación de la mano con MediaPipe

La capacidad de percibir la forma y el movimiento de las manos puede ser un componente vital para mejorar la experiencia del usuario en diversos ámbitos y plataformas tecnológicas. Por ejemplo, puede constituir la base para la comprensión del lenguaje de signos y el control de los gestos de la mano, y también puede permitir la superposición de contenidos e información digital sobre el mundo físico en la realidad aumentada. Si bien es algo natural para las personas, la percepción robusta de las manos en tiempo real es una tarea de visión por computadora decididamente desafiante, ya que las manos a menudo se ocluyen [1] a sí mismas o entre sí (por ejemplo, oclusiones de dedos/palmas y apretones de

[1]La oclusión es la situación en la cual una parte de la mano u objeto bloquea o cubre parcialmente otra parte de la mano, ocurre cuando una parte de la mano se encuentra detrás o bloqueada por otra parte de la misma mano, lo que dificulta su detección y seguimiento preciso.

manos) y carecen de patrones de alto contraste.

MediaPipe Hands es una solución de seguimiento de manos y dedos de alta fidelidad. Emplea el aprendizaje automático (ML) para inferir 21 puntos de referencia 3D de una mano a partir de un solo fotograma. Mientras que los enfoques actuales dependen principalmente de potentes entornos de escritorio para la inferencia, éste método logra un rendimiento en tiempo real en un teléfono móvil, e incluso se adapta a múltiples manos [20, 51, 68, 69].

MediaPipe Hands utiliza un flujo de procesamiento de ML que consiste en múltiples modelos que trabajan juntos: Un modelo de detección de la palma de la mano que opera sobre la imagen completa y devuelve un cuadro delimitador de la mano orientado. Un modelo de referencia de la mano que opera en la región de la imagen recortada definida por el detector de la palma y devuelve puntos clave de la mano en 3D de alta fidelidad.

Al proporcionar la imagen de la mano recortada con precisión al modelo de puntos de referencia de la mano, se reduce drásticamente la necesidad de aumentar los datos (por ejemplo, rotaciones, traslación y escala) y, en su lugar, la red neuronal artificial puede dedicar la mayor parte de su capacidad a la precisión de la predicción de coordenadas. Además, los recortes también pueden generarse basándose en los puntos de referencia de la mano identificados en el fotograma anterior, y sólo cuando el modelo de puntos de referencia ya no puede identificar la presencia de la mano se invoca la detección de la palma para volver a localizar la mano.

La funcionalidad de la línea de construcción se lleva a cabo a través de una gráfica de MediaPipe. Esta gráfica emplea una subgráfica especializada para el seguimiento de los puntos de referencia de la mano provenientes del módulo de puntos de referencia específico para este fin. Además, se encarga de la representación visual utilizando una subgráfica

exclusiva para el renderizado de la mano.

Dentro de la subgráfica de seguimiento de la mano, se hace uso interno de otra subgráfica que contiene información de referencia para la mano, proveniente del mismo módulo, y también de una subgráfica de detección de la palma suministrada por el módulo de detección de la palma.

4.1.1. Modo de detección de la palma de la mano

Para detectar la ubicación inicial de las manos, el modelo consiste en la detección de una sola toma optimizado para usos móviles en tiempo real. La detección de manos es una tarea decididamente compleja, la falta de características en las manos hace que sea comparativamente difícil detectarlas de forma fiable a partir de sus características visuales únicamente. En cambio, proporcionar un contexto adicional, como las características del brazo, del cuerpo o de la persona, ayuda a la localización precisa de la mano.

En primer lugar, entrenaron un detector de palmas en lugar de un detector de manos, ya que la estimación de los recuadros delimitadores de objetos rígidos como las palmas y los puños es mucho más sencilla que la detección de manos con dedos articulados. Además, como las palmas son objetos más pequeños, el algoritmo de supresión no máxima funciona bien incluso para los casos de auto-oclusión de dos manos, como los apretones de manos. Por otra parte, las palmas pueden modelarse utilizando cajas delimitadoras cuadradas (anclas en la terminología de ML), ignorando otras relaciones de aspecto, y reduciendo así el número de anclas en un factor de 3 a 5. En segundo lugar, se utiliza un extractor de características codificador-decodificador para obtener un mayor conocimiento del contexto de la escena incluso para los objetos pequeños (de forma similar al enfoque de RetinaNet [70]). Por último, minimizamos la pérdida focal durante el entrenamiento para soportar una gran

cantidad de anclajes resultantes de la alta varianza de escala.

4.1.2. Modelo de puntos de referencia de la mano

Tras la detección de la palma de la mano en toda la imagen, el modelo de puntos de referencia de la mano realiza una localización precisa de las coordenadas de 21 puntos de la mano en 3D dentro de las regiones de la mano detectadas mediante regresión, es decir, predicción directa de coordenadas. El modelo aprende una representación interna consistente de la postura de la mano y es robusto incluso con las manos parcialmente visibles y las auto-oclusiones.

Con el propósito de abarcar una gama más amplia de posiciones de la mano y ofrecer una supervisión más detallada de la estructura geométrica de la misma, se utiliza como base un modelo sintético de alta calidad de la mano. Este modelo se representa en diversos fondos mediante un proceso de renderizado que asigna las coordenadas 3D correspondientes. La figura 4.1 muestra los 21 puntos definidos por MediaPipe para la representación de la mano.

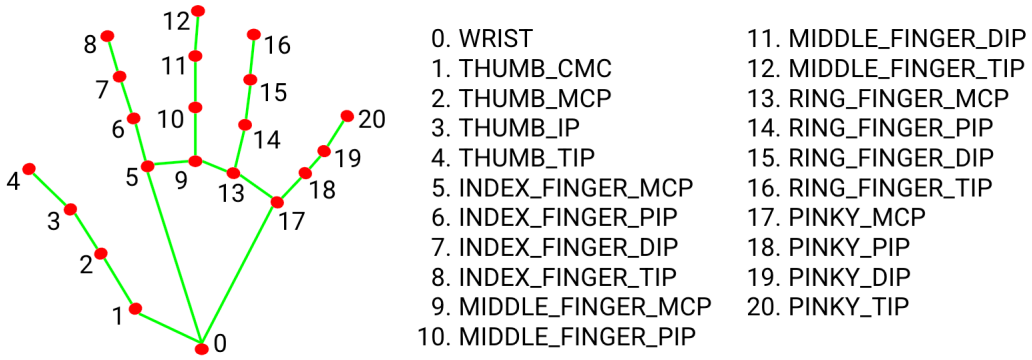


Fig. 4.1: Puntos de la mano obtenidos utilizando MediaPipe [20].

4.2. Modelado de la mano

Para poder profundizar en el sistema de gestos, es necesario distinguir primero entre el significado de una postura y de un gesto, la postura es una sola imagen que representa una sola orden como la señal de alto, mientras que el gesto es una secuencia de posturas que se refieren a un significado único cuando se combinan esas posturas juntas como mover la mano en una dirección específica para cambiar el volumen de una radio o televisión [32].

La mano debe ser modelada en el sistema para ser procesada correctamente. La aplicación exigente tiene un impacto significativo en la selección del modelo empleado [71]. El modelo de la mano puede ser temporal (movimiento) o espacial (forma) [71], se han construido reconocedores especiales para realizar un seguimiento del modelado temporal como el modelo oculto de markov (HMM) [72], la red neuronal (NN), el modelo basado en reglas y la máquina de estado finito, el modelado especial puede dividirse en modelo basado en la apariencia y modelo basado en 3D [32].

El modelo basado en la apariencia también se denomina modelo 2D o modelo basado en la vista. El modelo basado en la apariencia puede estar formado por plantillas, características de representación de la forma, vectores propios, las características de representación de la forma pueden ser geométricas o no geométricas, las características geométricas se consideran características vivas ya que pueden ser procesadas por separado como la ubicación de la punta del dedo y la ubicación de la palma, por el contrario, las características no geométricas se consideran características ciegas ya que no pueden ser vistas individualmente y se requiere un procesamiento colectivo.

El modelo 3D describe la forma de la mano y puede dividirse en modelos volumétricos y modelos esqueléticos, los modelos volumétricos son complejos de llevar a cabo especialmente

en aplicaciones en tiempo real en las que el factor de velocidad es muy crítico, por lo que, otros modelos geométricos como los cilindros y las esferas se consideran como alternativas a dicho modelo para la aproximación de la forma de la mano.

El otro tipo de modelo es el **modelo esquelético**, que captura la estructura de la mano mediante una representación en 3D con un conjunto reducido de parámetros en comparación con el modelo anterior. Al aplicar un modelo volumétrico a uno basado en la apariencia, el resultado se vuelve complejo y requiere un tiempo considerable para calcular los parámetros. Este enfoque se puede denominar, en el presente contexto, análisis por síntesis.

Esta clasificación se lleva a cabo a través de distintas categorías basadas en la interacción del movimiento de la mano, que puede manifestarse en dos tipos: manipulación directa y gestos simbólicos. En la manipulación directa, la ubicación actual de la mano se interpreta como el siguiente comando, mientras que en los gestos simbólicos, la acción se deriva del movimiento del objeto.

Sin embargo, aunque el modelado 3D resuelve el problema de la auto-oclusión, no resulta práctico para aplicaciones en tiempo real, dado que el tiempo es un factor limitante en este tipo de modelado.

La figura 4.2 muestra una taxonomía de los distintos modelos de la mano, clasificados según diversas categorías que se basan en la interacción del movimiento de la mano. No obstante, el modelado en 3D soluciona el problema de la auto-oclusión.

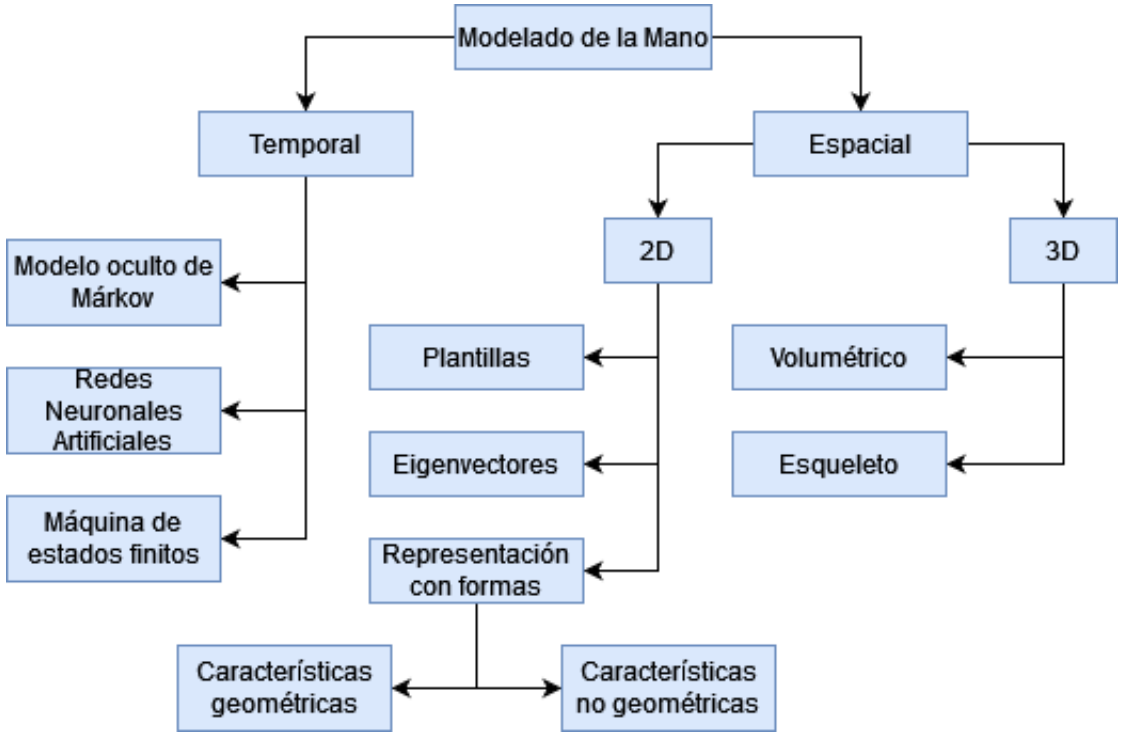


Fig. 4.2: Taxonomía de los distintos modelos de la mano [32].

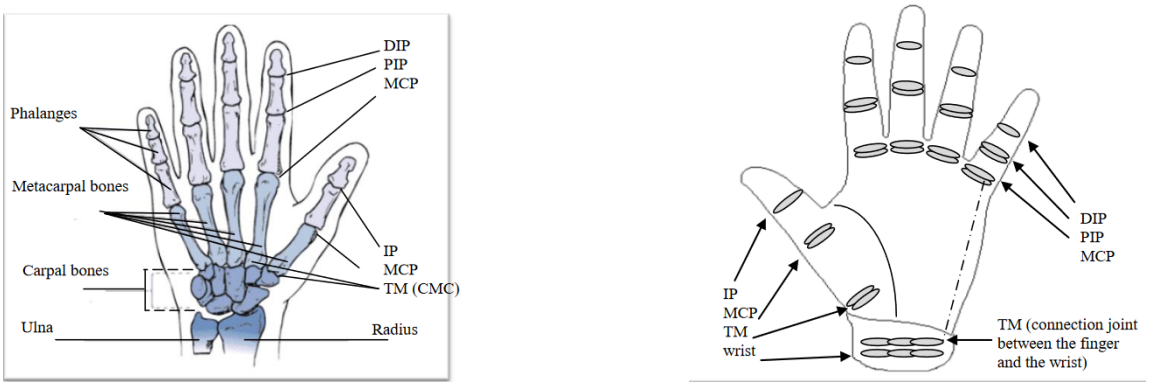
En este trabajo vamos a utilizar el modelo 3D de esqueleto el cual nos lo genera MediaPipe, para hacer posteriormente un análisis sobre el modelo de esqueleto y proponer un sistema de reconocimiento de gestos basado en la extracción de características.

4.2.1. Composición Morfológica

La mano humana es considerada como un objeto articulado con 27 huesos que forman la estructura de la mano y su capacidad para capturar, sostener, agarrar y soltar objetos sin problemas, cinco dedos están en la mano humana, cuatro dedos que son el meñique, el anular, el medio y el índice están alineados juntos y unidos a los huesos de la muñeca en un lazo, el pulgar tiene cierta distancia de ellos.

Los 27 huesos se distribuyen como 8 huesos en el área de la muñeca, y 19 huesos para los dedos desde el extremo de la uña hasta el área de la muñeca como se ve en la siguiente imagen. Sin embargo, las articulaciones de la mano tienen los siguientes nombres, inspirados en su ubicación: interfalángica distal (DIP) e interfalángica proximal (PIP), utilizadas para formar un solo dedo, excepto el pulgar, que no tiene ni distal ni proximal, ya que el número de sus huesos es menor que el de los demás dedos, por lo que sólo tiene interfalángica (IP), la articulación de conexión entre los dedos y los huesos metacarpianos se denomina Metacarpofalángica (MCP) que son cinco una por cada dedo, finalmente, la articulación de conexión entre los huesos metacarpianos y los huesos del carpo (muñeca) se denominan Trapeciometacarpianos (TM) o también Carpometacarpianos (CMC), estas diferentes formas de conexiones tienen un DoF [2] diferente como sigue: Las DIP tienen 4 DoF, las PIP tienen 4 DoF, una para las IP, el total es de 9 hasta ahora, las MCP tienen 2 DoF cada una, esto hará que la suma sea de 19 DoF, dos DoF más para la articulación TM para el dedo pulgar solamente y 6 DoF para la muñeca, por lo tanto, 27 DoF son proporcionadas por esta misteriosa criatura, fuera de toda duda, y sobresaliente mano humana, la figura 4.3 muestra las DoF de la mano.

[2]Degree of Freedom, DoF, por sus siglas en ingles, se refiere a la cantidad de movimientos independientes que pueden realizar las diferentes articulaciones de la mano.



a) Los huesos de la mano

b) Las articulaciones de la mano

Fig. 4.3: Composición morfológica de la mano [32].

Para entender el significado de DoF, considere el plano xyz, si el objeto puede cambiar su dirección sólo en una dirección permanentemente, se llama tiene un DoF como el ferrocarril que decide la dirección del tren en un curso todo el tiempo, para dos DoFs, el objeto puede moverse a través de dos coordenadas diferentes permanentemente, como un coche de conducción que puede ir en el plano xy en cualquier momento, por lo que, con respecto a los dedos, cada articulación que tiene un DoF puede realizar un movimiento de dirección en dos formas hacia adelante y hacia atrás que son la flexión y la expansión, para dos DoFs, el dedo puede realizar la abducción y la aducción, así como la flexión y la expansión, y así sucesivamente.

4.3. Visualización gráfica de MediaPipe

Partiendo de un pequeño set de datos, se utilizó Plotly [3], una biblioteca interactiva de código abierto, para poder observar con mayor detalle el comportamiento y la representación

[3]Plotly es una biblioteca de visualización interactiva que permite crear gráficos y visualizaciones de datos de alta calidad. Ofrece una amplia gama de tipos de gráficos y funciones interactivas, como zoom y selección de datos.

de la mano que nos ofrece MediaPipe, 21 coordenadas en xyz.

La visualización desempeña un papel esencial en el estudio de la solución de detección propuesta por MediaPipe, así como en el análisis y comprensión de cómo las coordenadas de la mano varían en función de su posición. La utilización de Plotly como herramienta de visualización interactiva resultó invaluable en este proceso. Al proporcionar una representación en tres dimensiones, Plotly permitió una exploración más profunda del modelo de la mano y brindó la flexibilidad de realizar modificaciones o agregar elementos al mismo, mejorando así nuestra comprensión del sistema.

Adicionalmente, para obtener una perspectiva comparativa, también realizamos una visualización adicional en dos dimensiones utilizando la biblioteca Matplotlib. Esto nos permitió evaluar cómo se ve afectada la información y la representación de la mano al reducir la dimensión de visualización. Al realizar esta comparación entre las visualizaciones en dos y tres dimensiones, pudimos identificar posibles pérdidas de información y comprender mejor la importancia de la visualización tridimensional en la representación de la mano.

En la figura 4.6 podemos observar la imagen de la mano, su representación en dos dimensiones con Matplotlib y en tres dimensiones con Plotly.

La combinación de herramientas de visualización como Plotly y Matplotlib fue fundamental para el análisis, comprensión y mejora del modelado de la mano proporcionado por MediaPipe. Estas herramientas nos permitieron explorar en detalle el comportamiento de las coordenadas de la mano, comprender su relación con la posición de la mano y evaluar la influencia de la representación en dos y tres dimensiones en la información visualizada.

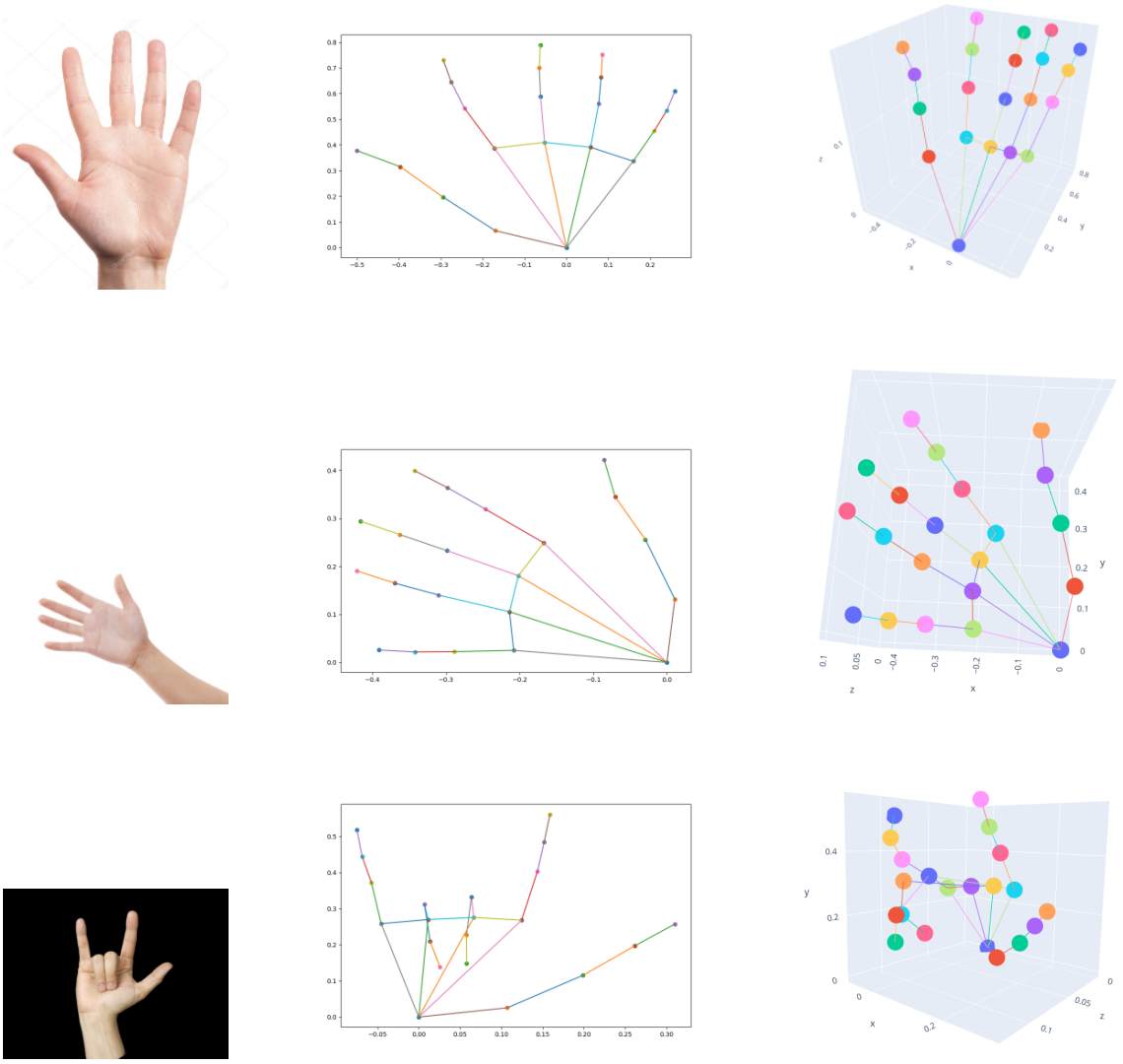


Fig. 4.6: Visualización de MediaPipe utilizando Plotly y Matplotlib

4.4. Conjunto de Datos HaGRID

Para llevar a cabo este trabajo, se utilizará el conjunto de datos HaGRID (Hand Gesture Recognition Image Dataset, por sus siglas en inglés) [43], el cual contiene 18 gestos de la mano distintos. Este conjunto de datos es extremadamente rico y diverso, ya que incluye aproximadamente 30 mil imágenes por gesto, cada una de las cuales presenta una serie de características diferentes, como diferentes fondos, perspectivas, resoluciones, formatos, calidad, tamaño, entre otras.

Es importante destacar que este conjunto de datos, aunque no es muy común en trabajos relacionados debido a su reciente publicación en el año 2022, proporciona una excelente base para la tarea que nos ocupa. En muchos casos, la entrada del programa o las imágenes a procesar no siempre serán óptimas para su detección y reconocimiento. Gracias al software de MediaPipe, que homologa la salida a partir de 21 puntos, podemos hacer un modelado independientemente de la entrada, lo que es de gran ayuda para lograr una detección y reconocimiento precisos.

El uso de este conjunto de datos nos permitirá realizar un análisis exhaustivo de los gestos de la mano y sus correspondientes características, lo que nos permitirá crear un modelo altamente preciso y eficiente. En las figuras 4.7 y 4.8 se pueden observar los gestos del dataset el cual incluye algunas de las señas más comunes que se utilizan en el día a día, y también se pueden observar algunos ejemplos del dataset como son los gestos desde distintas distancias, con diferente luz, posición, etc.

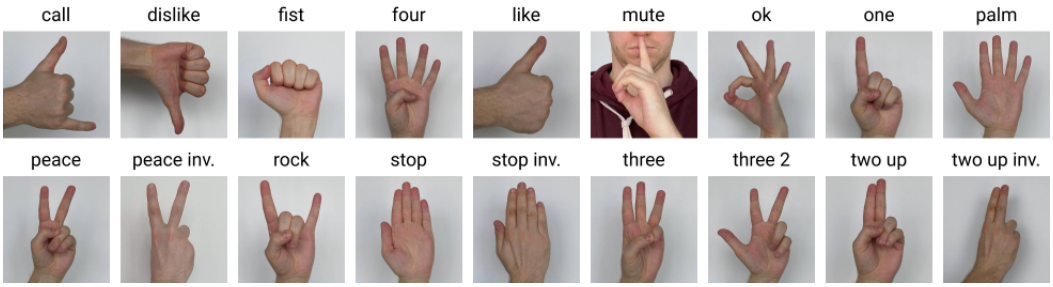


Fig. 4.7: Los 18 gestos distintos del dataset [43].

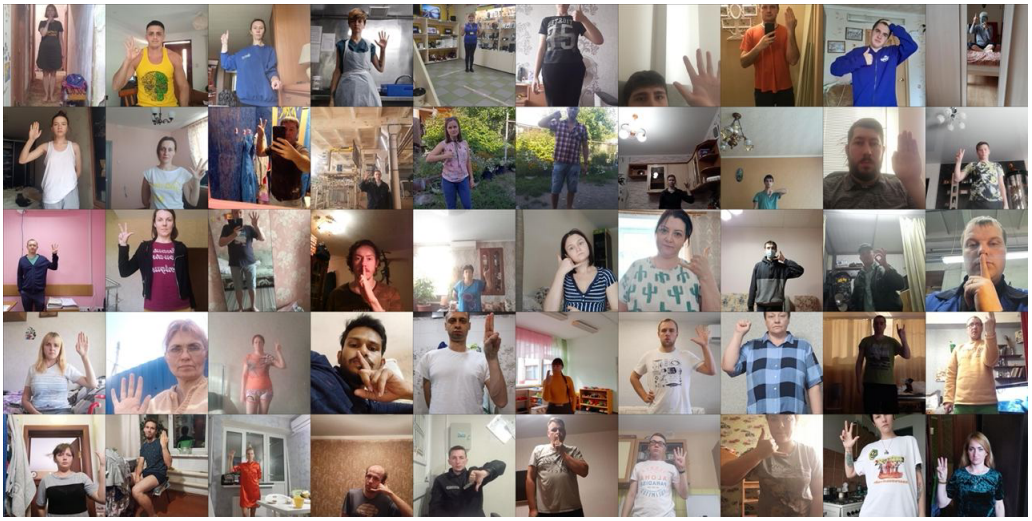


Fig. 4.8: Algunas imágenes del dataset [43].

Capítulo 5

Extracción de características

Las técnicas de extracción de características (FE, por sus siglas en inglés, feature extraction) se han convertido en una necesidad evidente en muchos procesos que tienen mucho que ver con la visión por computadora, la detección y localización de objetos, el procesamiento de imágenes, la recuperación de imágenes, el reconocimiento de voz (SR), la minería de datos, el reconocimiento de patrones, el aprendizaje automático y la bioinformática. Se utiliza para extraer las características más distintivas presentes en un conjunto de datos (imagen, texto, voz) que se utilizan para representar y describir los datos. Los datos son una colección de varias características destacadas. Por otro lado, el procesamiento digital de imágenes es un método de procesamiento que se ocupa de imágenes en color, imágenes binarias e imágenes en escala de grises utilizando una computadora.

La computadora tiene la capacidad de recuperar imágenes de una base de datos correspondiente a un dominio específico, ya sea que estas imágenes tengan o no textura. Este proceso se conoce como recuperación de imágenes (IR, por sus siglas en inglés). La IR se lleva a cabo con el objetivo de encontrar imágenes que compartan características similares o distintas en términos de su espacio de características. Esta tarea se realiza mediante la búsqueda, navegación o recuperación de imágenes en una extensa base de datos compuesta

por diversas imágenes.

El procesamiento de imágenes y la visión por computadora se utilizan para seleccionar características de contenido de una imagen. La extracción de características se divide en dos partes: filtros y envolturas. Los filtros no utilizan aprendizaje automático, mientras que los envoltorios utilizan técnicas de agrupación, clasificación o reconocimiento [73].

Las características desempeñan un papel importante en el ámbito de la visión por computadora o el procesamiento de imágenes para la identificación de información relevante. Antes de extraer las características de la imagen, se realizan varias técnicas de preprocesamiento de imágenes, como normalización, umbralización, binarización, cambio de tamaño, etc., en la imagen constituyente. Las características se clasifican a grandes rasgos en características generales (GF, por sus siglas en inglés, general features) y características específicas del dominio (DSF). Las GF son características autónomas de la aplicación, como el color, la forma y la textura [74], mientras que las DSF dependen de la aplicación e incluyen características conceptuales.

El tema central en este trabajo es crear una propuesta de extracción de características a partir de un modelo de curvas que parte del modelo esquelético de la mano. Este modelo de curvas consiste en utilizar un tipo de curva muy específico que son las curvas bézier, las cuales nos van a ayudar, a partir de sus propiedades, a obtener datos cuantitativos bastante precisos respecto a la forma, posición, orientación, etc.

La propuesta respecto al modelo surgió a partir del trabajo de Ansar et al. [41] en donde en su trabajo utilizan las curvas bézier para hacer el rastreo de una mano en movimiento, donde una curva representa la trayectoria que sigue un punto de la mano respecto a un tiempo determinado. En el trabajo de Ansar et al. también realizan la

construcción de un vector de características en el cual las propiedades y características de las curvas generadas por la trayectoria son añadidas a este vector y sirven como parámetro para distinguir ciertas trayectorias de la mano, eso en conjunto con las otras características como la nube de puntos o el modelado a partir de rombos obtuvieron un modelo bastante preciso.

Gracias al trabajo de Ansar et al. se obtuvo la idea de utilizar las curvas bézier pero no para seguir la trayectoria de un punto, sino como una forma de ver la mano utilizando curvas bézier, es decir, para el caso de los dedos podríamos decir que un dedo es una curva bézier la cual a diferencia de segmentos completamente rectos divididos en ciertos puntos de unión del dedo, trabajar con una curva completa puede llegar a tener ciertas ventajas respecto a su suavidad y aproximación. Esta idea empezó a tener mayor fuerza cuando se observó que además de generar curvas por cada dedo podríamos generar curvas entre puntos de los dedos del mismo nivel, por ejemplo, generar una curva con la punta de cada uno de los dedos; solo en conjunto con estás dos propuestas podríamos precisar mejor respecto a un gesto, sobre qué tanta curvatura tiene un dedo o una curva, y cómo se modifica una curva respecto a los demás dedos.

Para poder obtener este modelo de curvas se deben realizar tres etapas de preprocesamiento las cuales además de ser muy simples tienen una gran utilidad y pueden llegar a darnos rasgos significativos de la mano con los cuales podemos extraer características que se pueden adicionar al modelado de curvas.

En el proceso de extracción de características, atravesamos tres etapas distintas. En primer lugar, se adquiere una imagen de la mano realizando un gesto específico. Luego, en la segunda etapa, se emplea la biblioteca de MediaPipe para procesar esta imagen y obtener 21 puntos en coordenadas (x, y, z), los cuales constituyen una representación del modelo

esquelético de la mano al unir estos puntos para formar su estructura básica. Finalmente, en el tercer y último paso, se genera un modelo de curvas a partir de estos mismos 21 puntos, utilizando como base el modelado esquelético previamente obtenido.

La etapa 2 se conoce como preprocesamiento, mientras que la etapa 3 se refiere a la obtención del vector de características, que implica un procesamiento más exhaustivo basado en el modelado esquelético de la mano. Ambas etapas permiten obtener características significativas; sin embargo, en la tercera etapa, el enfoque se centra específicamente en el modelado de curvas. Estas curvas implican un análisis más profundo y complejo en comparación con la etapa 2.

5.1. Preprocesamiento

En el siguiente diagrama podemos observar cómo el software de MediaPipe identifica la mano y obtenemos 21 puntos en un plano de 3 dimensiones junto con la relación de los puntos de los dedos y la palma. Las gráficas se desarrollaron utilizando las bibliotecas de Matplotlib para la gráfica en 2 dimensiones y Plotly para las gráficas en 3 dimensiones, a partir de la implementación visual de MediaPipe podemos observar con mayor claridad como es que MediaPipe posiciona los puntos y la orientación de los mismos.

En la figura 5.1 se puede observar como es la representación visual de MediaPipe utilizando el graficador Plotly desde diferentes perspectivas con el cual se puede observar la profundidad que MediaPipe detecta respecto a la imagen de una mano de palma que se utilizó como entrada; adicionalmente podemos observar la gráfica generada por Matplotlib en únicamente dos dimensiones.

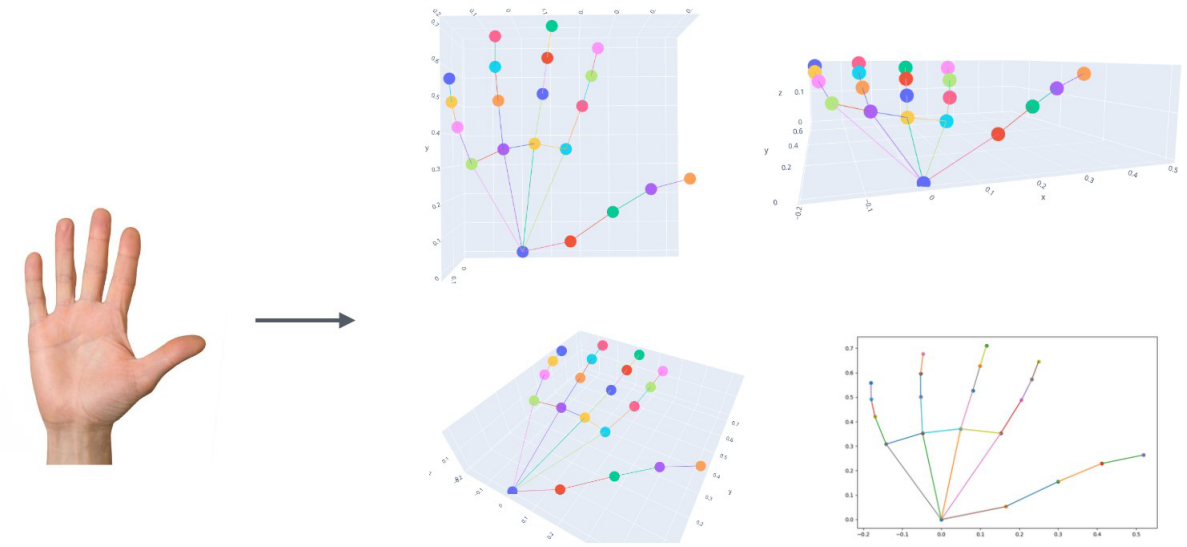


Fig. 5.1: Diferentes perspectivas de la mano utilizando Plotly.

El primer paso del proceso de preprocesamiento implica una transformación lineal simple. El objetivo es ubicar el punto cero, denotado como p_0, que representa la base de la palma de la mano, en el origen de los ejes cartesianos, es decir, en el punto $(0, 0, 0)$. A partir de esta transformación, todos los demás puntos se desplazan y ajustan en consecuencia.

Esta transformación lineal es fundamental para establecer un marco de referencia común y consistente en el análisis. Al posicionar el punto cero en el origen de los ejes, se crea un punto de referencia fijo desde el cual se pueden medir y calcular las coordenadas de los demás puntos de la mano. Esto es especialmente relevante para lograr una representación adecuada y precisa de la geometría de la mano.

Al realizar esta transformación lineal, se asegura que todos los puntos de interés relacionados con la mano se ajusten adecuadamente en relación con la base de la palma. Esto facilita la posterior extracción de características y el modelo de curvas para el análisis

de la mano.

Es importante destacar que esta transformación lineal se lleva a cabo de manera sistemática y repetible, lo que garantiza la consistencia en los resultados obtenidos.

Además, como parte de esta transformación lineal, se realiza un proceso individual para cada punto p_i de la mano. Se aplica la siguiente operación:

$$p_i = (x_i - x_0, y_i - y_0, z_i - z_0)$$

Donde (x_i, y_i, z_i) son las coordenadas originales del punto p_i, y (x_0, y_0, z_0) representa las coordenadas del punto cero, es decir, la base de la palma de la mano ubicada en el origen de los ejes.

Esta operación permite ajustar cada punto p_i en relación a la posición del punto cero. Al restar las coordenadas del punto cero de las coordenadas originales de cada punto, se logra un desplazamiento relativo desde el punto cero, lo que resulta en nuevas coordenadas relativas (x'_i, y'_i, z'_i) para cada punto.

Este proceso garantiza que todos los puntos de la mano se ubiquen correctamente en relación al punto cero, lo que facilita su posterior procesamiento y análisis. Al expresar las coordenadas de los puntos de la mano de forma relativa al punto cero, se establece un sistema de referencia consistente y coherente para todos los puntos, lo que es esencial para realizar cálculos y modelar con precisión las características de la mano.

El segundo paso del preprocesamiento consiste en multiplicar todos los puntos por -1, para que la orientación de la mano tuviera una semejanza mayor a como la vemos, es decir si la mano se encuentra abierta de palma, entonces para todos los puntos se cumple

que para todo p_i, $y_i \geq 0$, entonces tenemos que $p_i = (x_i \times -1, y_i \times -1, z_i \times -1)$.

Una vez terminado el preprocesamiento tenemos un punto base como referencia que es el $(0, 0, 0)$, con esto podemos saber si por cada p_i, $y_i > 0$ entonces los dedos y la palma se encuentran sobre el punto base como en la imagen, si $z_i > 0$ entonces existe una alta posibilidad que la mano se encuentra de palma, y utilizando x_i podemos también apoyarnos respecto a la orientación de la mano, es decir, siempre el punto p_4 va a pertenecer al dedo pulgar, y el punto p_{20} siempre va a pertenecer al dedo meñique, entonces se pueden hacer algunas evaluaciones respecto a si $x_4 > x_{20}$ y con esto obtener información respecto a su orientación, incluyendo los y_i y z_i.

Desde esta etapa temprana de preprocesamiento, podemos obtener ciertas caracteristicas de los puntos dados por media pipe, esta información aunque sea muy simple de observar y calcular nos va a ayudar para el aprendizaje de máquina y así poder diferenciar ciertos gestos o por el contrario relacionarlos.

En la siguiente sección, se brindará una explicación exhaustiva acerca de las curvas Bézier y su relevancia en el contexto de la extracción de características y el modelado de la mano.

Las curvas Bézier, denominadas así en honor al matemático francés Pierre Bézier, son una herramienta fundamental en el campo del modelado y la representación gráfica. Estas curvas se caracterizan por su capacidad de describir formas suaves y precisas mediante el control de puntos de anclaje y puntos de control. La flexibilidad y versatilidad de las curvas Bézier las convierten en un recurso idóneo para representar geometrías complejas, como las estructuras anatómicas de la mano.

En el contexto específico del análisis de la mano, las curvas Bézier desempeñan un

papel fundamental en la extracción de características relevantes. Mediante su utilización, es posible capturar y representar con precisión la curvatura, formas y rotaciones características de la mano. Estas curvas permiten modelar de manera analítica y cuantitativa las diferentes articulaciones y segmentos de la mano, lo que facilita la identificación y el análisis de patrones específicos.

La extracción de características a partir de las curvas Bézier implica la identificación y el cálculo de parámetros relevantes, como la curvatura, la longitud o las rotaciones de la mano. Estos parámetros proporcionan información cuantitativa que puede ser utilizada para distinguir entre diferentes gestos o posturas realizadas por la mano.

Además, el modelado de la mano mediante curvas Bézier permite una representación precisa y compacta de la estructura anatómica de la mano. Estas curvas pueden capturar la forma general de la mano, así como los detalles específicos de cada articulación y dedo.

5.2. Curvas Bézier

Una curva de Bézier es una representación matemática de una curva suave definida por un conjunto de puntos de control. Debe su nombre a Pierre Bézier, un ingeniero francés que la desarrolló en los años 60 mientras trabajaba para el fabricante de automóviles Renault [75].

La curva se construye conectando una serie de puntos de control con líneas rectas y, a continuación, curvando esas líneas mediante funciones matemáticas para crear una curva suave que pase por esos puntos. La forma de la curva viene determinada por la posición y el número de puntos de control utilizados.

De manera más formal las curvas Bézier son splines que permiten al usuario controlar las pendientes en los nudos. A cambio de esta libertad adicional, ya no se garantiza la suavidad de la primera y segunda derivadas a través del nudo, que son características automáticas de las splines cúbicas. Las splines de Bézier son apropiados para los casos en que ocasionalmente se necesitan esquinas (primeras derivadas discontinuas) y cambios bruscos de curvatura (segundas derivadas discontinuas) [76].

Una curva bézier esta formada por un punto de inicio, un punto final, y n puntos de control, en este trabajo nos vamos a enfocar en las curvas bézier cúbicas en 3 dimensiones, esto quiere decir que la curva bézier consta de 4 puntos, un punto de inicio, un punto final, y 2 puntos de control, donde cada punto se encuentra en el plano xyz. Un punto importante a considerar es que la curva bézier no tiene que pasar necesariamente por los puntos de control, idealmente no debería de pasar por algún punto de control.

La forma paramétrica de una curva bézier cúbica se define como:

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3,\ t \in [0,1]$$

En la figura 5.2 se pueden observar dos curvas bézier cúbicas, las cuales consisten en un punto de inicio, un punto final y dos puntos de control, los cuales afectan la trayectoria de la curva.

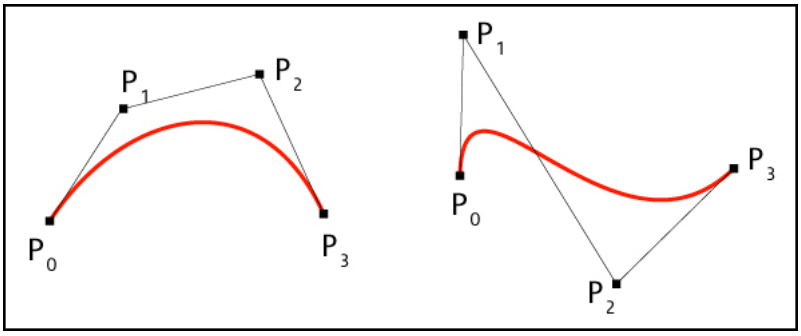


Fig. 5.2: Ejemplo de curvas bézier cubicas [77].

Para entender mejor que es una curva bézier, en la siguiente sección explicaré que son las splines, en que consisten y para qué se utilizan.

5.2.1. Splines

Las splines representan un enfoque alternativo a la interpolación de datos. En la interpolación polinómica, se utiliza una única fórmula, dada por un polinomio, para pasar por todos los puntos de datos. La idea de las splines es utilizar varias fórmulas, cada una de ellas un polinomio de bajo grado, para pasar por los puntos de datos.

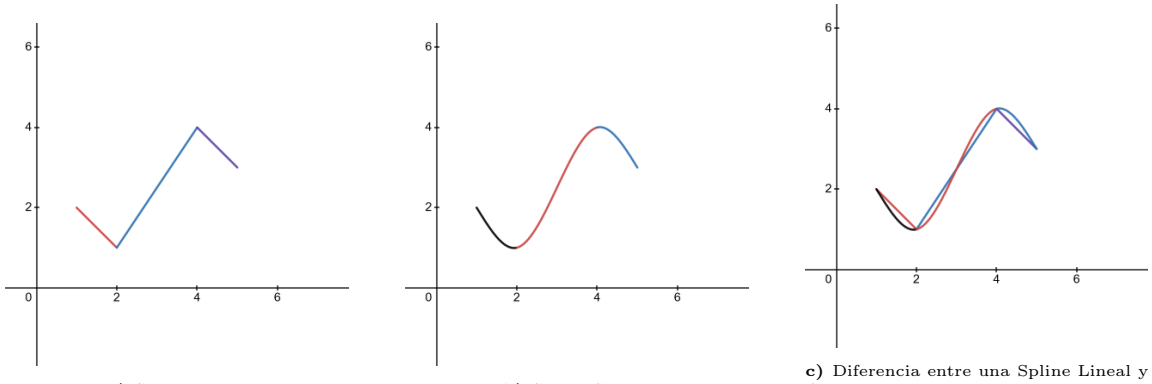
El ejemplo más sencillo de spline es un spline lineal, en el que se ”conectan los puntos” con segmentos rectilíneos. Supongamos que tenemos un conjunto de puntos de datos $(x_1, y_1), \ldots, (x_n, y_n)$ con $x_1 < \cdots < x_n$. Un spline lineal consiste en $n-1$ segmentos de línea que se dibujan entre pares de puntos vecinos.

El spline lineal interpola con éxito un conjunto arbitrario de n puntos de datos. Sin embargo, las splines lineales carecen de suavidad. Las splines cúbicos están pensados para solventar esta deficiencia de las splines lineales.

Una spline cúbica es una spline construida con polinomios de tercer orden por partes que pasan por un conjunto de m puntos de control. La segunda derivada de cada polinomio se suele poner a cero en los puntos finales, ya que esto proporciona una condición de contorno que completa el sistema de $m - 2$ ecuaciones. Se obtiene así un spline cúbico 'natural' y un sistema tridiagonal sencillo que puede resolverse fácilmente para obtener los coeficientes de los polinomios.

En la siguiente imagen podemos ver una spline lineal, una spline cúbica y ambas en una misma gráfica. Podemos observar que la spline cúbica, tiene cierta curvatura respecto a los puntos, pero al igual que la spline lineal sigue pasando por lo puntos; esa es una característica de las splines cúbicas que tendremos en cuenta y compararemos con las curvas bézier [76].

En la figura 5.3 se pueden observar dos tipos de Splines que son la Spline Lineal y la Spline Cúbica, se puede observar la suavidad de la curva de la Spline Cúbica a comparación con las lineas rectas de la Spline Lineal; en ambos casos la Spline pasa por todos los puntos.



a) Spline Lineal

b) Spline Cúbica

c) Diferencia entre una Spline Lineal y Cúbica

Fig. 5.3: Ejemplos de Splines.

5.2.2. B-Splines

Las splines son polinomios por partes que se conectan suavemente entre sí. Los puntos de unión de los polinomios se llaman nudos. Para un spline de grado n, cada segmento es un polinomio de grado n, lo que sugeriría que necesitamos $n+1$ coeficientes para describir cada trozo. Sin embargo, existe una restricción de suavidad adicional que impone la continuidad del spline y sus derivadas hasta el orden $n-1$ en los nudos, de modo que, efectivamente, sólo hay un grado de libertad por segmento. Este tipo de splines se caracterizan de forma única en términos de una expansión B-spline.[78]

$$s(x) = \sum_{k \in Z} c(k)\beta^n(x-k)$$

Esta definición implica que los desplazamientos enteros de la B-spline central de grado n denotada por $\beta^n(x)$; los parámetros del modelo son los coeficientes de la B-spline $c(k)$. Las B-splines, definidas a continuación, son funciones simétricas en forma de campana construidas a partir de la convolución $(n+1) - fold$ de un pulso rectangular β^0

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0 & else \end{cases}$$

$$\beta^n(x) = \beta^0 \cdot \beta^0 \cdots \beta^0(x) \ (n+1 \text{ veces})$$

Dado que el modelo B-spline es lineal, el estudio de las propiedades básicas puede decirnos mucho sobre las splines en general. Gracias a esta representación, cada spline se

caracteriza inequívocamente por su secuencia de coeficientes B-spline $c(k)$, que tiene la conveniente estructura de una señal discreta, aunque el modelo subyacente sea continuo (representación discreta/continua). Una B-spline es simplemente una generalización de una curva de Bézier.

En la figura 5.4 se puede observar un ejemplo de una B-spline la cual es una curva que es afectada por 4 puntos de control, únicamente pasando por el punto de inicio y fin.

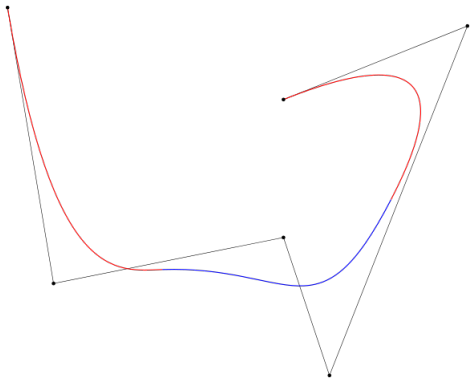


Fig. 5.4: Ejemplo de B-spline [79].

5.2.3. Curvatura

Una curva espacial definida paramétricamente en tres dimensiones dada en coordenadas cartesianas por $\gamma(t) = (x(t), y(t), z(t))$, la curvatura se define como:

$$\kappa = \frac{\sqrt{(z''y' - y''z')^2 + (x''z' - z''x')^2 + (y''x' - x''y')^2}}{((x')^2 + (y')^2)^{\frac{3}{2}}}$$

donde la comilla denota la derivada con respecto al parámetro t. Para obtener la curvatura de una curva bézier cúbica en 3 dimensiones simplemente debemos de definir la $\gamma(t)$ como $B(t)$.

κ mide la desviación directa de la curva, es decir, la magnitud de κ es la medida de cuánto se desvía de la tangente en un punto de la curva. Sea θ el ángulo entre la tangente en el punto P de la curva y el eje x. La curvatura de la curva es el valor absoluto del cambio de θ a lo largo del arco unitario.

En la figura 5.5 se puede observar gráficamente la definición de la curvatura de una curva la cual es el ángulo entre la tangente y en este caso el eje x, respecto a un punto de la curva.

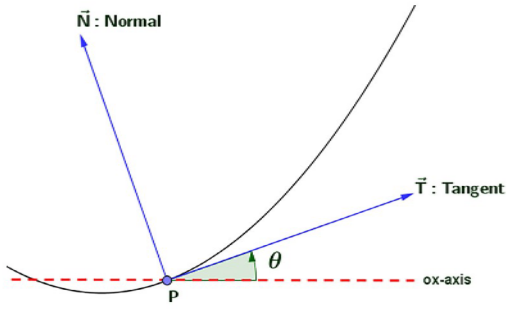


Fig. 5.5: La curvatura de una curva [80].

La curvatura de una curva Bézier se presenta como una característica fundamental en el proceso de identificación y diferenciación de gestos. En esta sección se ha proporcionado una visión general de cómo obtener la curvatura de una curva, particularmente en el caso de las curvas Bézier. Ahora, se explorará cómo esta información puede ser aplicada al vector de características.

La curvatura de una curva Bézier es una medida cuantitativa que describe la tendencia de la curva a curvarse en cada punto a lo largo de su trayectoria. Esta característica es especialmente valiosa en el análisis de gestos, ya que proporciona información relevante sobre la forma de la mano.

Al incorporar la curvatura como una característica en el vector de características, se obtiene un enfoque más completo y detallado en el modelado de la mano. Esto permite una mayor capacidad de discriminación entre diferentes gestos, ya que la curvatura puede variar significativamente entre distintos movimientos. Por ejemplo, un gesto con una curvatura más pronunciada puede indicar una posición específica de la mano, mientras que una curvatura más suave puede representar otro tipo de gesto.

Una vez obtenida la curvatura en cada punto, esta información puede ser incorporada al vector de características junto con otras medidas, como las coordenadas de los puntos y el ángulo de rotación.

5.2.4. Ángulo de rotación

Definimos el ángulo de rotación de una curva bézier como el ángulo de su vector tangente en un punto $t \in [0, 1]$ respecto a cada uno de los ejes en R^3.

Primero definimos los ejes de la siguiente forma:

$$e_1 = (1, 0, 0)$$

$$e_2 = (0, 1, 0)$$

$$e_3 = (0, 0, 1)$$

También definimos la curva bézier como una función paramétrica $B(t)$ y su derivada como $B'(t)$.

Ahora utilizaremos la ecuación del ángulo entre dos vectores para obtener el ángulo

de rotación respecto a cada uno de los ejes, entonces tenemos que para el eje x:

$$cos(\theta_1) = \frac{B'(t) \cdot e_1}{||B'(t)|| \cdot ||e_1||} = \frac{(x'(t), y'(t), z'(t)) \cdot (1, 0, 0)}{\sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}} = \frac{x'(t)}{\sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}}$$

$$\theta_1 = \arccos(\frac{x'(t)}{\sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}})$$

Y para los ejes y y z, tenemos que:

$$\theta_2 = \arccos(\frac{y'(t)}{\sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}})$$
$$\theta_3 = \arccos(\frac{z'(t)}{\sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}})$$

Por cada punto en t de la curva bézier $B'(t)$ tenemos los 3 ángulos de la curva respecto a cada uno de los ejes.

5.2.5. Triangulo de Bézier

Un triángulo de Bézier es un tipo especial de superficie de Bézier que se crea por interpolación (lineal, cuadrática, cúbica o de grado superior) de puntos de control. Vamos a definir un triángulo de Bézier como un mapeo desde el triangulo unitario en R^2 hacia un triángulo en una dimensión arbitraria. Utilizamos coordenadas baricéntricas

$$\lambda_1 = 1 - s - t, \lambda_2 = s, \lambda_3 = t$$

para los puntos del triángulo unitario $\{(s,t)|0 \leq s, t, s + t \leq 1\}$. Utilizando estos pesos obtenemos combinaciones convexas de puntos $v_{i,j,k}$ en algún espacio vectorial:

$$B(\lambda_1, \lambda_2, \lambda_3) = \sum_{i+j+k=d} \binom{d}{ijk} \lambda_1^i \lambda_2^j \lambda_3^k \cdot v_{i,j,k}$$

Con orden lineal ($n = 1$), el triángulo de Bézier resultante es en realidad un triángulo plano regular, con los vértices del triángulo iguales a los tres puntos de control. Un triángulo de Bézier cuadrático ($n = 2$) presenta 6 puntos de control, todos ellos situados en las aristas. El triángulo de Bézier cúbico ($n = 3$) está definido por 10 puntos de control y es el triángulo de Bézier de orden más bajo que tiene un punto de control interno, no situado en las aristas. En todos los casos, las aristas del triángulo serán curvas de Bézier del mismo grado.

Un triángulo cúbico de Bézier puede expresarse de una forma más general:

$$p(s,t,u) = \sum_{i+j+k=3, i,j,k \geq 0} \binom{3}{ijk} s^i t^j u^k \alpha^i \beta^j \gamma^k = \sum_{i+j+k=3, i,j,k \geq 0} \frac{6}{i!j!k!} s^i t^j u^k \alpha^i \beta^j \gamma^k$$

de acuerdo con la formulación del § triángulo de Bézier de enésimo orden.

Los vértices del triángulo son los puntos α^3, β^3 y γ^3. Las aristas del triángulo son a su vez curvas de Bézier, con los mismos puntos de control que el triángulo de Bézier.

Al eliminar el término γu, se obtiene una curva de Bézier regular. Además, aunque no es muy útil para la visualización en una pantalla física de computadora, añadiendo términos adicionales se obtiene un tetraedro de Bézier o un politopo de Bézier.

Debido a la naturaleza de la ecuación, todo el triángulo estará contenido dentro del volumen rodeado por los puntos de control, y las transformaciones afines de los puntos de control transformarán correctamente todo el triángulo de la misma manera.

En la figura 5.6 se puede observar gráficamente el comportamiento del triangulo de bézier el cual genera una curvatura respecto a puntos de control, como es con las curvas bézier.

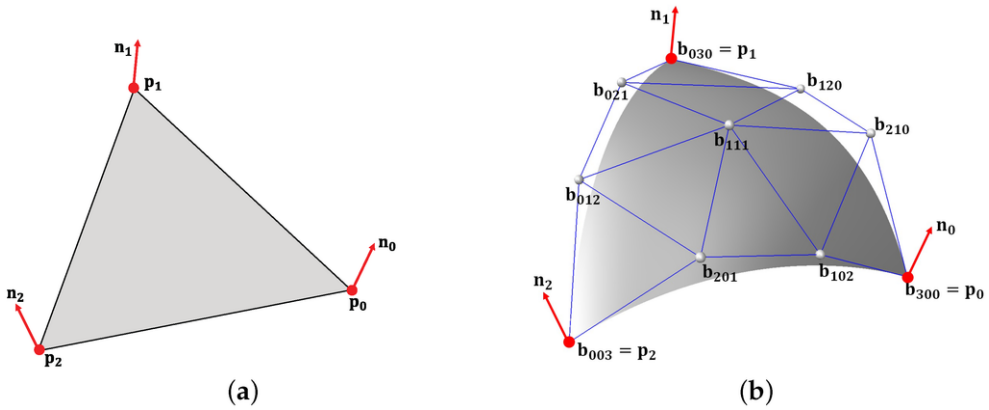


Fig. 5.6: Ejemplo del triangulo de bézier [81].

El ángulo de rotación de una curva Bézier es una característica esencial en la identificación y diferenciación de gestos. Su inclusión en el vector de características permite un modelado más completo y detallado de la mano. El cálculo y registro de los ángulos de rotación en cada punto de la curva posibilita capturar la orientación y la postura de la mano, mejorando así la capacidad de discriminación entre diferentes gestos en el análisis.

5.3. Modelado de la mano

El objetivo de este trabajo es proponer una nueva forma de modelar la mano a partir de curvas bézier, triángulos de bézier, además del ángulo de rotación y curvatura, etc. y así tener una forma mucho más precisa para diferenciar un gesto de otro.

Una primera parte del modelado de la mano consiste en crear curvas bézier a partir de los dedos de la mano, cada dedo esta conformado por 4 puntos en 3 dimensiones, suponiendo que el punto de inicio es la base del dedo, el punto final la punta del dedo, y los otros dos puntos los puntos de control, crearíamos un curva bézier por cada dedo.

Por otro lado también vamos a crear curvas bézier de forma 'horizontal' esto quiere decir que los puntos de la curva bézier van a estar conformados por el nivel de cada uno de los dedos índice, dedo medio, anular y meñique; esto es por ejemplo un curva se conforma por la punta de los dedos índice, dedo medio, anular y meñique, donde el punto de inicio es la punta del dedo meñique, el punto final es la punta del dedo índice, y los puntos de control son la punta del dedo medio y el anular, este modelado se maneja por los niveles de los dedos.

Por último, para cubrir toda la mano como una superficie, utilizaremos el triángulo de bézier para el cual los puntos que forman el triángulo son la base de la palma, es decir el origen $(0, 0, 0)$, y cada uno de los puntos de las puntas de los dedos. Recordemos que el orden de los puntos en las curvas y en el triangulo de bézier es importante por lo que el orden de los puntos de las puntas de los dedos será del meñique hasta el pulgar, siendo el punto de inicio la base de la palma y el punto final la punta del dedo pulgar.

En resumen, la propuesta del modelado de la mano a partir de curvas bézier consiste en 5 curvas bézier 'verticales' una por cada dedo, 4 curvas bézier horizontales, y 1 triangulo de bézier que cubre toda la mano. Con este simple modelado de 10 elementos matemáticos, a cada uno de ellos se les va a extraer ciertos parámetros discutidos anteriormente para alimentar un vector de características el cual posteriormente se utilizara para hacer aprendizaje de maquina supervisado.

En la figura 5.8 se pueden observar las gráficas de dos gestos de la mano distintos y sus diferentes curvas bézier, el primer gesto es una mano abierta de palma a la cuál se le pueden observar las curvas bézier por cada uno de los dedos y las curvas bézier horizontales por cada nivel de cada dedo. El segundo gesto es una mano haciendo la seña de 'rock' o 'cuernos' el cual a diferencia del gesto anterior es que el dedo anular y el dedo mayor se

encuentran doblados hacia la palma, el cambio de estos dos dedos afectaron completamente las curvas bézier horizontales a comparación de las curvas bézier horizontales del gesto de palma, lo cual es lo esperado.

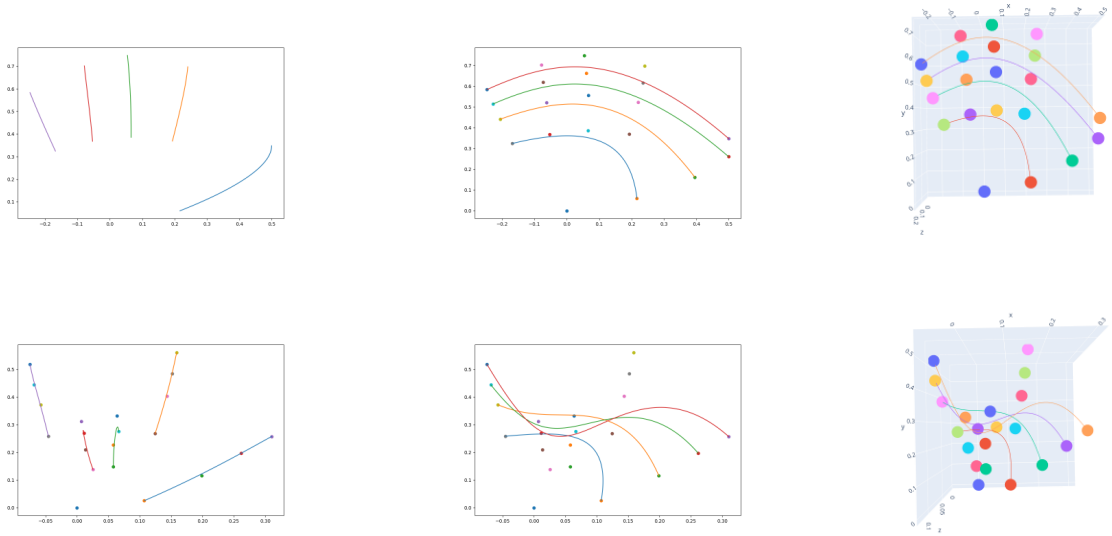


Fig. 5.8: Representación de la mano utilizando curvas bézier.

5.4. Vector de Características

El vector de características que se va a utilizar para hacer aprendizaje automático está conformado por: las coordenadas (x, y, z) de cada uno de los 21 puntos, la curvatura de las curvas bézier, la longitud de las curvas bézier, el ángulo de rotación respecto a cada uno de los ejes (x, y, z) en algún punto de la curva y el área formada por el triangulo de bézier, rotando el orden de los puntos.

Cada una de las coordenadas de los 21 puntos de la mano las agregamos al vector de características, pero cada coordenada de cada punto de manera independiente, es decir, $[x_1, y_1, z_1, x_2, y_2, z_2, \dots]$.

La curvatura de las curvas bézier de la mano, definidas previamente, son 9 curvas en total, de cada curva obtenemos la curvatura en varios puntos de la curva, estos puntos los definimos respecto al parámetro t, donde recordemos que $t \in [0, 1]$, para fines de este trabajo utilizaremos 20 valores que pertenecen a $[0, 1]$, si se requiere mayor precisión es posible aumentar el conjunto de valores para obtener más puntos de curvatura a lo largo de la misma.

Así mismo con la longitud de cada una de las curvas, las agregamos al vector de características.

El ángulo de rotación respecto a cada una de la curvas se define con los mismos 20 valores que van de $[0, 1]$ para el parámetro, debido a que así como en la curvatura, el ángulo de rotación depende en que punto de la curva te encuentres. Además el ángulo de rotación es respecto al eje x, y, y z.

Por último al vector de características agregamos el área formada por los 6 puntos extremos en la mano, es decir, la punta de cada uno de los dedos y la base de la mano. El triangulo de bézier depende fuertemente del orden de los puntos debido a los puntos de inicio, fin y de control; por lo que para mantener la estructura de la mano se rotan los puntos sobre el mismo orden.

Las características más significativas dentro del vector son: las coordenadas de los puntos, la curvatura y el ángulo de rotación. Estas características permiten cubrir la posición espacial de la mano, la forma de la mano y su orientación. Estos elementos resultan fundamentales para distinguir gestos en función del contexto de entrenamiento y los objetivos del programa.

La flexibilidad de ajustar las características del vector es de suma importancia para

adaptarse a las necesidades específicas. Por ejemplo, si la orientación de la mano no es relevante, pero se desea mantener la misma forma, se puede omitir el ángulo de las curvas en relación con cada uno de los ejes en el vector de características. De esta manera, es posible analizar y contextualizar el uso del vector de características de acuerdo a los requerimientos del sistema.

Al seleccionar las características más relevantes y ajustar su inclusión en el vector, se logra un enfoque más preciso y específico para el análisis de gestos y la identificación de patrones de la mano. Estas características clave permiten capturar de manera efectiva la información necesaria para la clasificación y reconocimiento de gestos.

Las otras características funcionan como complemento a las características que son más significativas, a partir de estos 'complementos' es como se pueden llegar a diferenciar ciertas entradas que son muy parecidas y por lo mismo ayuda a la precisión del modelo. Cabe destacar que en volumen de datos en el vector es mucho menor que las características que son más significativas.

El enfoque central de esta investigación se centra en la propuesta de un vector de características como método para abordar el modelado analítico de la mano. Este enfoque utiliza objetos matemáticos que poseen propiedades y características que pueden ser extraídas, lo que permite un preprocesamiento exhaustivo de los datos de entrada. El objetivo principal es proporcionar una alternativa al uso de imágenes de gestos de la mano en el entrenamiento de algoritmos de aprendizaje supervisado o no supervisado.

El vector de características utilizado en este enfoque permite evitar la necesidad de depender exclusivamente de imágenes de gestos de la mano para el entrenamiento. En lugar de ello, se enfoca en analizar y controlar las características del vector. Esto evita el concepto

de 'caja negra' en el campo de la inteligencia artificial, donde los patrones utilizados por los algoritmos para hacer predicciones sobre los datos son desconocidos.

La propuesta se centra en identificar patrones y formas de modelar la mano utilizando herramientas analíticas y controlables. Al construir características explícitas y detalladas, se busca lograr un modelado más preciso y adecuado de la mano. Estas características pueden ser analizadas y ajustadas de manera sistemática, permitiendo un mayor control sobre el proceso de entrenamiento de los algoritmos de aprendizaje automático.

Capítulo 6

Aprendizaje automático

En este capítulo, se lleva a cabo la evaluación de la propuesta para el modelado de la mano, empleando el aprendizaje automático, más concretamente, mediante el uso de redes neuronales. El objetivo principal de esta prueba consiste en demostrar la eficacia y viabilidad de dicha propuesta.

El enfoque adoptado en esta investigación implica el aprovechamiento de técnicas avanzadas de aprendizaje automático, específicamente el uso de redes neuronales, para capturar y representar de manera precisa la compleja estructura de la mano humana. Mediante este proceso, se busca proporcionar un enfoque innovador y efectivo para el modelado de la mano.

La relevancia de esta investigación radica en la búsqueda de soluciones que superen las limitaciones de los métodos tradicionales de modelado de la mano. En contraste, el enfoque basado en redes neuronales ofrece la promesa de una mayor flexibilidad y adaptabilidad, permitiendo una representación más fiel de la anatomía y movimientos de la mano. A continuación se da una breve explicación sobre qué es el aprendizaje automático, las redes neuronales y cómo está conformado el entrenamiento con el cual se va a hacer la evaluación de la propuesta.

El aprendizaje automático es una rama en evolución de los algoritmos computacionales que están diseñados para emular la inteligencia humana aprendiendo del entorno circundante. Las técnicas basadas en el aprendizaje automático se han aplicado con éxito en diversos campos que van desde el reconocimiento de patrones, la visión por ordenador, la ingeniería de naves espaciales, las finanzas, el entretenimiento y la biología computacional hasta las aplicaciones biomédicas y médicas.

Un algoritmo de aprendizaje automático es un proceso computacional que utiliza datos de entrada para lograr una tarea deseada sin ser programado literalmente (es decir, ”codificado” de forma rígida) para producir un resultado específico. Estos algoritmos están, en cierto sentido, ”codificados de forma flexible” ya que automáticamente modifican o adaptan su arquitectura a través de la repetición (es decir, la experiencia) para volverse cada vez mejores en la realización de la tarea deseada. El proceso de adaptación se llama entrenamiento, en el cual se proporcionan muestras de datos de entrada junto con los resultados deseados. Luego, el algoritmo se configura óptimamente para no solo producir el resultado deseado cuando se le presentan las entradas de entrenamiento, sino también generalizar y producir el resultado deseado a partir de nuevos datos no vistos anteriormente. Este entrenamiento es la parte de ”aprendizaje” del aprendizaje automático. El entrenamiento no tiene que limitarse a una adaptación inicial durante un intervalo finito. Al igual que los seres humanos, un buen algoritmo puede practicar el aprendizaje ”de por vida” a medida que procesa nuevos datos y aprende de sus errores.[82]

Existen muchas formas en las que un algoritmo computacional puede adaptarse en respuesta al entrenamiento. Los datos de entrada pueden ser seleccionados y ponderados para proporcionar resultados más decisivos. El algoritmo puede tener parámetros numéricos variables que se ajustan mediante una optimización iterativa. Puede tener una red de

posibles trayectorias computacionales que se organizan para obtener resultados óptimos. Puede determinar distribuciones de probabilidad a partir de los datos de entrada y utilizarlas para predecir resultados.

El ideal del aprendizaje automático es emular la forma en que los seres humanos (y otras criaturas conscientes) aprenden a procesar señales sensoriales (de entrada) para lograr un objetivo. Imaginemos una tarea de reconocimiento de patrones en la que queremos enseñar a una máquina a distinguir entre flores de diferentes especies. Cada flor es única en su apariencia, pero podemos aprender a diferenciarlas. En lugar de programar a la máquina con múltiples representaciones exactas de cada especie de flor, podemos entrenarla mediante ejemplos reales. Este enfoque se conoce como aprendizaje supervisado, donde proporcionamos a la máquina muestras de entrenamiento con características como color, forma y tamaño, junto con la etiqueta de la especie correspondiente. De esta manera, la máquina aprende a reconocer las similitudes y diferencias entre las flores, incluso cuando presentan propiedades variables dentro de cada especie, pero mantienen características fundamentales que las identifican. Lo más importante es que una máquina bien entrenada debería ser capaz de reconocer una flor de una especie nunca antes vista.

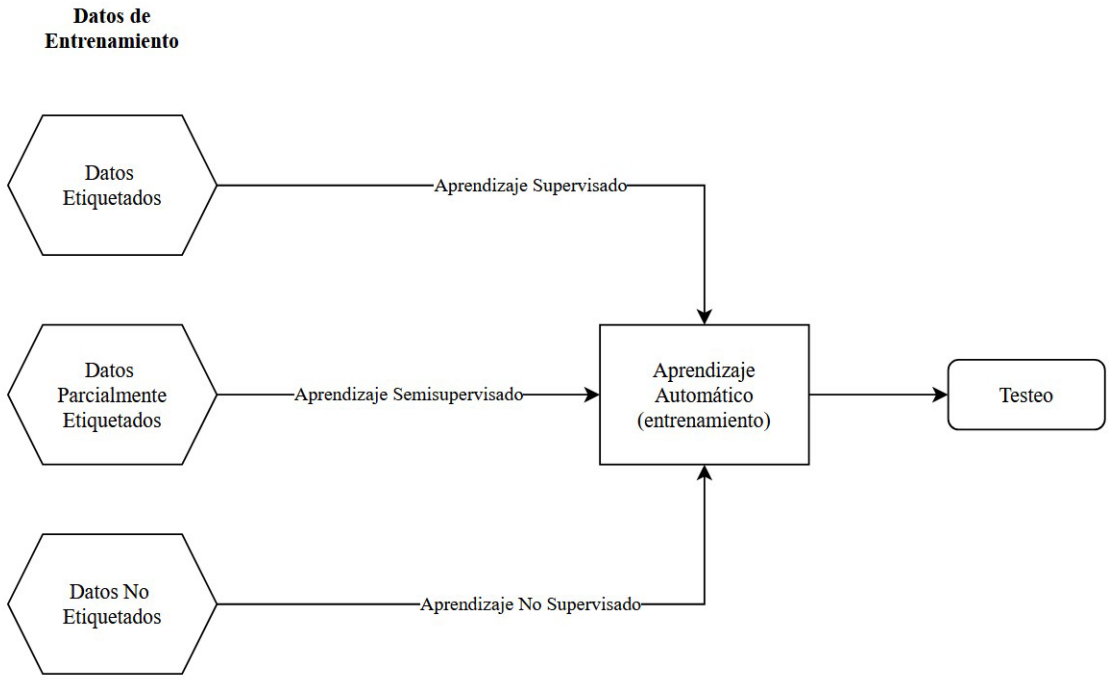


Fig. 6.1: Categorías de algoritmos de aprendizaje automático [82].

El segundo tipo de aprendizaje automático es el aprendizaje no supervisado, que se utiliza para abordar diversos desafíos. Un ejemplo para comprenderlo mejor es imaginar que queremos lanzar un dardo al centro de una diana. En este caso, el dispositivo o incluso una persona puede tener una variedad de grados de libertad en el mecanismo que controla la trayectoria del dardo.

En el aprendizaje automático no supervisado, a diferencia del supervisado, no contamos con ejemplos etiquetados para guiar el proceso. En lugar de eso, el algoritmo busca patrones y estructuras dentro de los datos sin que se le indique explícitamente qué buscar. Es como si el sistema tuviera que descubrir por sí mismo la mejor forma de acertar en el centro de la diana sin recibir instrucciones precisas sobre cómo hacerlo.

Los algoritmos no supervisados pueden ser de gran utilidad en situaciones donde no tenemos información previa o etiquetas para entrenar al modelo. Son capaces de explorar y descubrir relaciones ocultas dentro de los datos, lo que los convierte en una herramienta valiosa para la detección de anomalías, agrupación de elementos similares y reducción de dimensiones en conjuntos de datos complejos.

Un tercer tipo de aprendizaje automático es el aprendizaje semisupervisado, donde parte de los datos están etiquetados y otras partes no lo están. En este escenario, la parte etiquetada se puede utilizar para ayudar en el aprendizaje de la parte no etiquetada. Este tipo de escenario se asemeja más a los procesos de la naturaleza y emula de manera más cercana cómo los seres humanos desarrollan sus habilidades. En la figura 6.1 se pueden observar los distintos tipos de aprendizaje automático con su respectivo conjunto de datos de entrenamiento.

El término ”redes neuronales” se refiere históricamente a las redes de neuronas en el cerebro de los mamíferos. Las neuronas son las unidades fundamentales de cálculo y se conectan entre sí en redes para procesar datos. Esto puede ser una tarea muy compleja. La dinámica de estas redes neuronales en respuesta a estímulos externos a menudo es bastante intrincada. Las entradas y salidas de cada neurona varían en forma de trenes de impulsos a lo largo del tiempo, pero también la red misma cambia con el tiempo: aprendemos y mejoramos nuestras capacidades de procesamiento de datos estableciendo nuevas conexiones entre neuronas. En la figura 6.2 podemos observar un esquema simplificado de una neurona y sus componentes principales que son el núcleo, el axon, la sinapsis, las dendritas y el cuerpo de la célula.

Los algoritmos de redes neuronales para el aprendizaje automático se inspiran en la arquitectura y la dinámica de las redes de neuronas en el cerebro. Los algoritmos utilizan

modelos de neuronas altamente idealizados. Sin embargo, el principio fundamental es el mismo: las redes neuronales artificiales aprenden al cambiar las conexiones entre sus neuronas. Estas redes pueden realizar una multitud de tareas de procesamiento de información.[83]

Por ejemplo, las redes neuronales pueden aprender a reconocer estructuras en un conjunto de datos y, hasta cierto punto, generalizar lo que han aprendido. Un conjunto de entrenamiento contiene una lista de patrones de entrada junto con una lista de etiquetas correspondientes, o valores objetivo, que codifican las propiedades de los patrones de entrada que se supone que la red debe aprender. Las redes neuronales artificiales pueden entrenarse para clasificar estos datos de manera muy precisa ajustando las fuerzas de conexión entre sus neuronas, y pueden aprender a generalizar los resultados a otros conjuntos de datos, siempre y cuando los nuevos datos no sean demasiado diferentes de los datos de entrenamiento. Un ejemplo destacado de un problema de este tipo es el reconocimiento de objetos en imágenes, por ejemplo, en una secuencia de imágenes capturadas por un automóvil autónomo.

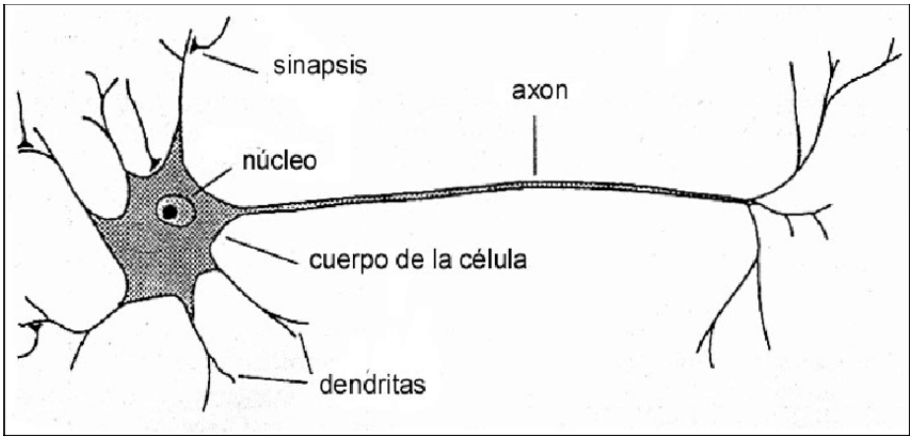


Fig. 6.2: Esquema de una neurona [84]

El interés reciente en el aprendizaje automático con redes neuronales se debe en parte al éxito de las redes neuronales en el reconocimiento visual de objetos.

En este trabajo, se utilizarán las redes neuronales como métrica y tipo de aprendizaje automático para evaluar la eficiencia, precisión y usabilidad del vector de características en la distinción de diferentes gestos de la mano. La red neuronal tomará como entrada datos cuantitativos numéricos, aproximadamente 450 valores por imagen, que corresponden a métricas de las curvas Bezier, como la curvatura, la rotación, las distancias entre los puntos y las coordenadas de los puntos, entre otros.

6.1. Redes Neuronales Artificiales

Las redes neuronales artificiales es el nombre que se le da a una rama de investigación de inteligencia artificial que tiene como objetivo simular el comportamiento inteligente imitando la forma en que funcionan las redes neuronales biológicas. Mientras que la mayoría de los métodos de inteligencia artificial buscan reproducir la inteligencia humana imitando lo que hacemos, las redes neuronales artificiales buscan reproducirla imitando la forma en que lo hacemos. Los orígenes de las redes neuronales artificiales preceden a la existencia de las computadoras por varias décadas, pero no fue hasta que las computadoras se volvieron ampliamente disponibles que se pudo lograr un verdadero progreso en el desarrollo de estos métodos. Hubo un breve periodo de estancamiento de aproximadamente una década después de la publicación de un libro que criticaba fuertemente la posibilidad de que las redes neuronales artificiales se desarrollaran en algo útil; sin embargo, desde entonces, ha habido un progreso espectacular y estas herramientas han pasado de ser rarezas utilizadas por especialistas a algoritmos de propósito general para el análisis de datos y tareas de reconocimiento de patrones.[85]

Una red neuronal artificial puede ser una máquina comparable producida para funcionar de la misma manera en que el cerebro humano realiza una tarea de interés. El

cerebro humano es como una máquina de procesamiento de información que tiene una variedad de operaciones de computación de señales complejas, que pueden coordinarse fácilmente para realizar una tarea. El elemento principal de este cerebro es el diseño único de su capacidad de procesamiento de información. Consta de muchas "neuronas" interconectadas de forma compleja que trabajan juntas para resolver problemas específicos a diario. Un ejemplo típico de la función de una red neuronal es el cerebro humano, que está conectado para enviar y recibir señales para la acción humana.[86]

El aprendizaje en el cerebro humano requiere ajustes en la relación sináptica entre y entre las neuronas, de manera similar al aprendizaje en las redes neuronales artificiales. En general, una red neuronal artificial funciona como una imitación del cerebro humano.

En la figura 6.3 podemos observar un esquema de una red neuronal de alimentación, donde la capa de entrada tiene cuatro unidades, la capa oculta tiene tres unidades y la capa de salida tiene solo una unidad; el número de unidades y el número de capas ocultas varia respecto a como esté estructurada la red neuronal y a los datos de entrada.

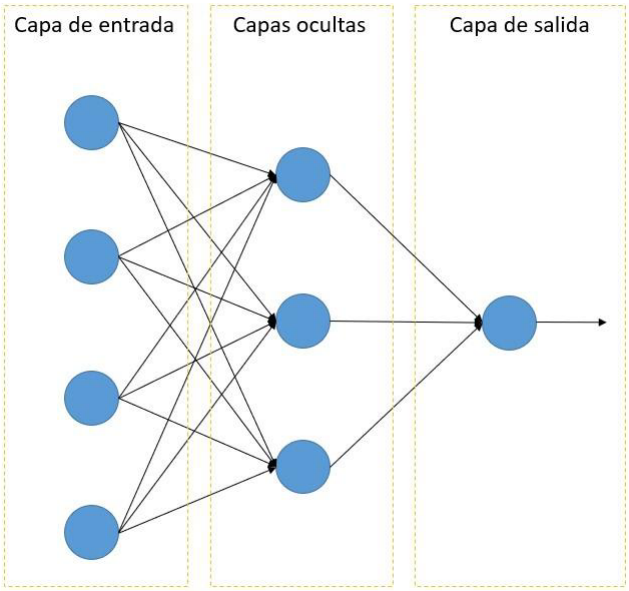


Fig. 6.3: Red neuronal de alimentación [86].

En la figura 6.4 podemos observar el esquema de una red neuronal de retroalimentación, donde a diferencia de una red neuronal de alimentación, se repite el entrenamiento entre las mismas capas, es decir, la información procesada en la capa c tiene una salida p, y p vuelve a ser procesada por c para producir la salida q. Debido a la gran variedad de capas y unidades dentro de la red la retroalimentación varia dependiendo sobre que capa y que unidad se hace la retroalimentación del aprendizaje.

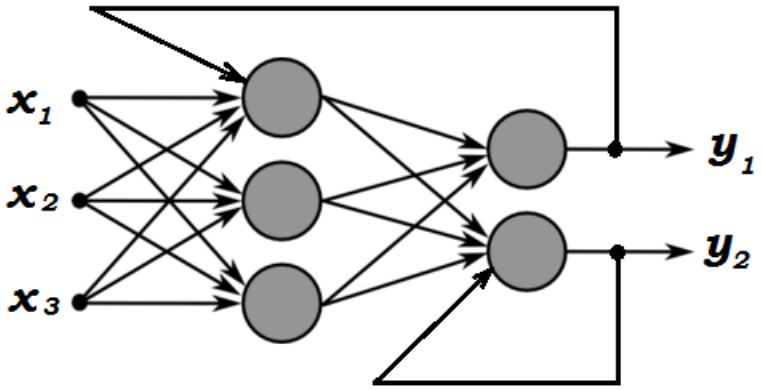


Fig. 6.4: Red neuronal de retroalimentación [87].

Las capas de la red neuronal son independientes entre sí, es decir, una capa específica puede tener un número arbitrario de nodos. Este número arbitrario de nodos se llama nodo de sesgo. Los nodos de sesgo siempre se establecen en uno, una función principal del sesgo es proporcionar a los nodos un valor constante que se puede entrenar, además de las entradas normales que recibe el nodo de la red. Es importante destacar que un valor de sesgo permite mover la función de activación hacia la derecha o hacia la izquierda, lo cual puede ser analíticamente importante para el éxito del entrenamiento de una red neuronal artificial. Cuando se utiliza la red neuronal como clasificador, los nodos de entrada y salida coincidirán con las características de entrada y las clases de salida.

6.1.1. Clasificación de redes neuronales artificiales

Una red neuronal de alimentación directa es un algoritmo de clasificación de aprendizaje automático compuesto por unidades de procesamiento organizadas en capas similares a las neuronas humanas; cada unidad en una capa se relaciona con todas las demás unidades

en las capas. Estas conexiones entre capas y unidades no son todas iguales, ya que cada conexión puede tener un peso o fuerza diferente. Los pesos de las conexiones de la red miden la cantidad potencial de conocimiento de la red. Además, las unidades de la red neuronal se conocen como nodos.

El procesamiento de información en la red implica la entrada de datos desde las unidades de entrada y pasa a través de la red, fluyendo de una capa a otra hasta llegar a las unidades de salida. Cuando la red neuronal actúa como clasificador, no hay retroalimentación entre las capas. En una red neuronal de alimentación directa, la información se transmite solo en una dirección, es decir, desde los nodos de entrada, a los nodos ocultos, si los hay, y luego a los nodos de salida. Con este comportamiento, se les llama redes neuronales de alimentación directa. Ejemplos de redes neuronales de alimentación directa son el perceptrón de una sola capa y el perceptrón multicapa.

En la figura 6.5 se puede observar un diagrama con la clasificación de las redes neuronales, donde se dividen en dos, las redes neuronales de alimentación y las redes de retroalimentación. Más adelante se hará un énfasis en las redes neuronales de alimentación principalmente con el perceptrón multicapa.

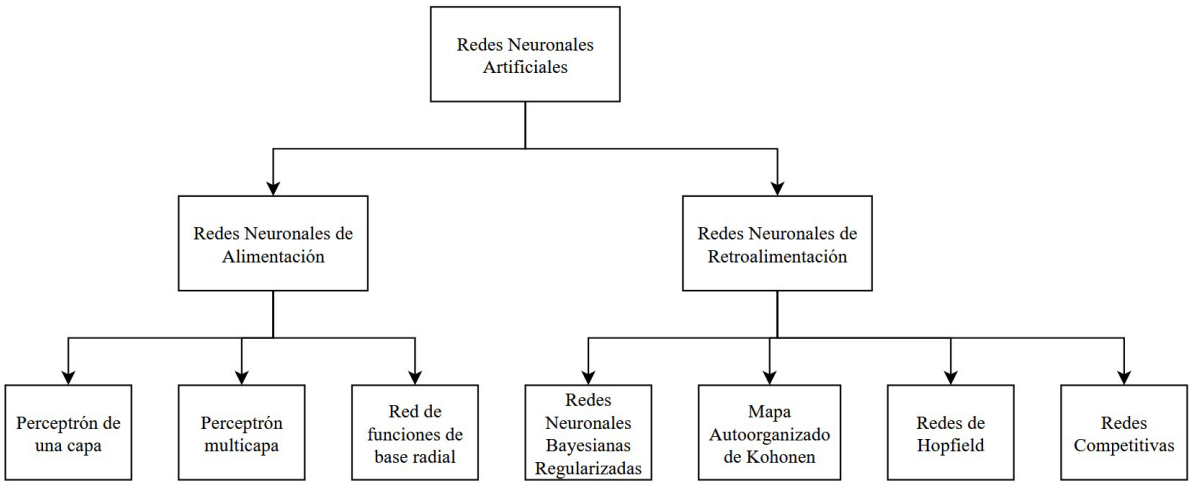


Fig. 6.5: Clasificación de las redes neuronales artificiales [86].

Es evidente que una capa solo se conecta a la capa anterior. Las aplicaciones de redes neuronales de alimentación directa se clasifican en dos categorías, como el control de sistemas dinámicos y los espacios donde se aplican técnicas clásicas de aprendizaje automático. Las redes neuronales con dos o más capas ocultas se llaman redes profundas porque la red se vuelve más compleja con más de una capa oculta. A diferencia de la red neuronal de alimentación directa, la red neuronal de retroalimentación puede utilizar un estado interno "memoria"(almacenar información) para procesar secuencias de entradas de datos. Las aplicaciones de la red neuronal de retroalimentación incluyen tareas como dessegmentación y reconocimiento de patrones.

En las redes neuronales de retroalimentación, las conexiones entre los nodos generan una gráfica coordinada en secuencia. Esta gráfica coordinada en secuencia permite que las redes neuronales de retroalimentación demuestren un comportamiento terrestre dinámico en una secuencia temporal. Ejemplos de esto son el mapa autoorganizado de Kohonen y la red neuronal recurrente. La red neuronal recurrente se refiere a un tipo estándar de red neuronal

que se extiende a lo largo del tiempo, con conexiones que se alimentan en el siguiente paso de tiempo en lugar de alimentarse en la siguiente capa concurrente en el tiempo.

6.2. Entrenamiento

Para el entrenamiento de la red neuronal sobre el vector de características, se utilizó una red neuronal de perceptrón multicapa. Adicionalmente para simular mejor el comportamiento de un cerebro humano, se adicionaron cuatro capas de eliminación entre las capas ocultas para eliminar rasgos o información demasiado especifica que en la realidad una persona no puede recordar y re-aprender a tal nivel de detalle.

La motivación detrás del uso de un perceptrón multicapa para probar la propuesta consiste en evitar la sobrecarga de una red neuronal o el uso de una red preentrenada. Se busca que la red sea lo suficientemente simple para demostrar la efectividad de la propuesta en relación con el modelado analítico de la mano, ya que este es parte del objetivo del trabajo: realizar un análisis más profundo y matemático sobre cómo modelar la mano utilizando objetos matemáticos y lograr una alta eficiencia.

La red neuronal es una red multicapa construida utilizando Keras, una biblioteca de aprendizaje profundo. La red consta de varias capas densas (también conocidas como capas completamente conectadas) con funciones de activación ReLU y una capa final con activación softmax para la clasificación. Esta conformada de la siguiente manera:

1. Capa de Entrada
    - Tipo: Capa densa (totalmente conectada)
    - Neuronas: 1024

- Función de Activación: ReLU
- Entrada: El tamaño de la entrada depende de las características utilizadas para hacer el entrenamiento.

2. Capa de Eliminación

   - Tipo: Capa de Eliminación (Dropout)
   - Proporción de Eliminación: 0.2
   - Descripción: La capa de eliminación elimina aleatoriamente conexiones entre las neuronas para evitar el sobreajuste.

3. Capa Oculta 1

   - Tipo: Capa densa (totalmente conectada)
   - Neuronas: 512
   - Función de Activación: ReLU

4. Capa de Eliminación

   - Proporción de Eliminación: 0.2

5. Capa Oculta 2

   - Tipo: Capa densa (totalmente conectada)
   - Neuronas: 256
   - Función de Activación: ReLU

6. Capa de Eliminación

   - Proporción de Elimincación: 0.2

7. Capa Oculta 3
    - Tipo: Capa densa (totalmente conectada)
    - Neuronas: 128
    - Función de Activación: ReLU
8. Capa de Eliminación
    - Proporción de Elimincación: 0.2
9. Capa Oculta 4
    - Tipo: Capa densa (totalmente conectada)
    - Neuronas: 64
    - Función de Activación: ReLU
10. Capa de Salida
    - Tipo: Capa densa (totalmente conectada)
    - Neuronas: Dependiendo del experimento realizado
    - Función de Activación: Softmax
    - Descripción: Utiliza softmax para producir probabilidades de pertenencia a cada clase.

El proceso continúa con la compilación del modelo, donde se define la función de pérdida, el optimizador y las métricas para evaluar el rendimiento del modelo durante el entrenamiento. Luego, el modelo se entrena utilizando los datos de entrenamiento que son el 80 % de los datos totales, durante 50 épocas con un tamaño de lote de entrenamiento de 64.

Finalmente, se evalúa el modelo utilizando los datos de prueba que representan el 20 % de los datos totales, y devuelve la precisión de prueba.

La arquitectura de la red neuronal tiene como objetivo la clasificación por clases, donde se buscan mutliples posibles clases de salida. Las capas de eliminación ayudan a regularizar el modelo y prevenir el sobreajuste. La función de activación ReLU es común en las capas ocultas debido a su capacidad para manejar problemas de desvanecimiento de gradientes y acelerar el entrenamiento.

6.2.1. Entrenamiento por clases

En la fase inicial de entrenamiento, se llevó a cabo un análisis comparativo exhaustivo que abarcó un conjunto de 18 clases distintas de gestos. El propósito primordial de esta etapa consistió en la identificación de la métrica o conjunto de características que exhibiera la máxima precisión. En este contexto, se procedió a realizar un proceso de entrenamiento con el fin de evaluar y determinar dicha métrica.

Es importante destacar que, adicionalmente, se llevó a cabo una segregación de gestos que manifestaban isomorfismo entre sí. Este fenómeno se refiere a la existencia de gestos con formas o estructuras idénticas, pero con orientaciones diferentes. Tomando este aspecto en consideración, se procedió a realizar un análisis comparativo adicional enfocado en los gestos que compartían esta característica. Un ejemplo de esto es la comparación entre el gesto de "me gustaↄ "no me gusta", en donde la configuración de la mano se mantiene constante, variando únicamente la orientación.

En este sentido, se implementó un proceso de entrenamiento específico para este subconjunto de gestos isomorfos. Cada uno de estos gestos fue sometido a un entrenamiento

individualizado de tipo uno a uno. El propósito detrás de este enfoque radicó en identificar la métrica o conjunto de características que ofreciera una mayor precisión en este contexto particular.

Es relevante resaltar que el proceso de entrenamiento abordó de manera separada cada una de las características involucradas en los gestos. Esto significó que se llevó a cabo una evaluación individual para cada característica en función de la lateralidad de la mano. Es decir, se compararon y analizaron las características de cada gesto tanto para la mano izquierda como para la mano derecha, además de considerar la combinación de ambas manos en conjunto.

En la tabla 6.1 se puede observar que la precisión del entrenamiento entre características no varia mucho, esto desde una perspectiva general al ingresar los 18 gestos a la red neuronal.

Gestos	Lateralidad	Características					Todas
		Coordenadas	Geométricas	Curvatura	Rotación	Bezier	
Todos	Izquierda	0.74	0.77	0.74	0.71	0.74	0.74
	Derecha	0.76	0.77	0.73	0.7	0.74	0.75
	Ambos	0.76	0.77	0.75	0.7	0.73	0.74

Tabla 6.1: Resultados de precisión para diferentes gestos y lateralidades, dependiendo la característica.

En la tabla 6.2 se puede observar que al hacer una comparativa 1 a 1 de gestos isomorfos, podemos observar que hay una variación más significativa de precisión respecto a la clasificación dependiendo de cada una de las características.

Por último repetimos el entrenamiento utilizando por un lado todos los gestos isomorfos y por otro lado el resto de los gestos, para observar si los gestos isomorfos tienen un impacto en la clasificación.

Gesto 1	Gesto 2	Lateralidad	Coordenadas	Geométricas	Curvatura	Rotación	Bezier	Todas
Like	Dislike	Izquierda	0,88	0,85	0,73	0,88	0,88	0,79
		Derecha	0,85	0,85	0,64	0,82	0,79	0,79
		Ambos	0,85	0,76	0,58	0,85	0,82	0,67
Mute	One	Izquierda	0,82	0,91	0,91	0,82	0,91	0,82
		Derecha	0,85	0,85	0,82	0,94	0,82	0,88
		Ambos	0,82	0,88	0,85	0,88	0,94	0,91
Stop	Stop Inv.	Izquierda	0,94	0,97	0,86	0,91	0,89	0,89
		Derecha	0,94	0,97	0,91	0,97	0,83	0,78
		Ambos	0,83	0,97	0,83	0,81	0,83	0,86
Peace	Peace Inv.	Izquierda	0,94	0,91	0,91	0,85	0,91	0,88
		Derecha	0,88	0,94	0,82	0,77	0,82	0,94
		Ambos	0,91	0,88	0,77	0,8	0,85	0,88
Two Up	Two Up Inv.	Izquierda	0,97	0,97	0,79	0,91	0,94	0,97
		Derecha	1	1	0,94	0,94	0,85	0,94
		Ambos	0,97	0,97	0,85	0,94	0,79	0,82

Tabla 6.2: Resultados de precisión para diferentes gestos y lateralidades, dependiendo la característica.

Tipo de Gesto	Lateralidad	Coordenadas	Geométricas	Curvatura	Rotación	Bezier	Todas
Gestos Isomorfos	Izquierda	0,8	0,84	0,69	0,61	0,76	0,73
	Derecha	0,81	0,75	0,75	0,69	0,74	0,73
	Ambos	0,76	0,8	0,67	0,65	0,72	0,83
Gestos NO Isomorfos	Izquierda	0,8	0,8	0,77	0,8	0,87	0,83
	Derecha	0,8	0,83	0,8	0,87	0,86	0,82
	Ambos	0,84	0,86	0,88	0,87	0,84	0,8

Tabla 6.3: Resultados de precisión para diferentes gestos y lateralidades, dependiendo la característica.

En la tabla 6.3 se puede observar que los gestos isomorfos tienen un impacto en algunas características, como en la rotación de las curvas bezier.

Analizando la tabla se observa que la clasificación de gestos isomorfos tiene una menor precisión respecto a gestos no isomorfos, esto es de esperarse debido a que en la clasificación lo que se busca es saber que tan diferente es una clase de otra, y los gestos isomorfos lo que hacen es tener un parecido muy alto respecto a otro gesto.

En una segunda fase, se realiza un entrenamiento mucho más detallado para así poder observar si la caracterización de gestos 1 a 1 ayuda a obtener una mayor precisión en la clasificación.

6.2.2. Entrenamiento detallado

En esta segunda fase de entrenamiento, se lleva a cabo un entrenamiento más detallado sobre qué característica o conjunto de características tiene mayor precisión en la clasificación.

Esto se realiza mediante un entrenamiento uno a uno de cada gesto, utilizando cada una de las características. Para lograr este objetivo, se realiza una modificación en la red neuronal en la capa de salida, definiéndola únicamente con dos neuronas para obtener la clasificación binaria. Esta modificación es el único cambio realizado para mantener la consistencia en la experimentación y el entrenamiento con respecto al entrenamiento de gestos en general.

En las siguientes matrices de confusión, se puede observar un análisis comparativo exhaustivo de cada uno de los gestos con cada una de las características. Cada matriz representa una caracterización distinta y no existe una distinción en la lateralidad, como ocurría en el entrenamiento anterior. Por lo tanto, los datos de entrenamiento incorporan

tanto la mano izquierda como la mano derecha. Los gestos G1 y G2 representan cada uno de los 18 gestos, numerados del 0 al 17.

Adicionalmente se crea una tabla con los resultados combinados de todas la matrices de confusión para una mejor observación comparativa sobre cada uno de los gestos, en donde podremos observar por el color, cuál es la característica que tiene mayor precisión entre cada par de gestos.

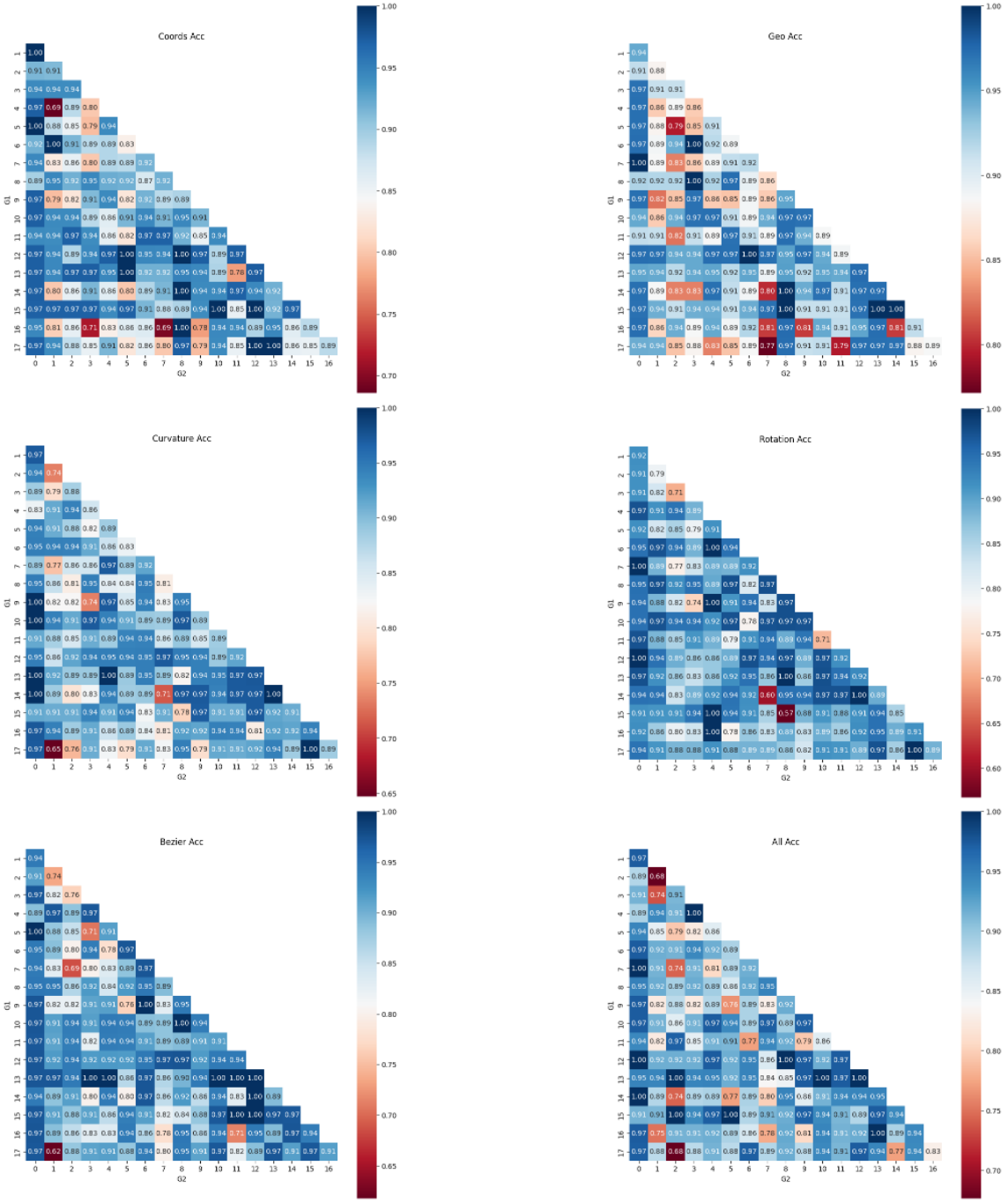
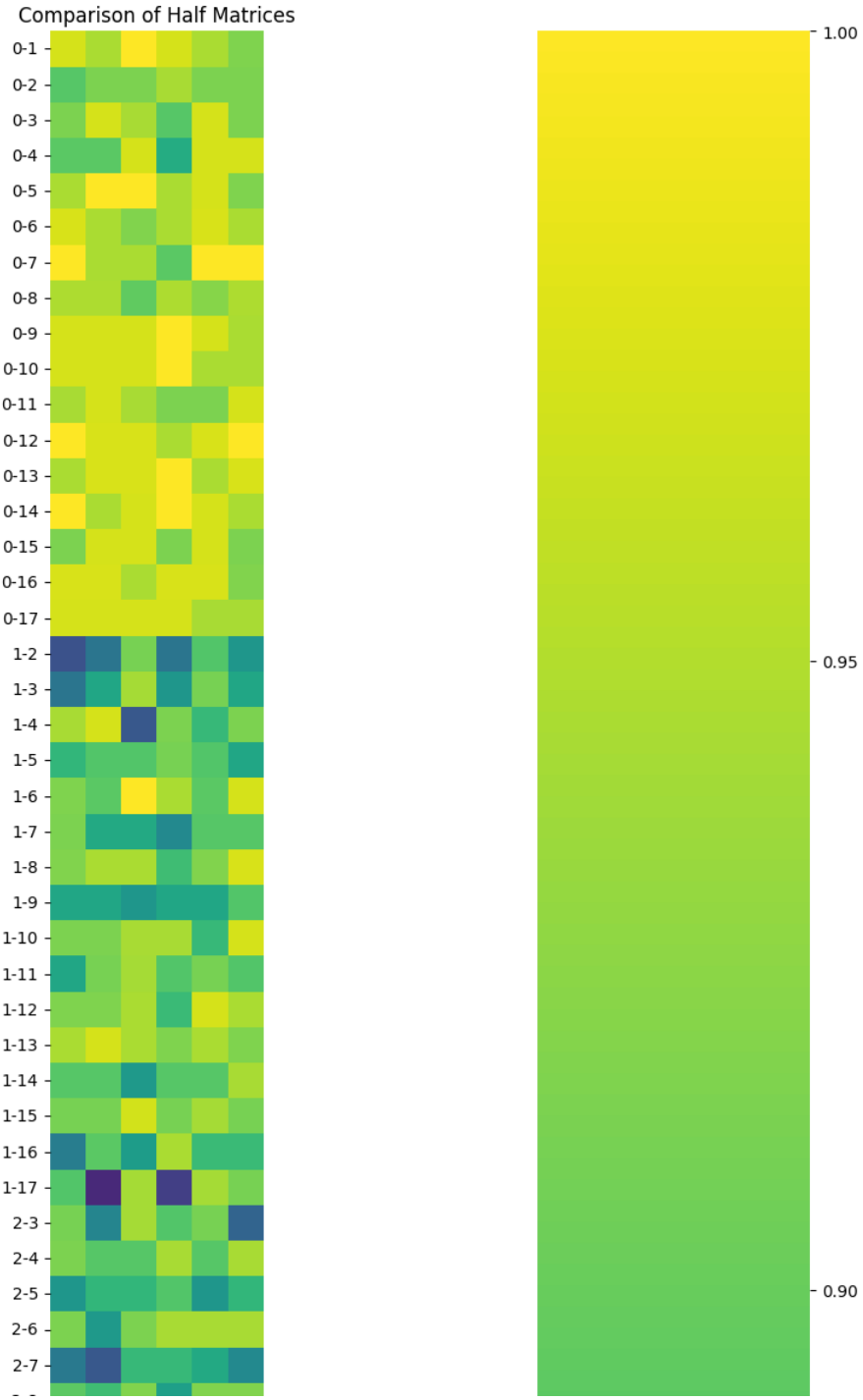
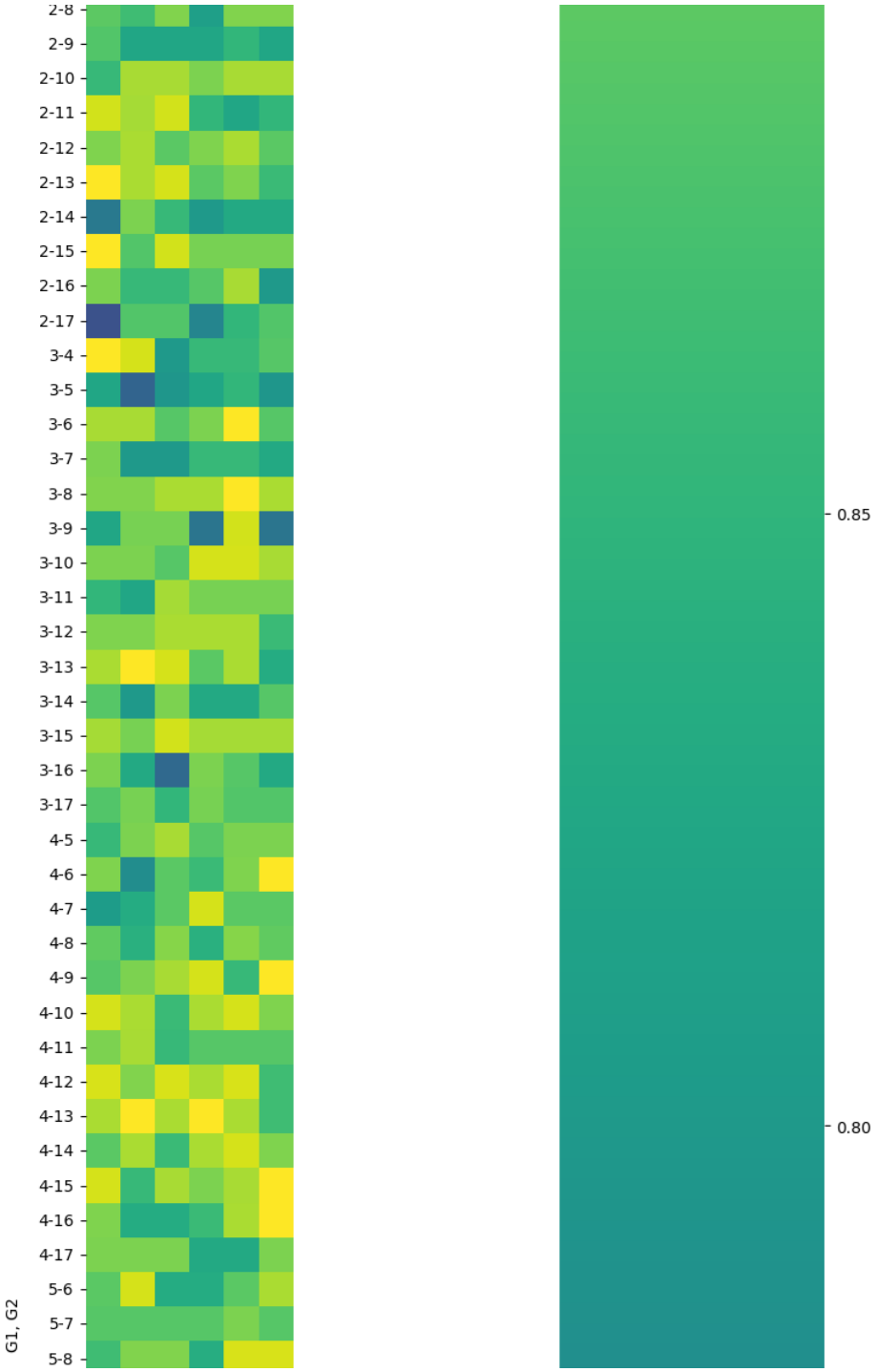
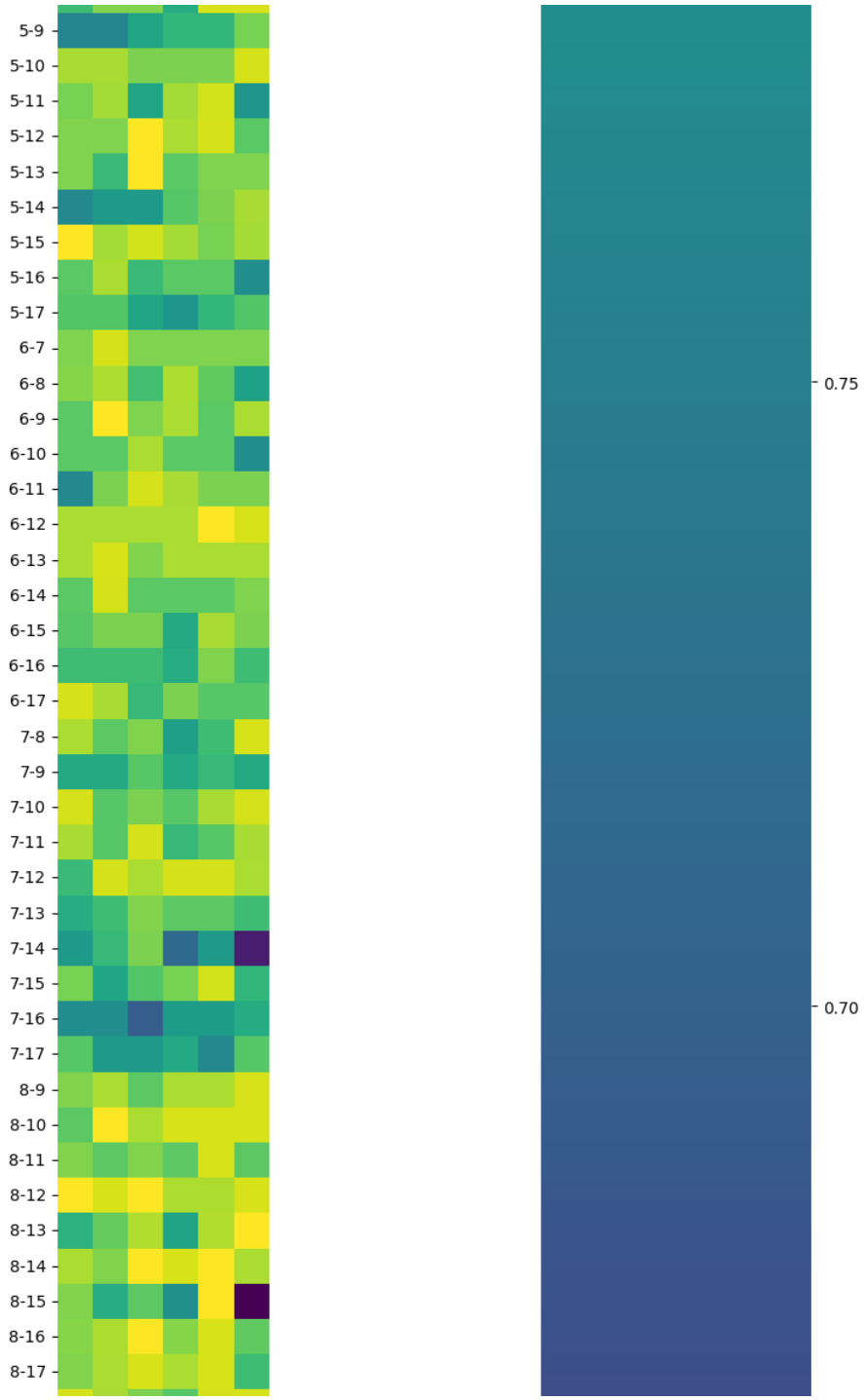


Fig. 6.6: Matrices de confusión


Comparison of Half Matrices
0-1
0-2
0-3
0-4
0-5
0-6
0-7
0-8
0-9
0-10
0-11
0-12
0-13
0-14
0-15
0-16
0-17
1-2
1-3
1-4
1-5
1-6
1-7
1-8
1-9
1-10
1-11
1-12
1-13
1-14
1-15
1-16
1-17
2-3
2-4
2-5
2-6
2-7
1.00
0.95
0.90


2-8
2-9
2-10
2-11
2-12
2-13
2-14
2-15
2-16
2-17
3-4
3-5
3-6
3-7
3-8
3-9
3-10
3-11
3-12
3-13
3-14
3-15
3-16
3-17
4-5
4-6
4-7
4-8
4-9
4-10
4-11
4-12
4-13
4-14
4-15
4-16
4-17
5-6
5-7
5-8
G1, G2
0.85
0.80


5-9
5-10
5-11
5-12
5-13
5-14
5-15
5-16
5-17
6-7
6-8
6-9
6-10
6-11
6-12
6-13
6-14
6-15
6-16
6-17
7-8
7-9
7-10
7-11
7-12
7-13
7-14
7-15
7-16
7-17
8-9
8-10
8-11
8-12
8-13
8-14
8-15
8-16
8-17
0.75
0.70

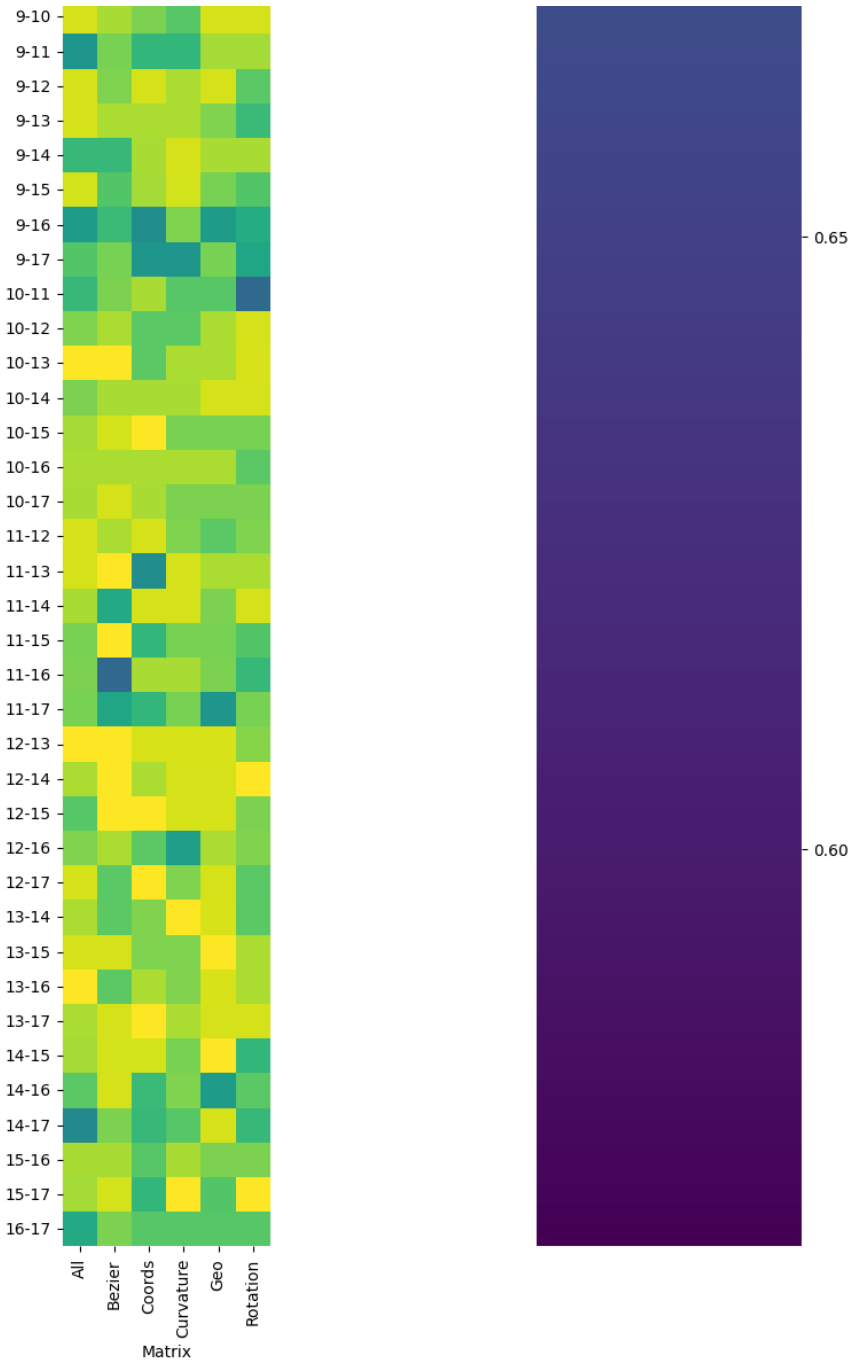


Fig. 6.7: Combinacion de todas las matrices de confusion

En las matrices de confusión 6.6 se observa que en la comparación 1 a 1 de cada uno de los gestos la precisión varia notablemente, en cada una de las características.

En la tabla 6.7 se observa directamente cual característica es más precisa respecto a cada par de vértices.

6.3. Análisis de Resultados

En la etapa inicial del proceso de entrenamiento, se puede observar un promedio aproximado de precisión del 75 % en los resultados obtenidos. Durante esta fase, se lleva a cabo el entrenamiento empleando un conjunto de 18 gestos específicos. En términos de lateralidad y las características empleadas en el proceso de entrenamiento, se observan cambios mínimos.

En este punto inicial de desarrollo, se hace énfasis en el concepto de isomorfismo de los gestos, que se refiere a la relación de forma y estructura entre diferentes gestos. Se destaca que, al analizar gestos no isomorfos en comparación con gestos isomorfos, se obtiene una métrica más efectiva en términos de la rotación de curvas bezier, así como en otras métricas también.

Tanto en situaciones donde se consideran múltiples gestos simultáneamente, como en aquellas donde se evalúan métricas más precisas, se logran resultados positivos. Sin embargo, en esta fase inicial, se revela un hallazgo intrigante. La nueva información adquirida indica que los gestos isomorfos tienden a ser menos precisos en la clasificación en comparación con sus contrapartes no isomorfos.

La segunda etapa del proceso de entrenamiento adopta un enfoque diferente. En

lugar de buscar una métrica universal o un conjunto de características que se aplique a todos los gestos, se opta por determinar la mejor métrica para cada par de gestos. Los resultados de esta fase son altamente satisfactorios, ya que se logra alcanzar una precisión del 100 % en múltiples ocasiones al clasificar los gestos en función de las métricas y características específicas para cada par.

Este nuevo enfoque demuestra ser beneficioso, ya que mejora la precisión en la clasificación binaria en su conjunto. En lugar de depender de una métrica general que se aplique a todos los gestos, ahora se tiene la capacidad de utilizar la métrica más adecuada para cada par de gestos. Esta propuesta ayuda a comprender y distinguir de manera más efectiva las particularidades de cada gesto en el proceso de clasificación. En definitiva, este estudio demuestra la importancia de considerar las relaciones individuales entre los gestos para lograr una clasificación más precisa y significativa.

Conclusiones

Este estudio ha demostrado la viabilidad y la efectividad de utilizar técnicas de extracción de características específicas de imágenes mediante software de visión por computadora para facilitar el aprendizaje automático, particularmente en el contexto de la clasificación de gestos de la mano.

El uso de herramientas como las curvas de Bezier, con su capacidad para describir de manera precisa y flexible las formas y contornos de los gestos, ha permitido obtener resultados prometedores en términos de precisión de clasificación. La exploración de propiedades como la curvatura, rotación y longitud ha ampliado el espectro de características que pueden ser extraídas y utilizadas para la clasificación de gestos.

La evaluación de dos enfoques de entrenamiento de modelos de aprendizaje automático ha proporcionado una visión clara de las ventajas y limitaciones de cada enfoque. Mientras que el primer enfoque, basado en conjuntos de gestos con características similares, ha demostrado una precisión promedio del 75 %, el segundo enfoque, que considera cada gesto de manera individual, ha mostrado una mayor variabilidad en la precisión, con valores que oscilan entre el 60 % y el 100 %.

Estos resultados sugieren que la exploración continua de nuevas herramientas y

técnicas de extracción de características podría conducir a mejoras adicionales en la precisión del aprendizaje automático, incluso en ausencia de conjuntos de datos masivos. Además, este trabajo destaca la importancia de considerar diferentes enfoques de modelado para adaptarse a las peculiaridades y variaciones de los datos de gestos de la mano.

Bibliografía

[1] N. Mohamed, M. B. Mustafa y N. Jomhari. A Review of the Hand Gesture Recognition System: Current Progress and Future Directions. *IEEE Access*, **9**:157422-157436, 2021. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3129650 (citado en las págs. 5, 8).

[2] A. Choudhury, A. Talukdar y K. Sarma. *A Review on Vision-Based Hand Gesture Recognition and Applications*. En ago. de 2015, páginas 261-286. ISBN: 13: 9781466684935. DOI: 10.4018/978-1-4666-8493-5.ch011 (citado en las págs. 6, 7).

[3] W. Zhou, C. Lyu, X. Jiang, P. Li, H. Chen e Y.-H. Liu. Real-time implementation of vision-based unmarked static hand gesture recognition with neural networks based on FPGAs. En páginas 1026-1031, dic. de 2017. DOI: 10.1109/ROBIO.2017.8324552 (citado en la pág. 6).

[4] A. Sharrma, A. Khandelwal, K. Kaur, S. Joshi, R. Upadhyay y S. Prabhu. Vision based static hand gesture recognition techniques. En páginas 0705-0709, abr. de 2017. DOI: 10.1109/ICCSP.2017.8286451 (citado en la pág. 6).

[5] A. Mahadi, F. Johora y M. Yousuf. An Efficient Approach of Training Artificial Neural Network to Recognize Bengali Hand Sign. En páginas 152-157, feb. de 2016. DOI: 10.1109/IACC.2016.37 (citado en la pág. 6).

[6] O. R. Chanu, A. Pillai, S. Sinha y P. Das. Comparative study for vision based and data based hand gesture recognition technique. *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*:26-31, 2017 (citado en la pág. 6).

[7] R. Y. L. H. N. W. R. R. R. Wahyu y Yuyu. Implementation of real-time static hand gesture recognition using artificial neural network. En páginas 1-6, ago. de 2017. DOI: 10.1109/CAIPT.2017.8320692 (citado en la pág. 6).

[8] D. Lu, Y. Yu y H. Liu. Gesture recognition using data glove: An extreme learning machine method. *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*:1349-1354, 2016 (citado en la pág. 6).

[9] H. Gunawardane y N. Medagedara. Comparison of Hand Gesture inputs of Leap Motion Controller Data Glove in to a Soft Finger. En oct. de 2017. DOI: 10.1109/IRIS.2017.8250099 (citado en la pág. 6).

[10] E. Pramunanto, S. Sumpeno y R. Legowo. Classification of hand gesture in Indonesian sign language system using Naive Bayes. En páginas 187-191, ago. de 2017. DOI: 10.1109/ISSIMM.2017.8124288 (citado en la pág. 6).

[11] D. Naglot y M. Kulkarni. Real time sign language recognition using the leap motion controller. *2016 International Conference on Inventive Computation Technologies (ICICT)*, **3**:1-5, 2016 (citado en la pág. 6).

[12] H. Badi, A. Hamza y S. Hasan. New method for optimization of static hand gesture recognition. En páginas 542-544, sep. de 2017. DOI: 10.1109/IntelliSys.2017.8324347 (citado en la pág. 6).

[13] T. Salunke y S. Bharkad. Power point control using hand gesture recognition based on hog feature extraction and k-nn classification. En páginas 1151-1155, jul. de 2017. DOI: 10.1109/ICCMC.2017.8282654 (citado en la pág. 6).

[14] H. Saha, S. Tapadar, S. Ray, S. Chatterjee y S. Saha. A Machine Learning Based Approach for Hand Gesture Recognition using Distinctive Feature Extraction. En páginas 91-98, ene. de 2018. DOI: 10.1109/CCWC.2018.8301631 (citado en la pág. 6).

[15] Q. Zhang, M. Yang, Z. Qinghe y X. Zhang. Segmentation of hand gesture based on dark channel prior in projector-camera system. En páginas 1-6, oct. de 2017. DOI: 10.1109/ICCChina.2017.8330336 (citado en la pág. 6).

[16] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta y K. Sharma. A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors. *IEEE Sensors Journal*, **16**(16):6425-6432, 2016. DOI: 10.1109/JSEN.2016.2581023 (citado en la pág. 6).

[17] C. Zhang, Y. Tian y M. Huenerfauth. Multi-modality American Sign Language recognition. En páginas 2881-2885, sep. de 2016. DOI: 10.1109/ICIP.2016.7532886 (citado en la pág. 6).

[18] C. J. L. Flores, A. E. G. Cutipa y R. L. Enciso. Application of convolutional neural networks for static hand gestures recognition under different invariant features. *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*:1-4, 2017 (citado en la pág. 6).

[19] M. Oudah, A. Al-Naji y J. Chahl. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *Journal of Imaging*, **6**(8), 2020. ISSN: 2313-433X. DOI: 10.3390/jimaging6080073. URL: https://www.mdpi.com/2313-433X/6/8/73 (citado en la pág. 7).

[20] Google. MediaPipe. https://google.github.io/mediapipe/, 2022 (citado en las págs. 8, 27, 28, 30).

[21] S. Inc. 3D Hand Tracking. https://docs.snap.com/lens-studio/references/templates/object/3d-hand-tracking, 2022 (citado en las págs. 8, 27).

[22] Y.-L. Chung, H.-Y. Chung y W.-F. Tsai. Hand gesture recognition via image processing techniques and deep CNN. *J. Intell. Fuzzy Syst.*, **39**:4405-4418, 2020 (citado en la pág. 8).

[23] S. P. More y A. Sattar. Hand gesture recognition system using image processing. En *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, páginas 671-675, 2016. DOI: 10.1109/ICEEOT.2016.7754766 (citado en la pág. 8).

[24] J. M. Fajardo, O. Gomez y F. Prieto. EMG hand gesture classification using handcrafted and deep features. *Biomedical Signal Processing and Control*, **63**:102210, 2021. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2020.102210. URL: https://www.sciencedirect.com/science/article/pii/S1746809420303426 (citado en la pág. 8).

[25] Y. Li y P. Zhang. Static hand gesture recognition based on hierarchical decision and classification of finger features. *Science Progress*, **105**(1):00368504221086362, 2022. DOI: 10.1177/00368504221086362. eprint: https://doi.org/10.1177/00368504221086362. URL: https://doi.org/10.1177/00368504221086362. PMID: 35296188 (citado en la pág. 8).

[26] R. Ambar, C. K. Fai, M. H. A. Wahab, M. M. A. Jamil y A. A. Ma'radzi. Development of a Wearable Device for Sign Language Recognition. *Journal of Physics: Conference Series*, **1019**(1):012017, 2018. DOI: 10.1088/1742-6596/1019/1/012017. URL: https://dx.doi.org/10.1088/1742-6596/1019/1/012017 (citado en la pág. 8).

[27] A. A. C. Illahi, M. F. M. Betito, C. C. F. Chen, C. V. A. Navarro e I. V. L.Or. Development of a Sign Language Glove Translator Using Microcontroller and Android Technology for Deaf-Mute. En *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, páginas 1-5, 2021. DOI: 10.1109/HNICEM54116.2021.9731835 (citado en la pág. 8).

[28] A. Tagliasacchi, M. Schroder, A. Tkach, S. Bouaziz, M. Botsch y M. Pauly. Robust Articulated-ICP for Real-Time Hand Tracking. *Computer Graphics Forum*, **34**, ago. de 2015. DOI: 10.1111/cgf.12700 (citado en las págs. 8, 18).

[29] L. Ge, Y. Cai, J. Weng y J. Yuan. Hand PointNet: 3D Hand Pose Estimation Using Point Sets. En *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 8417-8426, 2018. DOI: 10.1109/CVPR.2018.00878 (citado en la pág. 9).

[30] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai y J. Yuan. 3D Hand Shape and Pose Estimation from a Single RGB Image, 2019. DOI: 10.48550/ARXIV.1903.00812. URL: https://arxiv.org/abs/1903.00812 (citado en la pág. 9).

[31] A. Sharma, A. Mittal, S. Singh y V. Awatramani. Hand Gesture Recognition using Image Processing and Feature Extraction Techniques. *Procedia Computer Science*, **173**:181-190, 2020. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.

2020.06.022. URL: https://www.sciencedirect.com/science/article/pii/S187705092031526X. International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM2020 (citado en la pág. 9).

[32] M. Hasan y P. Mishra. Hand Gesture Modeling and Recognition using Geometric Features: A Review. *Canadian Journal on Image Processing and Computer Vision*, **3**:12-26, mar. de 2012 (citado en las págs. 9, 31, 33, 35).

[33] C.-H. Wu y C. H. Lin. Depth-based hand gesture recognition for home appliance control. En *2013 IEEE International Symposium on Consumer Electronics (ISCE)*, páginas 279-280, 2013. DOI: 10.1109/ISCE.2013.6570227 (citado en las págs. 9, 11).

[34] J. Francis y A. Kadan. Significance of Hand Gesture Recognition Systems in Vehicular Automation-A Survey. *International Journal of Computer Applications*, **99**:50-55, ago. de 2014. DOI: 10.5120/17389-7931 (citado en las págs. 9, 11).

[35] E. Nasr-Esfahani, N. Karimi, S. M. R. Soroushmehr, M. H. Jafari, M. A. Khorsandi, S. Samavi y K. Najarian. Hand Gesture Recognition for Contactless Device Control in Operating Rooms. *CoRR*, **abs/1611.04138**, 2016. arXiv: 1611.04138. URL: http://arxiv.org/abs/1611.04138 (citado en las págs. 9, 11).

[36] L. Ballan, A. Taneja, J. Gall, L. Van Gool y M. Pollefeys. Motion Capture of Hands in Action Using Discriminative Salient Points. En A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato y C. Schmid, edición, *Computer Vision – ECCV 2012*, páginas 640-653, Berlin, Heidelberg. Springer Berlin Heidelberg, 2012 (citado en la pág. 12).

[37] C. Nolker y H. Ritter. Visual recognition of continuous hand postures. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, **13**:983-94, feb. de 2002. DOI: 10.1109/TNN.2002.1021898 (citado en las págs. 12, 19).

[38] C. Wan, A. Yao y L. V. Gool. Direction matters: hand pose estimation from local surface normals, 2016. arXiv: 1604.02657 [cs.CV] (citado en la pág. 13).

[39] I. Oikonomidis, N. Kyriazis y A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. En *2012 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 1862-1869, 2012. DOI: 10.1109/CVPR.2012.6247885 (citado en la pág. 13).

[40] I. Oikonomidis, N. Kyriazis y A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. En volumen 1, ene. de 2011. DOI: 10.5244/C.25.101 (citado en la pág. 13).

[41] H. Ansar, A. Ksibi, A. Jalal, M. Shorfuzzaman, A. Alsufyani, S. A. Alsuhibany y J. Park. Dynamic Hand Gesture Recognition for Smart Lifecare Routines via K-Ary Tree Hashing Classifier. *Applied Sciences*, **12**(13), 2022. ISSN: 2076-3417. DOI: 10.3390/app12136481. URL: https://www.mdpi.com/2076-3417/12/13/6481 (citado en las págs. 14, 42).

[42] A. Jalal, I. Akhtar y K. Kim. Human Posture Estimation and Sustainable Events Classification via Pseudo-2D Stick Model and K-ary Tree Hashing. *Sustainability*,

12(23), 2020. ISSN: 2071-1050. DOI: 10.3390/su12239814. URL: https://www.mdpi.com/2071-1050/12/23/9814 (citado en la pág. 15).

[43] A. Kapitanov, A. Makhlyarchuk y K. Kvanchiani. HaGRID - HAnd Gesture Recognition Image Dataset, 2022. arXiv: 2206.08219 [cs.CV] (citado en las págs. 15, 38, 39).

[44] D. K. Ghosh y S. Ari. Static Hand Gesture Recognition Using Mixture of Features and SVM Classifier. En *2015 Fifth International Conference on Communication Systems and Network Technologies*, páginas 1094-1099, 2015. DOI: 10.1109/CSNT.2015.18 (citado en la pág. 16).

[45] J. Li, J. Wang y Z. Ju. A Novel Hand Gesture Recognition Based on High-Level Features. *International Journal of Humanoid Robotics*, **15**:1750022, oct. de 2017. DOI: 10.1142/S0219843617500220 (citado en la pág. 16).

[46] H. Badi, S. A. Kareem y S. Husien. Feature Extraction Technique for Static Hand Gesture Recognition. En 2015 (citado en la pág. 16).

[47] A. Chahid, R. Khushaba, A. Al-Jumaily y T.-M. Laleg-Kirati. A Position Weight Matrix Feature Extraction Algorithm Improves Hand Gesture Recognition, 2020. DOI: 10.1109/EMBC44109.2020.9176097. URL: http://hdl.handle.net/10754/665147 (citado en la pág. 17).

[48] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu y J. Li. Gesture recognition by instantaneous surface EMG images. *Scientific Reports*, **6**:36571, nov. de 2016. DOI: 10.1038/srep36571 (citado en la pág. 17).

[49] M. Pu, C. Y. Chong y M. K. Lim. Robustness Evaluation in Hand Pose Estimation Models using Metamorphic Testing, 2023. arXiv: 2303.04566 [cs.CV] (citado en la pág. 17).

[50] J. M. Rehg y T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. En J.-O. Eklundh, edición, *Computer Vision — ECCV '94*, páginas 35-46, Berlin, Heidelberg. Springer Berlin Heidelberg, 1994. ISBN: 978-3-540-48400-4 (citado en la pág. 18).

[51] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang y M. Grundmann. MediaPipe Hands: On-device Real-time Hand Tracking. https://arxiv.org/abs/2006.10214, 2020. DOI: 10.48550/ARXIV.2006.10214 (citado en las págs. 18, 28).

[52] S. Deepa. *PRINCIPLES OF SOFT COMPUTING*. Wiley India Pvt. Limited, 2007. ISBN: 9788126510757. URL: https://books.google.com.mx/books?id=CXruGgP0BTIC (citado en la pág. 19).

[53] D. Lee e Y. Park. Vision-based remote control system by motion detection and open finger counting. *IEEE Transactions on Consumer Electronics*, **55**(4):2308-2313, 2009. DOI: 10.1109/TCE.2009.5373803 (citado en la pág. 19).

[54] Y. Wang y G. Mori. Max-margin hidden conditional random fields for human action recognition. En *2009 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 872-879, 2009. DOI: 10.1109/CVPR.2009.5206709 (citado en la pág. 19).

[55] F. Zhan. Hand Gesture Recognition with Convolution Neural Networks. En *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, páginas 295-298, 2019. DOI: 10.1109/IRI.2019.00054 (citado en las págs. 19, 25).

[56] J. Yu, M. Qin y S. Zhou. Dynamic gesture recognition based on 2D convolutional neural network and feature fusion. *Scientific Reports*, **12**:4345, mar. de 2022. DOI: 10.1038/s41598-022-08133-z (citado en la pág. 20).

[57] Y.-C. Jhaung, Y.-M. Lin, C. Zha, J.-S. Leu y M. Köppen. Implementing a Hand Gesture Recognition System Based on Range-Doppler Map. *Sensors*, **22**(11), 2022. ISSN: 1424-8220. DOI: 10.3390/s22114260. URL: https://www.mdpi.com/1424-8220/22/11/4260 (citado en la pág. 20).

[58] D. Hernandez Peña. Reconocimiento de señas con movimiento en el lenguaje de señas mexicano, 2021 (citado en la pág. 21).

[59] J. Zheng, Z. Zhao, M. Chen, J. Chen, C. Wu, Y. Chen, X. Shi y T. Yiqi. An Improved Sign Language Translation Model with Explainable Adaptations for Processing Long Sign Sentences. *Computational Intelligence and Neuroscience*, **2020**:1-11, oct. de 2020. DOI: 10.1155/2020/8816125 (citado en la pág. 21).

[60] J. Pu, W. Zhou y H. Li. Iterative Alignment Network for Continuous Sign Language Recognition. En *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 4160-4169, 2019. DOI: 10.1109/CVPR.2019.00429 (citado en la pág. 21).

[61] P. K. Prasad, A. P. Shibu et al. Intelligent Human Sign Language Translation using Support Vector Machines Classifier. *IJRAR-International Journal of Research and Analytical Reviews (IJRAR)*, **5**(4):461-466, 2018 (citado en la pág. 22).

[62] H. Wang, X. Chai, X. Hong, G. Zhao y X. Chen. Isolated Sign Language Recognition with Grassmann Covariance Matrices. *ACM Trans. Access. Comput.*, **8**(4), 2016. ISSN: 1936-7228. DOI: 10.1145/2897735. URL: https://doi.org/10.1145/2897735 (citado en la pág. 23).

[63] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, T. S. Alrayes, H. Mathkour y M. A. Mekhtiche. Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation. *IEEE Access*, **8**:192527-192542, 2020. DOI: 10.1109/ACCESS.2020.3032140 (citado en la pág. 23).

[64] G. Huang, I. Laradji, D. Vázquez, S. Lacoste-Julien y P. Rodríguez. A Survey of Self-Supervised and Few-Shot Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*:1-20, 2022. DOI: 10.1109/TPAMI.2022.3199617 (citado en la pág. 25).

[65] R. Verschae y J. Ruiz-del Solar. Object Detection: Current and Future Directions. *Frontiers in Robotics and AI*, **2**, 2015. ISSN: 2296-9144. DOI: 10.3389/frobt.2015.

00029. URL: https://www.frontiersin.org/articles/10.3389/frobt.2015.00029 (citado en la pág. 25).

[66] Z. Cao, T. Simon, Y. Raaj, H. Joo e Y. Sheikh. OpenPose, 2022. URL: https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/index.html (citado en la pág. 27).

[67] Banuba. Baunba Software. https://www.banuba.com/, 2022 (citado en la pág. 27).

[68] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg y M. Grundmann. MediaPipe: A Framework for Building Perception Pipelines, 2019. DOI: 10.48550/ARXIV.1906.08172. URL: https://arxiv.org/abs/1906.08172 (citado en la pág. 28).

[69] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg y M. Grundmann. MediaPipe: A Framework for Perceiving and Processing Reality. En *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019. URL: https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf (citado en la pág. 28).

[70] Y. Li y F. Ren. Light-Weight RetinaNet for Object Detection, 2019. arXiv: 1905.10011 [cs.CV] (citado en la pág. 29).

[71] S. K. Kang, M. Y. Nam y P. K. Rhee. Color Based Hand and Finger Detection Technology for User Interaction. En *2008 International Conference on Convergence and Hybrid Information Technology*, páginas 229-236, 2008. DOI: 10.1109/ICHIT.2008.292 (citado en la pág. 31).

[72] K. Oka, Y. Sato y H. Koike. Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. En *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, páginas 429-434, 2002. DOI: 10.1109/AFGR.2002.1004191 (citado en la pág. 31).

[73] A. O. Salau y S. Jain. Feature Extraction: A Survey of the Types, Techniques, Applications. En *2019 International Conference on Signal Processing and Communication (ICSC)*, páginas 158-164, 2019. DOI: 10.1109/ICSC45622.2019.8938371 (citado en la pág. 42).

[74] T. Shih, J. Huang, C.-S. Wang, J. Hung y C.-H. KAO. An Intelligent Content-based Image Retrieval System Based on Color, Shape and Spatial Relations. **25**, jul. de 2001 (citado en la pág. 42).

[75] P.-J. Laurent y P. Sablonnière. Pierre Bézier: An engineer and a mathematician. *Computer Aided Geometric Design*, **18**(7):609-617, 2001. ISSN: 0167-8396. DOI: https://doi.org/10.1016/S0167-8396(01)00056-5. URL: https://www.sciencedirect.com/science/article/pii/S0167839601000565. Pierre Bzier (citado en la pág. 48).

[76] T. Sauer. *Numerical analysis*. Addison-Wesley Publishing Company, 2011 (citado en las págs. 49, 51).

[77] M. Abdelaal y S. Schön. Predictive Path Following and Collision Avoidance of Autonomous Connected Vehicles. *Algorithms*, **13**:52, feb. de 2020. DOI: 10.3390/a13030052 (citado en la pág. 50).

[78] M. Unser. Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, **16**(6):22-38, 1999. DOI: 10.1109/79.799930 (citado en la pág. 52).

[79] W. mula. *B-spline with control points/control polygon, and marked component curves*. Wikipedia, 2013. URL: https://es.wikipedia.org/wiki/B-spline#/media/Archivo:B-spline_curve.svg (citado en la pág. 53).

[80] I. Kucukoglu, B. Simsek e Y. Simsek. Multidimensional Bernstein polynomials and Bezier curves: Analysis of machine learning algorithm for facial expression recognition based on curvature. *Applied Mathematics and Computation*, **344-345**:150-162, 2019. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2018.10.012. URL: https://www.sciencedirect.com/science/article/pii/S0096300318308725 (citado en la pág. 54).

[81] C.-K. Lee, H.-D. Hwang y S.-H. Yoon. Bézier Triangles with G2 Continuity across Boundaries. *Symmetry*, **8**(3), 2016. ISSN: 2073-8994. DOI: 10.3390/sym8030013. URL: https://www.mdpi.com/2073-8994/8/3/13 (citado en la pág. 58).

[82] I. E. Naqa, R. Li y M. J. Murphy. *Machine learning in Radiation oncology: Theory and applications*. Springer International Publishing, 2015 (citado en las págs. 66, 68).

[83] B. Mehlig. *Machine Learning with Neural Networks*. Cambridge University Press, 2021. DOI: 10.1017/9781108860604. URL: https://doi.org/10.1017\%2F9781108860604 (citado en la pág. 70).

[84] F. Farfán. *Control Cerebral de Interfases: Análisis Exploratorio de Técnicas Paramétricas Digitales para la Detección y Cuantificación de Estados Mentales*. Tesis doctoral, ene. de 2005. DOI: 10.13140/RG.2.1.1649.1123 (citado en la pág. 70).

[85] J. Zou, Y. Han y S.-S. So. *Overview of Artificial Neural Networks*. En *Artificial Neural Networks: Methods and Applications*. D. J. Livingstone, edición. Humana Press, Totowa, NJ, 2009. ISBN: 978-1-60327-101-1. DOI: 10.1007/978-1-60327-101-1_2. URL: https://doi.org/10.1007/978-1-60327-101-1_2 (citado en la pág. 71).

[86] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed y H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, **4**(11), 2018. ISSN: 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2018.e00938. URL: https://www.sciencedirect.com/science/article/pii/S2405844018332067 (citado en las págs. 72, 73, 76).

[87] I. Arroyo-Fernández. *Evaluación de dos técnicas de reconocimiento de patrones para su implementación en el Simulador de pilotaje automático de taller del STC Metro de la Cd. de México*. Tesis doctoral, nov. de 2013. DOI: 10.13140/RG.2.1.3938.3524 (citado en la pág. 74).